# Distributed Universal Adaptive Networks

Cassio G. Lopes , *Senior Member, IEEE,* Vítor H. Nascimento, *Senior Member, IEEE,* Luiz F. O. Chamon, *Member, IEEE*

***Abstract*—Adaptive networks (ANs) are effective real time techniques to process and track events observed by sensor networks and, more recently, to equip Internet of Things (IoT) applications. ANs operate over nodes equipped with collaborative adaptive filters that solve distributively an estimation problem common to the whole network. However, they *do not* guarantee that nodes do not lose from cooperation, as compared to its non-cooperative operation; that poor nodes are rejected and exceptional nodes estimates reach the entire network; and that performance is uniform over all nodes. In order to enforce such properties, this work introduces the concept of distributed universal estimation, which encompasses the new concepts of local universality, global universality and universality with respect to the non-cooperative operation. We then construct a new cooperation protocol that is proven to be distributively universal, outperforming direct competitors from the literature, as shown by several simulations. Mean and mean-square analytical models are developed, with good agreement between theory and simulations.**

***Index Terms*—Adaptive Networks, Distributed Adaptive Processing, Sensor Networks, Internet of Things, Universal Estimation.**

## I. Introduction

In several applications, a network of interconnected agents is in charge of observing events in a field of interest, usually performing event-related tasks. Such events leave a space-time signature that may be registered by a number of sensors properly placed throughout the geographical area where the events take place [1].

Applications of this framework are plentiful: detect a signal of interest; estimate physical quantities, such as temperature, pressure or wind velocity, solar incidence, the position and speed of a target, the spectrum of signals; network synchronization; identify structural failures, among others [2]–[5], [6]–[8]. In such applications, in the absence of a central node, an optimization problem common to the entire network must be solved distributively and cooperation among agents is either mandatory, or very desirable to promote improvement in network performance and robustness. The agents, or nodes, conduct partial processing over local data using low caliber processors and, via limited cooperation with nearby peers, the local results are collectively aggregated into a global solution that, ideally, should achieve the performance of an (hypothetical) central node omniscient of the network data. A subset of such problems is that of *distributed estimation*, which is the scope of this work.

One of the strategies in the literature to promote a collective behavior towards an asymptotic global solution is the *consensus strategy*, which usually consists of a distributed averaging of proper quantities related to the estimation problem, such as sample cross-covariance and auto-covariance, or even direct estimates [5], [9]–[14].

Most applications can be modeled as a vector of parameters to be estimated from the space-time data captured across the network. In this context, the concept of *adaptive networks* (ANs) arose as an effective real time technique to estimate the application-related parameter vector [15], [16], [17], [18]. When the comparison is meaningful, it has been shown that the cooperative strategy conducted by ANs may outperform that of consensus strategies [19]. ANs are the focus of this work.

In addition to data statistics, two factors define performance in ANs: the learning rules at the nodes and the cooperation protocol. The learning rule, such as the least mean-squares (LMS) or the recursive least-squares (RLS) rules [20], [21], is selected according to performance requirements and the local available processor. The cooperation protocol must restrict communications to local interactions only, favoring energy savings. Typical cooperation protocols are the incremental and the diffusion, with all their variants [16]–[18], [22]–[29].

In the incremental cooperation protocol, only one estimate is shared at a time and it is updated over a cycle visiting all the nodes exactly once [16], [22]. This technique allows for extreme energy savings, though it requires the definition of a Hamiltonian cycle across the network (an NP-hard problem) [30] and the network becomes hostage to local node performance, which may vary greatly. Later, a randomized incremental version was proposed that avoided the Hamiltonian cycle and promoted, on average, a good degree of performance uniformity across the network [23].

The diffusion protocol [17], [18], [24]–[29], [31], [32] is usually preferred since it explores more efficiently the available information, also providing a good performance uniformity across the network, although at a higher energy cost, as compared to the incremental protocols. Whichever the diffusion type, at any node the protocol starts by fusing estimates retrieved from nearby nodes. The fusing is obtained via a local function, usually a fixed linear combination of the estimates. This fixed design follows specific rules, such as the uniform and the Metropolis, which are node-degree-dependent convex combiners [17], and the relative-variance rule [33], which further accounts for node noise variance. A learning step follows, in which the fused estimate is injected into the local AF, which updates its estimate in response to the other

C. G. Lopes and V. H. are with the Dept. of Electronics and Systems, Escola Politécnica, University of Sao Paulo, São Paulo, SP, Brasil, {cassio.lopes,vitnasci}@usp.br.

L. F. O. Chamon is with the Excellence Cluster for Simulation Technology of the University of Stuttgart, Stuttgart, Germany, luiz.chamon@simtech.uni-stuttgart.de.

This research was funded by the ELIOT project, São Paulo Research Foundation (FAPESP) 2018/12579-7 and ANR-18-CE40-0030. The work of L. F. O. Chamon is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC 2075-390740016).

node estimates and to its local data [2], [17], [18], [33], [34].

Many works adopt the diffusion protocol and attempt to limit its cooperation complexity, saving energy and communication resources, while avoiding a corresponding network performance deterioration [25], [27], [31], [32]. In [25], a probabilistic selection of a subset of neighbors dramatically decreased the local diffusion cooperation, thus with major energy savings, while maintaining the performance properties to a great extent. Other approaches propose reducing the cooperation load by transmitting compressed versions of the local estimates, as in [32]; or performing a double compression over the local learning and fusion steps before sharing quantities with other nodes [27]. In a similar vein, the scheme in [31] proposes that each node transmits a subset of the local estimate entries to its neighbors at every iteration.

Another important line of work is comprised of selective cooperation policies over diffusion networks [17], [35], [28], [29], [36]–[38]. The idea is not to blindly cooperate; instead, adopt rules that give emphasis to better nodes, or that improve estimation performance by, for example, transmitting only sufficiently novel information.

With the inception of ANs, it was rapidly noticed that adapting the local fusion combiners, assigning larger weights to select better nodes, without discarding the less-fortunate ones, yielded network performance improvement [17], and different selective adaptive policies followed [36]–[38]. Also related are the works [28], [29], which elegantly adopt the classical universal adaptive convex [39] and affine [40] combinations at every node, but whose inputs are two independent and competing generic diffusion ANs; this generates an output AN whose *average* network mean-square performance is guaranteed to be at least as good as the best average network performance between the two input competing ANs.

Despite the improvement in average network mean-square performance achieved by selective diffusion networks, some nodes may be better off working independently when their cooperative and non-cooperative performances are compared [38]. This raises a nontrivial question: when, or how, should a node cooperate or not cooperate? In order to properly answer this question, in this work we depart from the standard standalone universal estimators [41], [42], adopted in [28] and [29] at every node, and develop the concept of *distributed universal networks*. In this new conceptual framework, an adaptive network will be considered universal if every node performs at least as well as the best available standalone node. This simple definition, due to the spatial data diversity, to the cooperation, and to the limited communication among nodes, unfolds into different kinds of universality: local universality, global universality, and universality with respect to (w.r.t.) the non-cooperative operation. These definitions are directly connected with desired properties of distributed adaptive systems, namely: (a) ability to reject bad nodes; (b) promotion of good nodes; (c) node performance homogeneity (See Section III-A). Such properties are advocated here to assess what good performance means in the context of distributed adaptive estimation.

