

Unsupervised Equalization over Multitask Adaptive Networks

Gustavo Depieri and Aline Neves

Abstract—Adaptive Networks (ANs) are groups of agents that collaborate to perform data-driven tasks such as optimization and inference. This work evaluates distributed filter estimation for unsupervised equalization over ANs using adaptive signal processing algorithms: the Constant Modulus Algorithm and Single Channel Independent Component Analysis (SCICA). Equalization is assumed to be performed using a two-filter cascade in a multitask setting, where agents experience different but correlated channels. The simulation results show the advantage of the collaborative algorithm over the standalone implementation. Additionally, a strategy for reducing data exchange while maintaining acceptable network performance is proposed for the distributed SCICA.

Keywords—Multitask Adaptive Networks, Unsupervised Equalization, Single Channel Independent Component Analysis, Constant Modulus Algorithm.

I. INTRODUCTION

Adaptive Networks (ANs) are systems that perform distributed parameter estimation across cooperative agents for tasks such as optimization and inference [1]. The motivation for research and applications involving ANs stems from the benefits provided by the collaborative and decentralized network approach, which enhances performance [1] and data privacy [2]. Decentralized systems are also shown to be more resilient to failure than centralized solutions, in which a node aggregates data collected from different network locations and performs parameter estimation, creating a critical single point that, in the case of failure, affects the whole system [1]. Examples of applications include wireless sensor networks, which are the research field of the earliest publications on ANs [2], and federated learning for the distributed training of machine learning models [2].

This work builds on the results and proposes an extension of the methods presented in [3] to address the online blind deconvolution problem using classical adaptive filter algorithms in the ANs setting, namely the Constant Modulus Algorithm (CMA) [4] and the Independent Component Analysis (ICA)-based algorithm, Single Channel ICA (SCICA) [5]. The CMA assumes that the modulus statistics of the source are known to perform equalization. The SCICA is related to the problem of Blind Source Separation (BSS), considering methods based on Independent Component Analysis (ICA) [6], but it assumes mixtures obtained from a single source. The SCICA is based on the premise of a non-Gaussian and i.i.d source. As it makes

weaker assumptions about the data, the SCICA can be applied to a broader class of sources than the CMA. The CMA and its related algorithm, SATO [4], are found in the literature on blind equalization over ANs in works such as [7], [8], while the SCICA application in ANs is proposed by [3].

For both CMA and SCICA, the ANs's multitask framework is considered for equalization, in which agents cooperate to estimate a related set of parameters whose optimal solutions are not unique across the network [9]. Concurrently, the channels are related, featuring both common and distinct impulse responses across agents, which align with the multitask model for ANs. The equalization filter, estimated by each agent, is as proposed in [3]: a two-filter cascade in which only the first filter's parameters are shared across the network, while the second filter is independently estimated by each agent, allowing them to adjust the equalizer for their channel-specific characteristics.

This work extends and evaluates the two-filter cascade for the CMA algorithm, namely AN-CMA. For the SCICA algorithm implemented over the ANs, this work proposes the AN-SCICA-S with a matrix row-sampling strategy and an interval between consecutive transmissions, thereby reducing energy consumption and network data transmission. The simulations show that the AN-CMA can perform equalization in multitask networks using the two-filter approach. At the same time, the AN-SCICA-S maintains satisfactory performance with a sampling-based parameter exchange strategy.

The main contributions of this paper, which extend the proposed methods in [3], are:

- Extending the two-filter equalization structure for the AN-CMA.
- A matrix row-sampling method for reducing the amount of data transmitted in the AN-SCICA-S network.

This work is organized as follows: Section II presents the general concept of an AN in the multitask framework for equalization exploited in this work. Section III briefly discusses the classical algorithms CMA and SCICA and shows the proposed algorithms for the AN-CMA and AN-SCICA-S. Section IV contains the simulations and analysis regarding the AN-CMA and AN-SCICA-S, and Section V concludes the work.

II. MULTITASK DIFFUSION ADAPTIVE NETWORKS

As stated in the introduction, this work considers the unsupervised adaptive filter estimation for equalization over diffusion multitask ANs. A multitask AN in this context comprises a collection of agents that receive data streams and

collaboratively estimate a set of parameters for the equalization filter that are related to each other.