After motivating proper definitions of distributed universality, a cooperation protocol is introduced that guarantees that nodes do not lose from cooperating: their performance will be always at least as good as if they operated individually, but often much better. The core idea is to preserve local estimates, while separately fusing the estimates received from neighboring nodes. A similar idea was presented in [37], in an effort to solely improve performance in heterogeneous networks, with a subsequent improved version in [43]. However, the idea of recasting the distributed estimation problem in a new universal estimation framework was put forward in our preliminary work [38], yielding a less complex and more efficient algorithm. Here, we extend that work in important ways: (a) formalize the aforementioned universality types (In [38] there were two types only); (b) prove that our adaptive distributed algorithm is indeed universal in the new sense; (c) develop analytic mean and mean-square models for our algorithm, for stationary and non-stationary cases; (d) provide comparisons with other relevant time-varying combiners from the literature [17], [36], [37], [43], in which the proposed algorithm stands out as the most efficient and the only one that is truly universal in all aspects.

This paper is organized as follows. Section II covers the fundamentals of adaptive networks, also introducing the main adaptive combiner strategies from the literature. Section III presents the original concept of universal estimation and how it should be upgraded to the distributed case, with a detailed discussion on the required new definitions of universality. In Section IV the universal distributed algorithm is constructed. We prove that, under reasonable conditions, our algorithm is distributed universal. Section V develops analytical models for the mean and mean-square algorithm evolution, also showing that the algorithm is stable and converges to the optimum vector in the mean, under typical conditions. The proposed strategy is then compared in Section VI to the two main competing algorithms from the literature, namely [36] and [43], also showing that the developed analytical mean-square model reasonably agrees with simulations.

*Remarks on notation:* small font letters refer to scalars and vectors, and capital letters to constants and matrices: $\epsilon$ is a scalar regularization factor; $M$ is the local filter order (a constant), and $A$ is the network constant adjacency matrix. We employ subscript indexing to denote time-varying vectors and matrices, and parentheses to describe time-varying scalars: at node $n$, $u_{n,i}$ is a (row) vector that collects the local scalar signal $u_n(i)$; $w_{n,i}$ is a local vector estimate for the network unknown vector $w^o$; and $H_{n,i}$ is a matrix that denotes the local learning rule at time $i$. This is usually clear from the context.

## II. Adaptive networks with adaptive combiners

An adaptive network structure is modeled as an (un)directed graph $\mathcal{G} = (V,E)$, where $V$ is the node set and $E$ is the edge set [30]. Algebraically, it is convenient to represent the network by its adjacency matrix $A$, defined as $[A]_{n\ell} = 1$ if nodes $n$ and $\ell$ are connected, and $[A]_{n\ell} = 0$ otherwise. By definition a node is connected to itself, i.e., $[A]_{nn} = 1$ for all $n \in V$. A connected network has a path connecting every two nodes $n$ and $\ell$. The neighborhood for node $n$ is the set $\mathcal{N}_n$ of nodes that have a direct connection with node $n$, including itself; that is, all the nodes that are at most one hop away from

node $n$. The strict neighborhood $\overline{\mathcal{N}}_n$ of $n$ does not contain node $n$ itself; in other words, $\overline{\mathcal{N}}_n = \mathcal{N}_n \backslash n$.

At time $i$, the $n$-th node has access to a scalar measurement $d_n(i)$ and to another signal $u_n(i)$, that is collected into an $1 \times M$ local row regressor vector[1]

$$u_{n,i} \triangleq [u_n(i) \;\; u_n(i-1) \;\; \ldots \;\; u_n(i-M+1)]. \quad (1)$$

The data model is then defined via a known local function $f_n$, subject to noise $v_n(i)$. Typically, $f_n$ is linear in terms of an unknown $M \times 1$ global vector of parameters $w^o$, i.e.,

$$d_n(i) = f_n[u_{n,i}] + v_n(i) = u_{n,i} w^o + v_n(i), \quad (2)$$

or by a linear-in-the-parameters nonlinear model such as a truncated Volterra series [44].

The goal of the $N$-node network is to estimate the unknown vector $w^o$ from the available space-time data set $\{d_n(i),u_{n,i}\}$, $n = 1,\ldots,N$. Since all nodes have the common goal $w^o$, it makes sense to cooperate, which not only improves overall performance, but may also enforce stability over the distributed adaptive process running at the nodes. For that matter, node $n$ runs a local adaptive filter of the form

$$\psi_{n,i} = \psi_{n,i-1} + H_{n,i} u_{n,i}^T [d_n(i) - u_{n,i}\psi_{n,i-1}]. \quad (3)$$

Equation (3) represents a stand-alone AF running locally and returning at time $i$ an $M \times 1$ estimate $\psi_{n,i}$ for the unknown global vector $w^o$, where $H_{n,i}$ is an $M \times M$ positive definite matrix that defines the local adaptive rule. For a scalar step-size $\mu_n$, the most common choices are $H_{n,i} = \mu_n I$, for the LMS rule; and $H_{n,i} = \frac{\mu_n}{\|u_{n,i}\|^2+\epsilon} I$, for the normalized LMS (NLMS) rule, where $0 < \epsilon \ll 1$ is a small regularization factor [21]. Other rules are also possible.

Associated with node estimates are figures of merit that assess performance. They are inherited from standard adaptive filtering: the mean-square error (MSE), the excess mean-square error (EMSE), and the mean-square deviation (MSD). For node $n$, they are respectively defined as

$$\begin{aligned} \mathrm{MSE}_n(i) &\triangleq E e_n^2(i) = E[d_n(i) - u_{n,i}\psi_{n,i-1}]^2 \\ \mathrm{EMSE}_n(i) &\triangleq E[u_{n,i} w^o - u_{n,i}\psi_{n,i-1}]^2 \\ \mathrm{MSD}_n(i) &\triangleq E\|w^o - \psi_{n,i-1}\|^2. \end{aligned} \quad (4)$$

The error definitions above are *local quantities* and depend on *which estimate is used locally* at the nodes to fulfill their tasks. In (4) it is assumed that the estimate $\psi_{n,i-1}$ is used locally, which is compatible to the case when the AFs evolve independently from other nodes. However, when cooperation takes place, additional definitions are needed (see below).

Cooperation may be achieved by fusing nearby estimates $\{\psi_{\ell,i} \;,\; \ell \in \mathcal{N}_n\}$, in terms of local scalar combiners $c_{n\ell}$ to be designed. The resulting fused estimate $\phi_{n,i}$ is then injected into the learning step, i.e., into the local AF. Collecting the fusion and the learning steps results in the standard Diffusion LMS [15], [17], given at node $n$ by:

$$\phi_{n,i-1} = \sum_{\ell \in \mathcal{N}_n} c_{n\ell} \psi_{\ell,i-1}, \quad (5)$$

$$\psi_{n,i} = \phi_{n,i-1} + H_{n,i} u_{n,i}^T [d_n(i) - u_{n,i}\phi_{n,i-1}] \; . \quad (6)$$

[1]The regressor may also have a more general structure. For instance, in adaptive antennas, $u_{n,i} = [u_{n,1}(i) \;\; u_{n,2}(i) \;\; \ldots \;\; u_{n,M}(i)]$ [20].

Notice that the fusion step (5) aggregates space-time information from the neighborhood and tends to be a (much) better estimate for $w^o$ than $\psi_{n,i-1}$ in (3). Subsequently, $\phi_{n,i-1}$ is used as an initial condition at time $i$ in the learning step (6), so that the local AF responds not only to its previous local estimate $\psi_{n,i-1}$, but also to those of its neighbors[2]. This is at the heart of the concept of an adaptive network: although the local node processes only local data, the fusion step couples local learning with nearby nodes. As every node $n$ proceeds the same way, the entire network adapts in real-time in order to track $w^o$ cooperatively, and in a fully distributed manner, from the observed space-time data $\{d_n(i), u_{n,i}\}$.

The aggregate estimate $\phi_{n,i-1}$ in (5) can be interpreted as a weighted least-squares estimate of $w^o$ given the received estimates $\{\psi_{\ell,i-1}\}$. This implies that the set $\{c_{n\ell} \geq 0\}$ must be convex (i.e., satisfy $\sum_{\ell \in \mathcal{N}_n} c_{n\ell} = 1$) in order for the estimates to be unbiased [17], [18]. Several simple topology-dependent designs have been proposed, such as the uniform rule $c_{n\ell} = \frac{1}{|\mathcal{N}_n|}$, where $|\mathcal{N}_n|$ is the degree (number of connections) of node $n$; or the Metropolis rule, which employs, for nodes $n$ and $\ell$

$$c_{n\ell} = \begin{cases} 1/\max(|\mathcal{N}_n|,|\mathcal{N}_\ell|), & \text{if } n \neq \ell \text{ are linked;} \\ 0, & \text{for } n \text{ and } \ell \textit{ not} \text{ linked;} \\ 1 - \sum_{\ell \in \overline{\mathcal{N}}_n} c_{n\ell}, & \text{for } n = \ell. \end{cases} \quad (7)$$

Another rule is based on relative variance [33], which assigns $c_{n\ell}$ to be inversely proportional to the noise variance $\sigma_{v,n}^2$:

$$c_{n\ell} = \begin{cases} \frac{\sigma_{v,n}^{-2}}{\sum_{\ell \in \mathcal{N}_n} \sigma_{v,\ell}^{-2}}, & \text{if } \; n \neq \ell \;\; \text{are} \;\; \text{linked;} \\ 0, & \text{for } n \text{ and } \ell \;\; \textit{not} \;\; \text{linked.} \end{cases} \quad (8)$$

Such combiners are typically organized into an $N \times N$ matrix $C = [c_{n\ell}]$, which will be stochastic for the uniform combiner and for (8), i.e., for $\mathbb{1} = \mathrm{col}[1 \;\; 1 \;\; 1 \ldots 1 \;\; 1]$ (with length given by the context), we have that $C\mathbb{1} = \mathbb{1}$. Combiner (7) leads to a doubly stochastic matrix: $\mathbb{1}^T C = \mathbb{1}^T$ and $C\mathbb{1} = \mathbb{1}$ [45].

The limited performance of fixed combiners led to the introduction of adaptive combiners $c_{n\ell}(i)$ that are able to account for network diversity and time-varying statistics [17], [36]–[38], [43]. *Adaptive Diffusion* [17] was inspired by parallel-independent combinations of AFs [39] and weighs local and neighborhood estimates using fixed combiners $\bar{c}_{n\ell}$, that are convex over $\overline{\mathcal{N}}_n$, and *one* adaptive combiner $\lambda_n(i)$ per node. Explicitly,

$$\begin{aligned} \phi_{n,i-1} &= \sum_{\ell \in \overline{\mathcal{N}}_n} \bar{c}_{n\ell} \, \psi_{\ell,i-1}, \\ w_{n,i-1} &= \lambda_n(i) \, \psi_{n,i-1} + [1 - \lambda_n(i)] \phi_{n,i-1}, \\ \psi_{n,i} &= w_{n,i-1} + H_{n,i} u_{n,i}^T (d_n(i) - u_{n,i} w_{n,i-1}), \end{aligned} \quad (9)$$

where the estimate $\phi_{n,i-1}$ fuses estimates $\{\psi_{\ell,i-1}, \ell \in \overline{\mathcal{N}}_n\}$ from the strict neighborhood via fixed combiners $\{\bar{c}_{n\ell}\}$. The local cooperative filter output $w_{n,i-1}$ fuses adaptively the local estimate $\psi_{n,i-1}$ with $\phi_{n,i-1}$ via the adaptive parameter

[2]Note that the error definitions (4) can be given in terms of $\psi_{n,i-1}$ or $\phi_{n,i-1}$, giving rise to the two versions of the diffusion protocol: respectively, combine-then-adapt (CTA) or adapt-then-combine (ATC) [34].

$\lambda_n(i)$, which minimizes the local output error $e_n = d_n(i) - u_{n,i}w_{n,i-1}$ in the mean-square sense.

Later, an alternative approach was taken by adopting $|\mathcal{N}_n|$ different adaptive combiners per node, one per estimate $\psi_\ell$ received from the neighborhood $\mathcal{N}_n$ [36]. It calculates an approximate $|\mathcal{N}_n|\times|\mathcal{N}_n|$ matrix $Q^{\text{MSD}}_{n,i-1}$ to the local (unknown) covariance matrix for the estimate error vector $\psi_{n,i-1}-w^o$, in order to minimize the local $\text{MSD}_n(i)$ w.r.t. the $|\mathcal{N}_n|$ adaptive combiners captured into the local vector $\{c_{n,i}\}$; this algorithm is referred to as the MSD-algorithm (MSD-alg). Similarly to the Adaptive Diffusion above, a locally fused estimate $\phi_{n,i-1}$ is injected into the local AF. Upon receiving the estimates $\{\psi_{\ell,i-1}\}$, $\ell\in\mathcal{N}_n$ from the neighborhood, and letting $S^{\text{MSD}}_n = (I - \frac{\mathbb{1}\mathbb{1}^T}{|\mathcal{N}_n|})$, where $\mathbb{1}$ is $|\mathcal{N}_n|\times 1$, the algorithm at node $n$ becomes

$$\begin{aligned}
\left[Q^{\text{MSD}}_{n,i-1}\right]_{k\ell} &= [\psi_{\ell,i-1}-\psi_{\ell,i-2}]^T[\psi_{k,i-1}-\psi_{k,i-2}], k,\ell\in\mathcal{N}_n,\\
\phi_{n,i-1} &= \sum_{\ell\in\mathcal{N}_n}[c_{n,i-1}]_\ell\psi_{\ell,i-1},\\
\psi_{n,i} &= \phi_{n,i-1} + H_{n,i}u^T_{n,i}(d_n(i)-u_{n,i}\phi_{n,i-1}),\\
c_{n,i} &= c_{n,i-1} - \mu_{M,n}\, S^{\text{MSD}}_n Q^{\text{MSD}}_{n,i-1}c_{n,i-1},
\end{aligned} \tag{10}$$

where matrix $Q^{\text{MSD}}_{n,i-1}$ is $|\mathcal{N}_n|\times|\mathcal{N}_n|$, $[c_{n,i}]_\ell$ is the $\ell$-th element of the combiner vector $c_{n,i}$ and $\mu_{M,n}$ is a scalar step-size that depends on two other parameters $\kappa_M$ and $\epsilon_M$, according to Equations (14) and (18) in [36]. The initial conditions for the algorithm are $\mathbb{1}^Tc_{n,-1}=1$, $[c_{n,i}]_\ell\geq 0$, and $\psi_{n,-1}=\psi_{n,-2}=0_{M\times 1}$. Here too both $\psi_{n,i-1}$ or $\phi_{n,i-1}$ can be used to define the output errors[3].