The whole network with K agents can be seen as optimizing the aggregated global cost function (1) [9].

$$J(\mathbf{b}_0, \dots, \mathbf{b}_{K-1}) = \sum_{k=0}^{K-1} J_k(\mathbf{b}_k), \quad (1)$$

where J_k represents the cost function of the k -th agent and $\mathbf{b}_k$ is its equalization filter, modeled here as a Finite Impulse Response (FIR) type with a dimension $M \times 1$. Each agent is assumed to sample from a random process x_k , which is the transmitted source linearly filtered by the channel. The desired source estimation y_k^o is obtained as:

$$y_k^o = \mathbf{b}_k^o * x_k, \quad (2)$$

where $*$ denotes the convolution operation and $\mathbf{b}_k^o = \arg \min_{\mathbf{b}_k} (J_k(\mathbf{b}_k))$ are the parameters of the optimum equalization filter for minimizing its cost function. Hence, the global minimum is achieved when each agent estimation converges to its local optimum, i.e., $J_{min} = J(\mathbf{b}_0^o, \dots, \mathbf{b}_{K-1}^o)$.

Once the multitask framework is assumed, it is important to clarify how agents can cooperate if their cost functions, and therefore the optimal parameter sets they are trying to find, differ. Some works in the literature model the problem by grouping agents with similar objective functions to cooperate, where the clusters are assumed to be known or are learned through unsupervised strategies [2]. For this work, the approach is based on [3]. It is assumed, for simplicity, that the network's behavior is synchronous, with no random failures in connections or agents. The source and noise will be modeled as independent across agents and time, as in multiple-input multiple-output (MIMO) systems. The channel and equalizer model that enables multitask AN cooperation is discussed next.

A. Channel and Equalization Model in the AN

The equalization is performed by the agents that are subjected to the following channel model [3]:

$$C_k = C_{(1)} * C_{(k,2)}, \quad (3)$$

where C_k is the channel impulse response for the k -th agent, expressed as a convolution of two underlying impulse responses: $C_{(1)}$, which is experienced commonly across all agents in the network, and $C_{(k,2)}$, which is different for each agent. This two stage channel model enables the multitask framework since part of the equalizer impulse response is shared among the agents. It is also supported by a reasonable practical assumption that the agents may experience similar channel fading under specific conditions; for example, as discussed in [8].

The observed random process x_k for the k -th agent is given by:

$$x_k = C_k * s_k + v_k, \quad (4)$$

with s_k as the transmitted source that is intended to be recovered, and C_k as the channel impulse response (3). The v_k term is the Additive White Gaussian Noise (AWGN). This

work assumes each source s_k is i.i.d and obtained through a different realization of a random process across agents.

Assuming that the channel (3) admits an inverse, the equalization filter is given by $\mathbf{b}_k^o(z) \propto \frac{1}{C_{(1)}(z)} \frac{1}{C_{(k,2)}(z)}$, where $\mathbf{b}_k^o(z)$, $C_{(1)}(z)$, and $C_{(k,2)}(z)$ are the Z-transforms of the FIR equalizer filter $\mathbf{b}_k^o$ and the channel stages $C_{(1)}$ and $C_{(k,2)}$, respectively.

Therefore, the two-stage channel model suggests implementing the equalizer as a two-filter cascade [3], as shown in Fig. 1, in which only the parameters of the first filter are shared across the network while the agents independently estimate the second one.

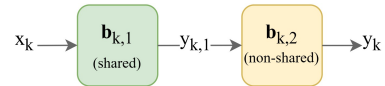


Fig. 1. A two-filter cascade equalization filter for the k -th agent in the network.