In [37], a least-squares (LS) adaptive combiner algorithm (LS-alg) was proposed with an important change in the protocol: keep the local AF estimate $\psi_{n,i-1}$ adapting independently from the rest of the network. In other words, a set of $|\overline{\mathcal{N}}_n|$ adaptive combiners collected into a local vector $c_{n,i}$ fuses the local independent AF estimate $\psi_{n,i-1}$ with the strict neighborhood estimates $\{w_{\ell,i-1},\,\ell\in\overline{\mathcal{N}}_n\}$, generating a local estimate $w_{n,i}$ that is ready for use. The procedure is the same as in the original Adaptive Diffusion, with a subtle, yet effective, difference: $w_{n,i}$ is *not* injected into the local AF. The original work had instabilities in the calculation of the combiners, as acknowledged by the authors, so that a stable and better version was published later in [43], and is described below. The identity matrix $I$ is $|\overline{\mathcal{N}}_n|\times|\overline{\mathcal{N}}_n|$ and $\mathbb{1}$ is $|\overline{\mathcal{N}}_n|\times 1$:

$$\begin{aligned}
\tilde{y}_{n,i} &= [\,u_{n,i}(w_{\ell,i-1}-\psi_{n,i-1})\,]_{\ell\in\overline{\mathcal{N}}_n}\quad (|\overline{\mathcal{N}}_n|\times 1)\\
e_n(i) &= d_n(i) - u_{n,i}\psi_{n,i-1}\\
P_{n,i} &= \sum_{p=1}^{i}\gamma_n^{i-p}\,\tilde{y}_{n,p}\tilde{y}^T_{n,p}\;,\quad z_{n,i} = \sum_{p=1}^{i}\gamma_n^{i-p}\,\tilde{y}_{n,p}\\
c_{n,i} &= (P_{n,i}+\epsilon_{LS}I)^{-1}\,z_{n,i}\\
w_{n,i} &= [1-\mathbb{1}^Tc_{n,i}]\,\psi_{n,i-1} + \sum_{\ell\in\overline{\mathcal{N}}_n}[c_{n,i}]_\ell\,w_{\ell,i-1},\\
\psi_{n,i} &= \psi_{n,i-1} + H_{n,i}u^T_{n,i}[d_n(i)-u_i\psi_{n,i-1}],
\end{aligned} \tag{11}$$

where $[c_{n,i}]_\ell$ is again the $\ell$-th element of vector $c_{n,i}$, $0\ll\gamma_n<1$ is a local forgetting factor and $\epsilon_{LS}>0$ ensures invertibility in $P_{n,i}+\epsilon_{LS}I$. The estimate $w_{n,i}$ tends to be better than $\psi_{n,i-1}$, thus it is adopted locally. As such, the error quantities (4) for (11) must be updated in terms of $w_{n,i}$.

The more recent works in [28] and [29] explore directly the original concept of universality at every node. At each iteration, each node runs two independent diffusion process represented by two local adaptive filters whose estimates, $w_{1,n,i}$ and $w_{2,n,i}$, are generated cooperating with nearby nodes and feed a local combiner

$$w_{n,i} = \lambda_1(i)w_{1,n,i} + \lambda_2(i)w_{2,n,i}, \tag{12}$$

whose combiners $\lambda_1(i)$ and $\lambda_2(i)$ are chosen to be either convex [29], or affine [28]. The output estimate $w_{n,i}$ is guaranteed to be classically universal in the mean-square error sense. As such, each node output is at least as good as its two input diffusion processes $w_{1,n,i}$ and $w_{2,n,i}$. The overall result is that the output network average performance is at least as good as the best average network performance between the two input diffusion processes. This important contribution is not universal in any of the proposed universality types introduced in this work. This is because the output estimates $w_{n,i}$ at every node are locally confined, they are not propagated to the neighborhood: there is no network level feedback and no network learning takes place, in the sense introduced in [38].

Extensive simulations show that both algorithms (10) and (11) consistently outperform both the original diffusion LMS [17] and its adaptive version (9). Although effective at improving performance, (10) and (11) are *not* distributed universal algorithms as the one proposed in [38], which is extended and studied in detail in this work.

## III. Distributed Universal Estimation

Our first task is to extend the concepts of universal estimation [41], [42] to the distributed case.

The design of an AF takes place by optimizing some figure of merit, say mean-square error, in terms of a set of parameters $\theta$, which may include the filter order $M$, the step-size $\mu$, the forgetting factor $\gamma$ for the RLS filter, the rank parameter $K$ for the APA filter, etc. Traditionally, such parameters must be designed from not always accurate analytical models that may depend on unknown quantities, so that empirical tests must be made. When the scenario is time-varying, the design is an even more challenging task.

In this context, universal estimation is a change in the design paradigm: instead of choosing a fixed parameter set $\theta$ under limited knowledge, select a pool of $K$ candidates $\{\theta_k\}$ for $\theta$, each of them forming an individual estimator, or expert, and present them to a supervisor, which is ultimately in charge of generating a reasonable estimate $w$ for $w^o$ by consulting the pool of experts. The central question is: what is a reasonable estimate? The answer to that is the very definition of universal estimation: an estimator is considered universal when its supervisor is able to at least match the performance of the best individual expert in the pool, in terms of the adopted figure of merit. Within a class, the pool $\{\theta_k\}$ has to be rich enough to span the unknown optimal $\theta$. Of course, this might represent an explosion in computational complexity,

[3]For the MSD supervisor, we use the latter in Sec. VI, since this usually results in better performance.

so care must be taken to find a trade-off. The supervisor is any function that consults the pool of experts and delivers a universal estimate w.r.t. the adopted figure of merit.

Such ideas have been extensively and successfully explored in the literature in the context of combinations of adaptive filters [39], [46], in terms of pools of filters with different step-sizes, filter orders, or even different learning rules [47]–[51]. The supervisor admits different designs, but one that is widely adopted and is efficient is the convex supervisor, which is an activation function of a free parameter to be adapted [50].

### A. Distributed universality

Typical distributed adaptive systems rely on several nodes consulting multiple data sources across the field of interest, therefore subject to a natural spatial diversity. In general, cooperation among the nodes is desirable, though it has to be implemented under limited communications. The combination of spatial diversity and limited communications may drive different nodes to different performances, which might not be acceptable in most applications. As such, a intuitive set of desired properties for ANs may be defined and will guide the development of a distributed universal protocol to operate in any network:

A) Ability to reject a bad node;
B) Ability to exploit an exceptional node;
C) Node performance homogeneity.

Rejecting a bad node (A) relates to the ability of avoiding using poor estimates that could degrade the performance of other nodes in the network. This is paramount in applications such as remote sensing networks, where sensor damage can degrade the information provided by nodes. Similarly, the ability to exploit estimates from an exceptional node (B) lends robustness to the AN: nodes operating in poor conditions can take advantage of better nodes. Finally, performance homogeneity (C) is fundamental if the network is to operate in a fully distributed manner. Indeed, each node must ultimately rely on its local estimate to perform actions, such as alerting to the presence of an intruder or anomalies in the field [2], [52], or even controlling some environmental variable [1], [24]. In many ways, promoting good performance across all nodes is more important than having better average performance with some nodes (much) worse than others.

In order to address the aforementioned desired properties, three main points must be tackled: define the experts; select figures of merit; and define what is a supervisor when multiple data sources are consulted and cooperation among nodes is a requirement.

In the AN case, the selection of the pool of experts is natural and is comprised of the $N$ adaptive nodes attempting to estimate $w^o$ from the data available across the network. The required expert diversity is guaranteed by the natural space-time data diversity, and/or by employing different AF parameters at the nodes, or even different learning rules.

Typical figures of merit adopted for ANs are global network quantities: average MSE, EMSE and MSD, defined from the local quantities previously presented in (4):

$$\begin{aligned} \mathrm{MSE}(i) &= \frac{1}{N}\sum_{n=1}^{N}\mathrm{MSE}_n(i), \\ \mathrm{EMSE}(i) &= \frac{1}{N}\sum_{n=1}^{N}\mathrm{EMSE}_n(i) \quad \text{(Network quantities)}, \\ \mathrm{MSD}(i) &= \frac{1}{N}\sum_{n=1}^{N}\mathrm{MSD}_n(i). \end{aligned} \tag{13}$$

These metrics provide a valid overall measure of network performance and are widely used in the literature, with or without cooperation. Cooperation does improve network performance according to these metrics, even if not everybody in the network will experience an improvement, as compared to the non-cooperative operation [25], [37]. This implies that we must look carefully into the way nodes cooperate, and should discriminate performance across the network by also inspecting the local quantities $\mathrm{MSE}_n(i)$, $\mathrm{EMSE}_n(i)$ and $\mathrm{MSD}_n(i)$.

One key point of universal estimators is that expert integrity is preserved, which implies that at some level they should work independently. This is because cooperation might lead to the poor performance of some experts to contaminate that of the good ones, and who's whom is application-dependent or may vary over time. In the AN case, this property is usually violated when cooperation is implemented. On the other hand, without cooperation the global performance of ANs is deterred to a great extent; furthermore, in some applications, such as source localization and scalar field estimation applications, nodes *must* cooperate in order to solve the global problem in a distributed manner [4], [8]. Therefore, in order to enforce the desired properties of ANs, cooperation demands that the definition (def.) of universality be extended.

**Def. 1** (Local universality)**.** A node $n$ of an AN is said to be locally universal when it holds that $n$ is at least as good as the best node in its neighborhood, i.e., the best node $m \in \mathcal{N}_n$.

**Def. 2** (Global universality)**.** An AN is said to be globally universal when for all nodes $n \in \{1,\ldots,N\}$ it holds that $n$ is as good as the best node in the network, i.e., the best node $m \in \{1,\ldots,N\}$.

**Def. 3** (Universality w.r.t. the non-cooperative strategy)**.** An AN is said to be universal with respect to the non-cooperative strategy when for all nodes $n \in \{1,\ldots,N\}$ it holds that node $n$ performs at least as well as if it were independent of the rest of the network.

**Def. 4** (Asymptotic universality)**.** An AN is said to be asymptotically universal, when it is universal for $i \to \infty$, i.e., at steady-state.

These definitions are inspired by those found in the contexts of universal prediction [41] and theory of individual sequences [53]. They are straightforward and intuitive, although nontrivial. For instance, Def. 3 might seem obvious, but it is often violated in standard diffusion ANs [17], as is the case of Def. 1.

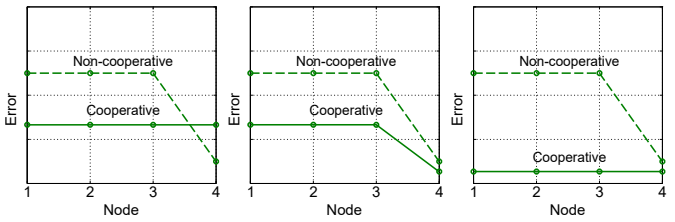



Fig. 1. Distributed asymptotic universality (Def. 4) in ANs: (a) Globally universal (Def. 2) but not universal w.r.t. the non-cooperative strategy (Def. 3); (b) Universal w.r.t. the non-cooperative strategy (Def. 3), but not globally universal (Def. 2); (c) Globally universal (Def. 2) and universal w.r.t. the non-cooperative strategy (Def. 3).

Figure 1 depicts examples in a hypothetical four-node connected network to illustrate the different definitions of universality. Considering asymptotic behavior (Def. 4), it is straightforward to see that local universality (Def. 1) implies both rejection of bad nodes and the exploitation of exceptional ones. Moreover, for a connected AN with an undirected graph, if every node is locally universal, then global universality (Def. 2) follows (if the network is undirected and locally universal, then the performance of each pair of connected nodes must be equal. If the network is connected, there is a path between every pair of nodes; thus the network must be globally universal). Thus, globally universal networks not only guarantee that underperforming nodes are isolated, but also that all nodes take advantage of the performance of superior ones. What is more, global universality guarantees node performance homogeneity. Finally, rejecting poor nodes is clearly related to the concept of universality w.r.t. the non-cooperative strategy (Def. 3), as it requires that cooperation only improves local performance.

Notice that Definitions 2 and 3 denote two different forms of distributed universality. Indeed, it is possible for ANs to be globally universal without being universal w.r.t. the non-cooperative strategy and vice-versa. In fact, global universality alone only leads to performance homogeneity across nodes. Universality w.r.t. the non-cooperative strategy, on the other hand, guarantees that cooperating is the best strategy for each individual node in the network instead of only for the network on average. Both concepts must apply to obtain all the aforementioned properties for ANs. Definition 4 is a realistic definition, in view of the limited communications and energy within the network: in a large network, with a large radius [30], an exceptional node estimate that is several hops away from another given node will take several algorithm cycles to be broadcast network-wide, making universality necessarily an asymptotic property.

### B. The quest for a distributed supervisor

The first point to make is that a single supervisor directly imported from the standard single data source case [39], [46], [50] is not viable to implement distributed universality. This is because in a network with $N$ distributed experts consulting $N$ data sources, the access of the single supervisor to all the expert estimates would require its centralization: nodes report their estimates to the sole supervisor which promptly reports back a universal estimate for local use[4]. Here universality is always guaranteed and its original definition is enough, since the network will access the best estimate and it will be the same for everybody. However, a central node is highly undesirable as it represents a catastrophic failure point for the network and the amount of communication resources (energy, more powerful transceivers, etc.) is prohibitively large; and it might be infeasible simply due to lack of connectivity, either because links are not available, or because routing protocols may impose too much overhead and excessive delays.

A second strategy would be to implement a fully connected network. In this setup, every node may be seen as a central node, and this hints at placing $N$ supervisors, one per node, embedded with the local expert. A given supervisor at node $k$ receives estimates from all the network experts and correctly generates the best estimate for local use. Due to the full connectivity, this may be replicated at all other nodes with their respective local supervisors, so that it is guaranteed that every node will have an estimate that is the best the network has to offer and all estimates will be statistically equivalent, since every node has access to the same set of experts. As a consequence, the original universality concept suffices too in this scheme. However, the implementation of this level of connectivity demands even larger resources than the centralized node case. Thus, it is also impractical.

We are left with the realistic scenario, mentioned earlier in this section: nodes have access only to their nearby peers, with different degrees of connectivity and with multiple data sources that reflect spatial diversity: these facts combined do not guarantee in general that a node will never loose from cooperation, that the network has the ability to reject bad nodes and/or promote exceptional ones, or even that the node performance will be uniform. In this scenario, we establish the main conceptual leap from standard universal estimation: define $N$ supervisors, one per node, and let the *supervisors cooperate*, instead of direct expert cooperation. In summary, we propose that a *distributed supervisor* may be implemented by a set of $N$ collaborative supervisors that shelter their local experts from the outer world. In such realistic scenarios, existing distributed adaptive systems may present, or not, the different kinds of universality formalized earlier. In response to that, in the following section we show in detail how to construct a cooperation protocol that explores the idea of distributed supervision, and that is proven to be universal w.r.t. all the Definitions 1–3.

## IV. A DISTRIBUTED UNIVERSAL COOPERATION PROTOCOL

Why should a node cooperate, if its performance might deteriorate? Should cooperation in the name of a greater good be enough? Such questions lie at the heart of what distributed supervision should deliver in the distributed multisource case.

To begin with, a more inviting cooperation protocol should guarantee that a node never loses from cooperating; its performance at least remains the same, as compared to its

[4]For instance, the MSE for each expert could be calculated and the supervisor would select the estimate corresponding to the expert with the smallest MSE.

independent operation. The idea is that all nodes improve, with the possible exception of the best node in the network, which should not worsen its non-cooperative performance. Due to topology constraints, a generic node is unaware of which, or where, is the best node in the network. If the protocol assures that there will be no losses in performance, then cooperating becomes an interesting deal.