Cooperation is achieved through information exchange among agents, who share and combine their local estimates with their neighbors, and whose connections are established according to the network topology [1]. In the Adapt then Combine (ATC) policy, the agents first adapt their local estimates, then share and combine them with neighboring nodes [1]. For diffusion networks performed at the first stage, the ATC equation is:

$$\begin{cases} \psi_k[n+1] \leftarrow \mathbf{b}_{k,1}[n] - \mu_{k,1} \nabla J_k(\mathbf{b}_{k,1}[n], \mathbf{x}_k[n]), \\ \mathbf{b}_{k,1}[n+1] \leftarrow \sum_{l \in \mathcal{N}_k} a_{kl} \psi_l[n+1]. \end{cases} \quad (5)$$

with the samples of the process x_k being stacked into a column vector $\mathbf{x}_k[n] = (x_k[n], x_k[n-1], \dots, x_k[n-M_1+1])^T$ of size $M_1 \times 1$. The $\mathcal{N}_k$ is the set of neighbors of the k -th agent, and the coefficients $\{a_{kl}\}$ represent connection weights for each neighbor $l \in \mathcal{N}_k$. They are defined by the topology of the network and by a rule, such as the Metropolis Rule and the Averaging Rule [1]. These coefficients have the convex property $\sum_{l \in \mathcal{N}_k} a_{kl} = 1$. The term $\nabla J_k(\cdot)$ is the gradient of the cost function J_k evaluated online at the current filter estimation and input data, and $\mu_{k,1}$ is the update step.

The first stage outputs the filtered signal sample

$$y_{k,1}[n] = \mathbf{b}_{k,1}^T[n+1] \mathbf{x}_k[n], \quad (6)$$

which is collected in the column vector $\mathbf{y}_{k,1}[n] = (y_{k,1}[n], y_{k,1}[n-1], \dots, y_{k,1}[n-M_2+1])^T$ of size $M_2 \times 1$ and fed as input data for the second stage.

At the second stage, the agent performs the non-cooperative gradient update as follows:

$$\mathbf{b}_{k,2}[n+1] \leftarrow \mathbf{b}_{k,2}[n] - \mu_{k,2} \nabla J_k(\mathbf{b}_{k,2}[n], \mathbf{y}_{k,1}[n]) \quad (7)$$

where $\mu_{k,2}$ is the update step. The combined cascade has the impulse response of the equalization filter $\mathbf{b}_k$ given by:

$$\mathbf{b}_k = \mathbf{b}_{k,1} * \mathbf{b}_{k,2}, \quad (8)$$

with size $M \times 1$, with $M = (M_1 + M_2 - 1)$. The signal at the output is $y_k = \mathbf{b}_k * x_k$, which should converge to the desired source estimation (2).

The combined channel and filter cascade has an impulse response $\Lambda_k[n]$:

$$\Lambda_k[n] = C_k * \mathbf{b}_k[n]. \quad (9)$$

The equalization is perfectly achieved when $\Lambda_k[n]$ corresponds to a distortionless system, for which the output signal is equal to the input signal, up to a delay and a scaling factor.

The following section presents the implementation of AN-CMA using the cascade filter and AN-SCICA-S with the proposed row-sampling technique.

III. AN-CMA AND AN-SCICA-S ALGORITHMS

This section presents the implementation of the CMA and the SCICA algorithm in the ANs scenario. For the CMA, the contribution of this work is the implementation of the multitask AN-CMA using the two-filter cascade model from [3]. For the AN-SCICA-S, this work proposes a row-sampling technique in order to reduce the number of shared parameters and adds a time interval between successive ATC combination steps for the first filter estimation.

A. AN-CMA Algorithm

The CMA algorithm [4] assumes that the modulus statistics of the source s_k are known and performs the minimization of the cost function $J_{cma} = E\{(R_2 - |y[n]|^2)^2\}$ with $R_2 = E\{|s[n]|^4\}/E\{|s[n]|^2\}$ [4]. Additionally, the CMA may be used for subgaussian sources with non-constant modulus. For the digital modulation Binary Phase Shift Keying (BPSK) and the standard uniform distribution sources considered in this work, $R_2 = 1$ and $R_2 = 0.6$, respectively.

Taking the stochastic gradient of J_{cma} , the first stage of the equalizer (5) for the AN-CMA becomes:

$$\begin{cases} \psi_k[n+1] \leftarrow \mathbf{b}_{k,1}[n] + \mu_{k,1}\hat{y}_{k,1}[n](1 - |\hat{y}_{k,1}[n]|^2)\mathbf{x}_k[n], \\ \mathbf{b}_{k,1}[n+1] \leftarrow \sum_{l \in \mathcal{N}_k} a_{kl}\psi_l[n+1], \end{cases} \quad (10)$$

with $\hat{y}_{k,1}[n] = \mathbf{b}_{k,1}^T[n]\mathbf{x}_k[n]$, and initializing $\mathbf{b}_k[0] = (0, \dots, 1, \dots, 0)^T$ as center-spike - the central tap equal to 1 and the others 0.

The second stage receives the data from the first stage as (6) and performs the second stage update as:

$$\mathbf{b}_{k,2}[n+1] \leftarrow \mathbf{b}_{k,2}[n] + \mu_{k,2}\hat{y}_k[n](1 - |\hat{y}_k[n]|^2)\mathbf{y}_{k,1}[n], \quad (11)$$

with $\hat{y}_k[n] = \mathbf{b}_{k,2}^T[n]\mathbf{y}_{k,1}[n]$. The equalizer filter is the combined impulse response as in (8).

B. AN-SCICA-S Algorithm

The SCICA [5] is a blind algorithm that uses higher-order statistics to perform the deconvolution of the observed signal, retrieving the source, which is assumed to be a non-Gaussian i.i.d process. The SCICA is a special case of the broader field of BSS within the ICA framework [6] that blindly separates statistically independent sources from an observed set of linearly mixed signals.

Some approaches associated with these ICA methods rely on the Central Limit Theorem, which states that the sum of random variables converges to a Gaussian distribution [6]. The objective function in these lines searches for a linear transformation that maximizes the non-Gaussianity of the signals and, therefore, the statistical independence assumption for retrieving the sources. An example of such an objective function is the negentropy, which measures the difference between the entropy of a Gaussian variable (the highest-entropy probability distribution) and the desired estimated signal. By maximizing the negentropy, one maximizes the non-Gaussianity and, therefore, recovers the sources [6].

The numerical approximation for the negentropy is $J_n(y) = (E\{G(y)\} - E\{G(\eta)\})^2$, where y is the estimated signal and η is a Gaussian variable that has the same second-order statistics as y . From many possibilities for $G(\cdot)$, this work chooses the non-linear function $G(y) = -\exp(-\frac{y^2}{2})$ [6].

For the SCICA, it is assumed that a source s is filtered by a system whose impulse response is C . The observed convoluted signal x is given by $x = C * s$. Both its impulse response and the source are unknown [5]. The system introduces statistical dependence between x and its delayed version. Thereby, the SCICA maximizes the statistical independence of the vector with the observed signal $\mathbf{x}[n] = (x[n], x[n-1], \dots, x[n-M+1])^T$ by estimating a filter $\mathbf{b}$ of size $M \times 1$ to recover the source. A commonly used pre-processing step is to first decorrelate $\mathbf{x}[n]$ by estimating a whitening matrix $\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^T$ (of size $M \times M$) using the eigen-decomposition of the correlation matrix $\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\}$, where $\mathbf{E}$ is a matrix with the eigen-vectors of $\mathbf{R}_x$ and $\mathbf{D}$ contains its eigen-values [6]. From the decorrelated signal $\mathbf{z}[n] = \mathbf{V}\mathbf{x}[n]$, a filter $\mathbf{w}$ ($M \times 1$) is obtained by a gradient method to optimize the negentropy, recovering the source. The SCICA estimation of the source is $y = (\mathbf{w}^T \mathbf{V}) * x = \mathbf{b} * x$:

In the following, the stochastic implementation of the proposed multitask AN-SCICA-S is provided, which is an extension of the multitask AN-SCICA proposed in [3]. The ‘‘adapt’’ step for the first stage is:

$$\begin{cases} \mathbf{z}_{k,1}[n] \leftarrow \mathbf{V}_{k,1}[n]\mathbf{x}_k[n], \\ \Psi_{V,k}[n+1] \leftarrow (\mathbf{I} + \mu_{k,1}^{(V)}(\mathbf{I} - \mathbf{z}_{k,1}[n]\mathbf{z}_{k,1}^T[n]))\Psi_{k,1}[n], \\ \psi_{w,k}[n+1] \leftarrow \mathbf{w}_{k,1}[n] + \mu_{k,1}^{(w)}\gamma\mathbf{z}_{k,1}[n]g(\mathbf{w}_{k,1}^T[n]\mathbf{z}_{k,1}[n]), \end{cases} \quad (12)$$