### A. Constructing the protocol

A protocol that promotes distributed universality with respect to the introduced Definitions 1–3 has two steps.

Firstly, notice that the concept of universality w.r.t. the non-cooperative strategy (Def. 3) motivates the idea of protecting local estimates from network perturbations, an idea that was also proposed in [37], albeit for different reasons. Indeed, allowing each node to operate as if it were independent of the network is a simple way to guarantee that its estimation process is not disturbed by underperforming neighbors. This, however, leads to a non-cooperative network. Hence, the nodes need a way to preserve their own estimates, without neglecting those from the rest of the network. This can be implemented much like what was done in the adaptive diffusion scheme (9) [17], in terms of a local independent estimate $\psi_n$ and an estimate $\phi_n$ fused from neighboring supervisor estimates

$$\phi_{n,i-1} = \sum_{\ell\in\overline{\mathcal{N}}_n} \overline{c}_{n\ell} w_{\ell,i-1}, \tag{14a}$$

$$w_{n,i} = \lambda_n(i)\,\psi_{n,i-1} + [1-\lambda_n(i)]\phi_{n,i-1}, \tag{14b}$$

$$\psi_{n,i} = \psi_{n,i-1} + H_{n,i}u_{n,i}^T(d_n(i) - u_{n,i}\psi_{n,i-1}). \tag{14c}$$

Two crucial differences in (14) from the adaptive diffusion (9) stand out: (a) the local estimate $\psi_{n,i}$ evolves independently according to the local learning rule, i.e., $w_{n,i}$ *is not injected* into the local AF [37], [38]; and (b) the local supervisor estimate $w_{n,i-1}$ is shared within its neighborhood, instead of sharing $\psi_{n,i-1}$. Intuitively, $w_{n,i-1}$ tends to be a better estimate than $\psi_{n,i-1}$. Furthermore, in [38] a network learning model was developed that shows how sharing $w_{n,i-1}$ implements a network-level feedback that, hop by hop, allows the best estimates to reach the whole network, while sharing $\psi_{n,i-1}$ not necessarily does. Finally, (14) unveils the need for one supervisor per node that implements cooperation, rather than the local adaptive filters: cooperation is carried out indirectly.

The last resource is an accelerating mechanism that feeds back the supervisor output into the local AF, every $L_n$ iterations [51]; it was already successfully applied in [38]:

$$\psi_{n,a} = \delta_{L_n}(i)\,w_{n,i} + [1-\delta_{L_n}(i)]\,\psi_{n,i-1}, \tag{15}$$

$$\psi_{n,i} = \psi_{n,a} + H_{n,i}u_{n,i}^T[d_n(i) - u_{n,i}\psi_{n,a}], \tag{16}$$

where $\delta_{L_n}(i) = \delta(i - rL_n)$ is the Kronecker delta, with $r \in \mathbb{Z}^+$. Note that an adaptive diffusion iteration, as in (9), is periodically implemented above: within an $L_n$-length cycle, the local AF experiences $L_n - 1$ independent iterations; at the $L_n$-th iteration, it is perturbed with the supervisor estimate $w_{n,i}$ exactly once, so that the local AF has access to the potentially best estimate in its neighborhood. As such, $L_n$ should not be small, since otherwise it violates the principle of preserving the local AF. On the other hand, too large values for $L_n$ do not accelerate the transient, returning to purely isolated local estimates in the limit $L_n \to \infty$ (i.e., transfers never occur). A simple design technique for $L_n$ is imported from [51], and a typical value for AN applications is $L_n = 1{,}000$. We note, however, that a good value for $L_n$ may be larger if the input signals are very correlated or if $M$ is large, such that convergence of the local experts is slow. Conversely, smaller values of $L_n$ might be useful for small $M$ and uncorrelated signals, a situation in which the local experts will converge quickly. In Section VI we show in Example 6 (See Fig. 13) that the network performance is relatively insensitive within a wide range for $L_n$.

We now collect all the equations into a complete *distributed universal adaptive network* with a generic learning rule at the nodes. Upon selecting the local learning rule via matrix $H_{n,i}$, and the reception of the supervisor estimates $\{w_{\ell,i-1}\}$ from the neighborhood, the proposed algorithm implemented at node $n$ is:

$$\lambda_n(i) = \frac{1}{1+e^{-a_n(i-1)}}, \tag{17a}$$

$$\check{\lambda}_n(i) = \begin{cases} \lambda_n(i), & \text{if } -a_+ < a_n(i) < a_+, \\ 0, & \text{if } a_n(i) = -a_+, \\ 1, & \text{if } a_n(i) = a_+. \end{cases} \tag{17b}$$

$$\phi_{n,i-1} = \sum_{\ell\in\overline{\mathcal{N}}_n} \overline{c}_{n\ell}\, w_{\ell,i-1}, \tag{17c}$$

$$w_{n,i} = \check{\lambda}_n(i)\,\psi_{n,i-1} + [1-\check{\lambda}_n(i)]\phi_{n,i-1}, \tag{17d}$$

$$e_n(i) = d_n(i) - u_{n,i}w_{n,i}, \tag{17e}$$

$$p_n(i) = \nu_n p_n(i-1) + [1-\nu_n]\,|u_{n,i}(\psi_{n,i-1} - \phi_{n,i-1})|^2, \tag{17f}$$

$$\widetilde{\mu}_{a,n} = \mu_{a,n} \,/\, [p_n(i) + \epsilon_p], \tag{17g}$$

$$a_n(i) = \big[a_n + \widetilde{\mu}_{a,n}u_{n,i}(\psi_n - \phi_n)e_n(i)\lambda_n[1-\lambda_n]\big]_{-a_+}^{a_+}, \tag{17h}$$

$$\psi_{n,a} = \delta_{L_n}(i)\,w_{n,i} + [1-\delta_{L_n}(i)]\,\psi_{n,i-1}, \tag{17i}$$

$$\psi_{n,i} = \psi_{n,a} + H_{n,i}u_{n,i}^*[d_n(i) - u_{n,i}\psi_{n,a}], \tag{17j}$$

where in (17h) $a_n = a_n(i-1)$, $\lambda_n = \lambda_n(i)$, $\psi_n = \psi_{n,i-1}$ and $\phi_n = \phi_{n,i-1}$.

In the algorithm above, (17a) is a convex activation function that represents the supervisor parameter, which is adapted in terms of the auxiliary variable $a_n(i)$ [54]. Eq. (17b) implements a truncation operation, either ceiling $\lambda_n(i)$ to 1, or flooring it to 0, depending on the limiting parameter $a_+$ (Typically $a_+ = 4$); this results in a smaller variance for the random variable $\check{\lambda}_n(i)$, also accelerating convergence [54]. The neighborhood supervisor estimates are fused into $\phi_{n,i-1}$ in (17c), which is used to generate the local supervisor output $w_{n,i}$ in (17d) for local use, with the associated estimation error $e_n(i)$ in (17e). The quantity $p_n(i)$ is a normalization factor, with the associated filtering parameter $0 \ll \nu_n < 1$ and regularization parameter $0 < \epsilon_p \ll 1$, that helps improving the convergence of the parameter $a_n(i)$; this also has the effect of considerably limiting the required range for $\mu_{a,n}$ in (17g), which typically can be chosen in the interval $(0, 1]$ when

normalization is adopted [54]. Equation (17h) is the actual update recursion for $a_n(i)$ and it drives $\lambda_n(i)$ in (17a).

Since (17) above evolved from the adaptive diffusion protocol, the fixed combiners $\{\bar{c}_{n\ell} \geq 0\}$ must also be convex over the strict neighborhood $\overline{\mathcal{N}}_n$, that is $\sum_\ell \bar{c}_{n\ell} = 1$ for $0 \leq n, \ell \leq N$, and $c_{n\ell} = 0$ for $\ell \neq \overline{\mathcal{N}}_n$. For that matter, the Uniform, (7) and (8) rules may be adopted to design $\{\bar{c}_{n\ell}\}$, mutatis mutandis. The Universal Adaptive Supervisor (17) is referred to as the *U-sup* algorithm. As with the LS-alg (11), the best available estimate at node n is $w_{n,i}$, thus the error definitions for U-sup must be updated w.r.t $w_{n,i}$. Besides performance, the computational complexity is of central importance in IoT and sensor network applications, and here we consider the number of multiplications per node per iteration $N_\times$ as the metric for comparison, disregarding the local AF operations (which are essentially the same for all distributed algorithms considered here).[5] The number of multiplications required for implementing the Universal supervisor algorithm (U-sup) (17), the Mean-Square Deviation combiners algorithm (MSD-alg) (10) and the Least-Squares combiners algorithm (LS-alg) (11), respectively, are

$$\begin{aligned} N_\times(\text{U-sup}) &= \big(|\mathcal{N}_n| + 3\big)M + 7 \\ N_\times(\text{MSD-alg}) &= \frac{\big(|\mathcal{N}_n|^2 + 3|\mathcal{N}_n| + 2\big)}{2}M + 2|\mathcal{N}_n|^2 \\ N_\times(\text{LS-alg}) &= 2\big(|\mathcal{N}_n| + 1\big)M + |\mathcal{N}_n|^3/3 + |\mathcal{N}_n|^2 + |\mathcal{N}_n| \,. \end{aligned} \tag{18}$$

The complexity of all algorithms depends, obviously, on the application, which is captured by $M$; but also depends on the network topology, captured by $|\mathcal{N}_n|$ (which assumes the algorithms explore the entire neighborhood at each node). For a given application, which means a fixed $M$, if the network infrastructure is enlarged for performance improvement, the U-sup complexity scales linearly with $|\mathcal{N}_n|$; the LS-alg has a linear term on $|\mathcal{N}_n|M$, but which is twice as complex as the corresponding term in U-sup, and has a cubic term $|\mathcal{N}_n|^3/3$ that is application independent; and the MSD-alg scales quadratically with $|\mathcal{N}_n|$, and has another quadratic term $2|\mathcal{N}_n|^2$ that is also application independent. As a numerical example, consider **Example 1** in Section VI: a network of $N = 15$ nodes, with $M = 50$ for the AFs, and average node degree of $|\mathcal{N}_n| = 6$, returns 457 multiplications for the U-sup algorithm, 814 for the LS-alg, and 1472 for the MSD-alg. For the same example, increasing node degree to $|\mathcal{N}_n| = 10$, the U-sup will require 657 multiplications (44% increase), 1543 for the LS-alg (90% increase) and 3500 for the MSD-alg (138% increase).

### *B. Universality of the proposed protocol*

Showing that algorithm (17) achieves the Definitions 1–3 of distributed universality is intricate, since (17) is a set of stochastic coupled nonlinear recursions; in particular, the recursions for the local supervisors $\lambda_n(i)$ are coupled due to the sharing of information across the network. In this section, we show that any steady-state solution to (17) must achieve universal performance according to Definitions 1–3, under three assumptions:

- A.1 The network is at steady-state, that is, all local filters have converged to their final MSD performance, and the supervisors have also converged to their final values (see Sec. V for a discussion about convergence);
- A.2 The local supervisors (17b), (17d) choose the best option (in terms of MSD) between $\psi_{n,i}$ and $\phi_{n,i}$ at steady-state;
- A.3 The local filters are independent, i.e., $L_n \to \infty$.

Note that Assumption A.2 was also used in [39] to prove that the convex combination scheme is universal. This assumption is justified since using [49, eq. (11) and (17)] it can be shown that the supervisor weight for the convex combination scheme minimizes the combination MSE if $\mu_a \to 0$. Minimization of MSE is equivalent to minimization of the MSD when the regressors $u_{n,i}$ are white [20]. Note also that Assumption A.2 is related to local properties of the local supervisors, and is thus not equivalent to assuming network universality. Assumption A.3 is equivalent to requiring that $L_n$ is large enough so that the local filters and supervisors have time to converge before a decision about transfer of coefficients is made.

In the next section we present a model for the transient behavior of (17) in the mean and mean-square senses, but the study of the limiting behavior of the resulting model is still considerably difficult, and is left for a future work.

We now explore the steady-state properties of the proposed AN distributed estimator (17). We show in the next two theorems that the proposed scheme is universal w.r.t. the non-cooperative strategy, and that, if a network reaches steady-state (constant MSDs at each node), then necessarily the supervisor leads to global universality.

**Theorem 1** (Universality w.r.t. the non-cooperative strategy)**.** *Under Assumptions A.1–A.3, the network feedback protocol described in* (17) *is asymptotically universal w.r.t. the non-cooperative strategy (Def. 3).*

*Proof:* From equation (17d), the output $w_{n,i}$ of node $n$ is a linear combination between the non-cooperative local estimate $\psi_{n,i-1}$ and the averaged estimates from its neighborhood $\phi_{n,i-1}$. Notice that the non-cooperative strategy is a particular case of (17) in which $\lambda_n(i) = 1$, for all $i$. Thus, since the linear combiner $\lambda_n(i)$ minimizes the local $\text{MSD}_n(i)$, the output of each node $w_{n,i}$ is guaranteed to be at least as good as its non-cooperative version. If the local estimate $\psi_{n,i-1}$ is better than $\phi_{n,i-1}$, then the supervisor $\lambda_n(i)$ will drive $w_{n,i}$ to at least the local performance; on the other hand, if $\phi_{n,i-1}$ is better, then $\lambda_n(i)$ will guide the local node output $w_{n,i}$ to the average neighborhood performance, which is better than the local non-cooperative by hypothesis. Therefore, node $n$ never loses from cooperating, thus it is universal w.r.t. the non-cooperative case (Definition 1). ■

**Theorem 2** (Asymptotic global universality)**.** *The network feedback protocol from (17) is globally universal (Definitions 2 and 4) under Assumptions A.1–A.2.*

*Proof:* Suppose that the estimates $w_{n,i}$ computed by

[5] Nevertheless, they still play a role in the computations for cooperation strategies.