where $\mathbf{I}$ is the identity matrix, $\Psi_{V,k}$, $\psi_{w,k}$ are the agent's local estimates for the matrix $\mathbf{V}_{k,1}$ and the vector $\mathbf{w}_{k,1}$, with sizes $M_1 \times M_1$ and $M_1 \times 1$, respectively; $\mu_{k,1}^{(\cdot)}$ are the update steps of the gradients; $g(y) = G'(y) = y \exp(-\frac{y^2}{2})$ is the derivative of the non-linearity, and γ is a parameter set to $+1$ for subgaussian sources and -1 for supergaussian sources. Although γ can be estimated online, this work assumes it is known.

For the ATC combination step in the first stage, the agents share and combine their local estimates with neighboring nodes. The proposed method of this work, which reduces the rate and the number of parameters shared, is implemented at this stage as follows: while the agents always use their own estimates $\Psi_{V,k}$, $\psi_{w,k}$ to update $\mathbf{V}_{k,1}$ and $\mathbf{w}_{k,1}$, they receive

and combine these with the estimates from neighboring nodes $\Psi_{V,l}$, $\psi_{w,l}$ ($l \in \mathcal{N}_k$) only in multiples of a predefined interval N (i.e., in $n = N, 2N, \dots$). Also, for $\Psi_{V,l}[n]$, each agent randomly selects a predefined subset of size M'_1 from its M_1 rows to be shared, reducing the total number of shared parameters from $M_1 + M_1^2$ to $M_1 + M'_1 M_1$. The combine step that represents the described rules is performed as:

$$\begin{cases} \mathbf{V}_{k,1}[n+1] \leftarrow \sum_{l \in \mathcal{N}_k} \hat{\mathbf{a}}_{kl}[n] \Psi_{V,l}[n+1], \\ \mathbf{w}_{k,1}[n+1] \leftarrow \sum_{l \in \mathcal{N}_k} \hat{a}_{kl}[n] \psi_{w,l}[n+1], \end{cases} \quad (13)$$

where $\hat{\mathbf{a}}_{kl}[n]$ and $\hat{a}_{kl}[n]$ are the combination weights that depend on the time index n . For updating $\mathbf{V}_{k,1}$, $\hat{\mathbf{a}}_{kl}[n]$ is a $M_1 \times M_1$ diagonal matrix with elements $\hat{a}_{kl}^{(i)}[n]$, where i stands for the row $i \in 1, \dots, M_1$. For $l = k$, the matrix represents each agents weights regarding their own estimates, and it is simply $\hat{\mathbf{a}}_{kk}[n] = a_{kk} \mathbf{I}$, with $\hat{a}_{kl}^{(i)}[n] = a_{kk}$ for every n . Whereas for $l \neq k$, $\hat{\mathbf{a}}_{kl}[n]$ captures the weights for combining the shared estimates from the neighbors $l \in \mathcal{N}_k$. In this case, if n is not a multiple of N , no combination is performed, resulting in $\hat{\mathbf{a}}_{kl}[n] = \mathbf{0}$ (the zero matrix); otherwise, the diagonal elements of $\hat{\mathbf{a}}_{kl}[n]$ are $\hat{a}_{kl}^{(i)}[n] \in \{0, a_{kl}\}$, with $\hat{a}_{kl}^{(i)}[n] = 0$ if the i -th row is not one of the M'_1 selected to be shared from neighbor l , and $\hat{a}_{kl}^{(i)}[n] = a_{kl}$ if it is. For updating $\mathbf{w}_{k,1}$, the connection weight $\hat{a}_{kl}[n]$ is given by: for $l = k$, $\hat{a}_{kk}[n] = a_{kk}$ for every n , and for $l \neq k$, $\hat{a}_{kl}[n] = a_{kl}$ if n is a multiple of N ; otherwise, $\hat{a}_{kl}[n] = 0$. For both $\hat{\mathbf{a}}_{kl}[n]$ and $\hat{a}_{kl}[n]$, the underlying weights a_{kl} are the same as the calculated connection weights in (5), and at each n , the algorithm has to linearly scale these weights to ensure the convex properties $\sum_{l \in \mathcal{N}_k} \hat{\mathbf{a}}_{kl}[n] = \mathbf{I}$ and $\sum_{l \in \mathcal{N}_k} \hat{a}_{kl}[n] = 1$.

The first stage filter ($\mathbf{b}_{k,1}^T[n+1] \leftarrow \mathbf{w}_{k,1}^T[n+1] \mathbf{V}_{k,1}[n+1]$) outputs the signal $y_{k,1}$:

$$y_{k,1}[n] = \mathbf{b}_{k,1}^T[n+1] \mathbf{x}_k[n], \quad (14)$$

that is fed as input to the second stage, which is performed independently by each agent as:

$$\begin{cases} \mathbf{z}_{k,2}[n] \leftarrow \mathbf{V}_{k,2}[n] \mathbf{y}_{k,1}[n], \\ \mathbf{V}_{k,2}[n+1] \leftarrow (\mathbf{I} + \mu_{k,2}^{(V)} (\mathbf{I} - \mathbf{z}_{k,2}[n] \mathbf{z}_{k,2}^T[n])) \mathbf{V}_{k,2}[n], \\ \mathbf{w}_{k,2}[n+1] \leftarrow \mathbf{w}_{k,2}[n] + \mu_{k,2}^{(w)} \gamma \mathbf{z}_{k,2}[n] g(\mathbf{w}_{k,2}^T[n] \mathbf{z}_{k,2}[n]), \end{cases} \quad (15)$$

where $\mu_{k,2}^{(\cdot)}$ are the update steps. Throughout the cascade, the output signal is an estimation of the source $y_k = \mathbf{b}_{k,2} * y_{k,1} = (\mathbf{b}_{k,2} * \mathbf{b}_{k,1}) * x_k$.

For both stages, the matrices $\mathbf{V}_{k,\cdot}$ and the vector $\mathbf{w}_{k,\cdot}$ can be initialized as the identity matrix $\mathbf{V}_{k,\cdot}[0] = \mathbf{I}$ and the centered-spike $\mathbf{w}_{k,\cdot}[0] = (0, \dots, 1, \dots, 0)^T$.

In the next section, the simulations with AN-CMA and AN-SCICA-S are presented.

IV. SIMULATIONS AND ANALYSIS

Throughout the simulations, the intersymbol interference (ISI) is considered for the performance evaluation of the

algorithms. The ISI is a measure of how close the channel-equalizer combined impulse response $\Lambda_k[n]$ (9) is to a distortionless system. The ISI_{avg} is calculated over the agents in the network across iteration n as:

$$\text{ISI}_{\text{avg}}[n] \text{ (dB)} = 10 \log \left(\frac{1}{K} \sum_{k=0}^{K-1} \text{ISI}_k[n] \right), \quad (16)$$

where K denotes the number of agents and $\text{ISI}_k[n]$ is the calculated average ISI for the k -th agent at iteration n . The $\text{ISI}_k[n]$ is obtained by considering $T = 15$ Monte Carlo repetitions of the experiment:

$$\text{ISI}_k[n] = \frac{1}{T} \sum_{t=1}^T \frac{\sum_{i=1}^M |\Lambda_{k,i}^t[n]|^2 - |\Lambda_k^t[n]|_{\max}^2}{|\Lambda_k^t[n]|_{\max}^2}, \quad (17)$$

where $\Lambda_k[n]$ is calculated by (9), $\Lambda_{k,i}^t[n]$ denotes its i -th coefficient at the t -th Monte Carlo experiment, and $|\Lambda_k^t[n]|_{\max}$ is its maximum absolute value coefficient.

The ring topology shown in Fig. 2a is used for the multitasking ANs. The agents have connections with two neighbors, and all connection weights are calculated by the Averaging Rule [1], which results in $a_{kl} = \frac{1}{3}$. The first stage of the equalizer is chosen to be of size $M_1 = 7$, while the second is $M_2 = 11$. The two-stage system provides an equivalent equalization filter $\mathbf{b}_k$ with $M = M_1 + M_2 - 1 = 17$ taps. For comparison, the simulations also consider networks with a non-cooperative topology in which $a_{kk} = 1$ and $a_{kl} = 0$ ($l \neq k$), and each agent estimates a single filter of size $M = 17$. This non-cooperative model corresponds to a classical stand-alone implementation of the CMA and SCICA algorithms across agents.

It considers that the agents feature different AWGN noise levels $E\{v_k^2\}$ and that the sources s_k have the same variance. The Signal-To-Noise Ratio (SNR) for each agent, shown in Fig. 2b, is calculated as $\text{SNR}_k \text{ (dB)} = 10 \log_{10} \left(\frac{E\{s_k^2\}}{E\{v_k^2\}} \right)$, and the whole network has an average noise variance corresponding to $\text{SNR}_{\text{avg}} = 20$ dB.

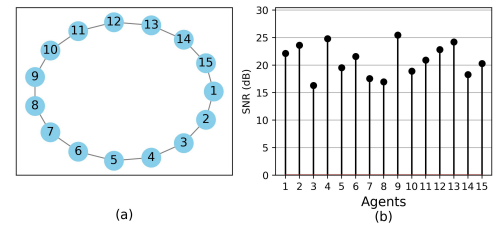


Fig. 2. Network with 15 agents and $\text{SNR}_{\text{avg}} = 20$ dB a) topology b) SNR_k distribution across agents.

The simulations are carried out with the following update steps; for the AN-SCICA and AN-SCICA-S: $\mu_{k,1}^{(V)} = \mu_{k,2}^{(V)} = 4 \times 10^{-3}$ and $\mu_{k,1}^{(w)} = \mu_{k,2}^{(w)} = 7 \times 10^{-3}$; for the AN-CMA, the update steps are $\mu_{k,1} = \mu_{k,2} = 7 \times 10^{-3}$. The stand-alone implementation (non-cooperative) of the SCICA and the CMA has only one filter, and they use the same update steps as their AN counterparts; that is $\mu_k^{(V)} = 4 \times 10^{-3}$ and $\mu_k^{(w)} = 7 \times 10^{-3}$ for the SCICA, and $\mu_k = 7 \times 10^{-3}$ for the CMA.

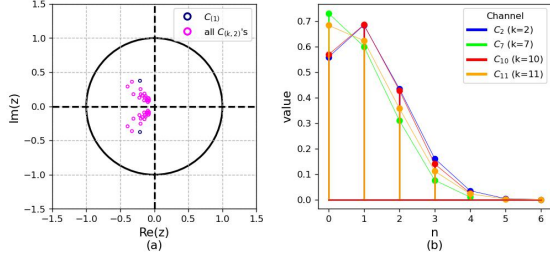


Fig. 3. Scenario 1: simulated channels a) Zeros of $C_k = C_1 * C_{k,2}$'s in the Z-Transform domain for all agents b) Example of the impulse responses for the channels C_k 's with $k = 2, 7, 10$, and 11 .

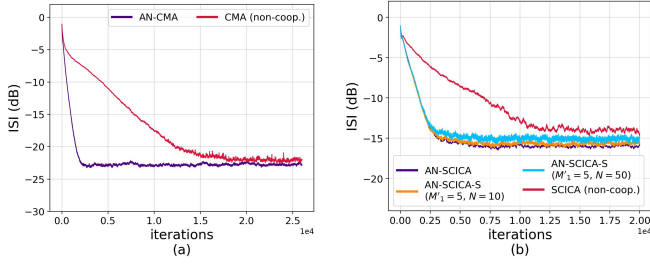


Fig. 4. Scenario 1: ISI_{avg} results a) CMA based algorithms with BPSK sources b) SCICA based algorithms with Laplace distributed sources.

The simulations and their results are presented next in two scenarios concerning the phase-characteristics of the channels and source distributions.

A. Scenario 1 - Minimum Phase Channels

Considering minimum phase channels, CMA based algorithms were simulated with BPSK sources using samples $s_k[n] \in \{-1, +1\}$, while the source samples for the SCICA based algorithms are derived from a standard Laplace distribution, which represents a more challenging scenario. Fig. 3a shows the zeros of the channels C_k , as defined in (3), in the Z-Transform domain, where the pair of zeros of C_1 (dark blue) is common among all agents, while the pair of zeros of $C_{k,2}$ (magenta) differs for each agent. Fig. 3b presents some examples of the impulse response of such channels.

The simulation results are shown in Fig. 4. Fig. 4a reveals that the AN-CMA multitask greatly outperformed the non-cooperative CMA in the speed of convergence while achieving a slightly better level of ISI_{avg} . The results in Fig. 4b indicate that AN-SCICA-S maintained satisfactory performance, comparable to that of AN-SCICA, while reducing the number of shared rows to $M'_1 = 5$ and exchanging data only in multiples of $N = 10$ and $N = 50$. The non-cooperative SCICA presented the slowest convergence and slightly worse ISI_{avg} .

B. Scenario 2 - Nonminimum Phase Channels

Considering nonminimum phase channels, with their zeros shown in Fig. 5a, the sources were sampled from a standard uniform distribution. Fig. 5b presents the simulation results for the ISI_{avg} . For the CMA algorithms, the difference in performance is notable since the non-cooperative CMA didn't

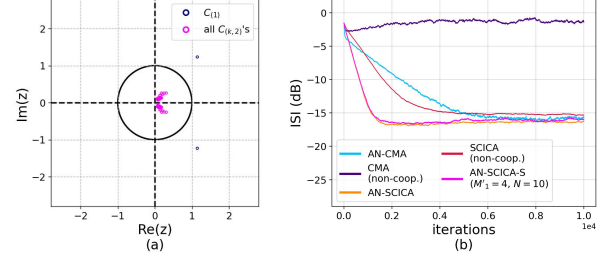


Fig. 5. Scenario 2: Simulation channels and ISI_{avg} results a) Zeros of $C_k = C_1 * C_{k,2}$'s in the Z-Transform domain for all agents b) CMA and SCICA based algorithms with standard uniform distributed sources.

converge, while the multitask AN-CMA achieved a satisfactory ISI_{avg} level. As with the SCICA-based algorithms, both cooperative network implementations outperform the non-cooperative SCICA in terms of convergence speed and residual ISI_{avg} level. The AN-SCICA-S with $M'_1 = 4$ and $N = 10$ presented a convergence curve closer to that of the AN-SCICA. Compared with AN-CMA, the AN-SCICA (-S) had faster convergence rates and slightly lower ISI_{avg} levels.

V. CONCLUSION

This work proposes extending the AN-CMA to the multitask setting, consisting of an equalizer with two cascaded filters and, within the same context, the AN-SCICA-S, which includes a simple mechanism to reduce data exchange among agents in the network compared to AN-SCICA. The simulation results suggest that the AN-CMA multitask is advantageous over the standalone CMA algorithm. The AN-SCICA-S also maintained good performance, even with the reduced information exchange among agents, and was superior to the standalone SCICA implementation. For future work, a theoretical analysis of the algorithms' convergence in this multitasking setting shall be derived.

REFERENCES

- [1] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing: Volume 3*, A. M. Zoubir, M. Viberg, R. Chellappa, and S. Theodoridis, Eds. Elsevier, 2014, pp. 323–453.
- [2] D. G. Tinglea, R. Candido, and M. T. M. Silva, "Adaptive diffusion networks: An overview," *Signal Processing*, vol. 223, p. 109570, 2024.
- [3] G. Depieri and A. Neves, "Single channel ICA for blind deconvolution over adaptive networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 12, pp. 260–269, 2026.
- [4] A. Moussa, M. Poulighen, M. Frikel, S. Bedoui, K. Abderrahim, and M. M'Saad, "An overview of blind equalization algorithms for digital communications," in *Proc. IEEE ICASSP*. Sousse, Tunisia: IEEE, March 2019, pp. 491–496.
- [5] M. Davies and C. James, "Source separation using single channel ICA," *Signal Processing*, vol. 87, no. 8, pp. 1819–1832, 2007.
- [6] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4, pp. 411–430, 2000.
- [7] K. Yuan, J. Zhuo, W. Gao, L. Zhang, and J. Chen, "Diffusion constant modulus algorithm for blind equalization," in *2022 IEEE International Conference on Signal Processing, Communications and Computing (IC-SPCC)*, 2022, pp. 1–6.
- [8] Y. Liu and Y. Cai, "Distributed blind equalization in networked systems," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4296–4300.
- [9] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.