each node resulted in different values of local $\mathrm{MSD}_n(i) = \mathrm{E}\,\|w^o - w_{n,i}\|^2$ at steady state. We show next that this leads to a contradiction. Assume then that the network is at steady-state and node $n_0$ has the worst performance of all nodes, that is, $\mathrm{MSD}_{n_0}(i) \geq \mathrm{MSD}_n(i)$ for all $n \neq n_0$ and $\mathrm{MSD}_{n_0}(i) > \mathrm{MSD}_\ell(i)$ for at least one $\ell \in \overline{\mathcal{N}}_{n_0}$ (such a node has to exist, since the number of nodes is finite and we are assuming that not all local MSDs are equal). Therefore, we have, at steady-state,

$$\begin{aligned}\mathrm{MSD}_{n_0}(i) &= \mathrm{E}\,\|w_{n_0,i} - w^o\|^2 \geq \mathrm{E}\,\|w_{n,i} - w^o\|^2 \\ &= \quad \mathrm{MSD}_n(i) \text{ for } n \in \overline{\mathcal{N}}_{n_0},\end{aligned} \tag{19}$$

with $\mathrm{MSD}_{n_0}(i) > \mathrm{MSD}_\ell(i)$ for at least one $\ell \in \overline{\mathcal{N}}_{n_0}$. Now, at steady-state $\mathrm{MSD}_n(i) = \mathrm{MSD}_n(i-1)$, and from (17c)

$$\begin{aligned}&\mathrm{E}\,\|\phi_{n_0,i-1} - w^o\|^2 = \mathrm{E}\Big\|\sum_{\ell\in\overline{\mathcal{N}}_{n_0}} \overline{c}_{n_0\ell} w_{\ell,i-1} - w^o\Big\|^2 \\ &\leq \mathrm{E}\sum_{\ell\in\overline{\mathcal{N}}_{n_0}} \overline{c}_{n_0\ell}\|w_{\ell,i-1} - w^o\|^2 = \sum_{\ell\in\overline{\mathcal{N}}_{n_0}} \overline{c}_{n_0\ell}\,\mathrm{MSD}_\ell(i-1),\end{aligned} \tag{20}$$

where we used the fact that $\|\cdot\|^2$ is a convex function and the $\{\overline{c}_{nn_0}\}$ add up to one. Since by hypothesis $n_0$ is the worst node, and at least one node in its neighborhood has a smaller MSD, we conclude that necessarily

$$\mathrm{E}\,\|\phi_{n_0,i-1} - w^o\|^2 \leq \sum_{\ell\in\overline{\mathcal{N}}_{n_0}} \overline{c}_{n_0\ell}\,\mathrm{MSD}_\ell(i-1) < \mathrm{MSD}_{n_0}(i-1). \tag{21}$$

The last inequality results from the hypotheses that $\sum_{\ell\in\overline{\mathcal{N}}_{n_0}} \bar{c}_{n_0\ell} = 1$, that $\mathrm{MSD}_n \leq \mathrm{MSD}_{n_0}$, and that there is $\ell_0 \in \overline{\mathcal{N}}_{n_0}$ such that $\mathrm{MSD}_{\ell_0} < \mathrm{MSD}_{n_0}$. This means that the supervisor for node $n_0$ should change its choice to reduce the MSD, contradicting Assumptions A.1-A.2. We conclude therefore that at steady-state all nodes must have the same performance. ■

## V. Performance Analysis

In this section we propose a model for the mean and mean-square performance of the new universal diffusion strategy. Let us start by introducing some additional assumptions. We extend our data model (2), now allowing for changes in the vector of unknown parameters:

$$d_n(i) = u_{n,i} w_{i-1}^o + v_n(i), \tag{22}$$

where $w_{i-1}^o \in \mathbb{R}^{M\times 1}$ is a time-varying vector of unknown parameters the network is trying to estimate, $v_n(i)$ is an i.i.d. zero-mean measurement noise with variance $\sigma_{v,n}^2$, independent of all regressor vectors $\{u_{\ell,i}\}$ in the network. The initial condition $w_{-1}^o$ is a random unit norm vector. We further assume that

- A.4 $\{u_{n,i}\}$ is a zero-mean i.i.d. sequence with covariance matrix $R_n$;
- A.5 $L_n \equiv L$ is the same for all nodes in the network;
- A.6 The optimum parameter vector $w^o$ may change according to a random walk model

$$w_i^o = w_{i-1}^o + q_i, \tag{23}$$

where $\{q_i\}$ is an i.i.d. vector sequence with zero mean and autocovariance matrix $Q = \mathrm{E}\,q_i q_i^T$;

- A.7 The stepsizes $\mu_n$ and $\mu_{a,n}$ are small enough for the usual slow adaptation approximations in adaptive filtering to be valid, and such that the variance of $a_n(i)$ can be disregarded [20], [21], [44], [54];
- A.8 The forgetting factors $\nu_n$ are close to one, so that the variance of $p_n(i)$ can be disregarded;
- A.9 $\check{\lambda}_n(i) = \lambda_n(i)$ always in (17b).

The discussion below assumes, for simplicity, that the local adaptive filters in all nodes are using the same algorithm, either LMS or NLMS. It would not be difficult to modify the models and arguments for other types of filters, or even for networks running different classes of filters at each node. Assumptions A.4, A.6, A.7 and A.8 are widely used in the literature [20], [21], [54]. Assumption A.5 is used only to simplify the analysis and can be easily relaxed. Assumption A.9 is used to simplify the model and will tend to increase the model variances.

### A. The global adaptive network model

We proceed by defining the local error vector quantities for each node as

$$\widetilde{\psi}_{n,i} = w_i^o - \psi_{n,i}, \qquad \widetilde{w}_{n,i} = w_i^o - w_{n,i}. \tag{24}$$

Next, collect the local quantities defined in Algorithm (17) into global variables:

$$\begin{aligned}&\widetilde{\psi}_i = \mathrm{col}\left(\widetilde{\psi}_{n,i}\right), \quad \widetilde{w}_i = \mathrm{col}\left(\widetilde{w}_{n,i}\right), \quad v_i = \mathrm{col}\left(v_n(i)\right),\\ &e_i = \mathrm{col}\left(e_n(i)\right), \quad U_i = \mathrm{diag}\left(u_{n,i}\right), \quad \mathcal{M}_i = \mathrm{diag}\left(H_{n,i}\right),\\ &a_i = \mathrm{col}\left(a_n(i)\right), \quad G = C \otimes I_M, \quad \mathcal{M}_a = \mathrm{diag}\left(\mu_{a,n}\right),\\ &p_i = \mathrm{col}\left(p_n(i)\right), \quad \bar{\nu} = \mathrm{diag}\left(\nu_n\right) \quad \xi_i = \mathbb{1}_N \otimes q_i.\end{aligned}$$

A set of equations describing the evolution of the entire network can then be obtained as follows.

Let $c_n^T$ be the $n$-th row of the combining matrix $C$ and $G_n = c_n^T \otimes I_M$ (the $n$-th block-row of $G$). Under A.9, the equations for the overall network can be written in terms of the error vectors as

$$\Lambda_i = \mathrm{diag}\left(\frac{1}{1+e^{-a_n(i-1)}}\right), \quad \mathcal{L}_i = \Lambda_i \otimes I_M, \tag{25}$$

$$\widetilde{w}_i = \mathcal{L}_i\widetilde{\psi} + (I_{MN} - \mathcal{L}_i)\,G\widetilde{w} + \xi_i, \tag{26}$$

$$\widetilde{\psi}_i = \begin{cases}\widetilde{w}, & \text{if } \delta_{L_n}(i) = 1,\\ \left(I_{MN} - U_i^T\mathcal{M}_i U_i\right)\widetilde{\psi} - U_i^T\mathcal{M}_i v_i + \xi_i, & \text{otherwise.}\end{cases} \tag{27}$$

$$e_i = U_i\widetilde{w}_i + v_i - U_i\xi_i, \tag{28}$$

$$p_i = \bar{\nu}p_{i-1} + (I_N - \bar{\nu})\,\mathrm{col}\left(\left|u_{n,i}\left(G_n\widetilde{w} - \widetilde{\psi}_n\right)\right|^2\right), \tag{29}$$

$$\mathcal{M}_{a,i} = \mathcal{M}_a\,\mathrm{diag}\left(\frac{1}{p_n(i)+\epsilon_p}\right), \tag{30}$$

$$a_i = \Big[a + \mathcal{M}_{a,i}\Lambda_i\,(I-\Lambda_i)\,\mathrm{diag}\left(u_{n,i}\left(G_n\widetilde{w} - \widetilde{\psi}_n\right)\right)^T e_i\Big], \tag{31}$$

where $\widetilde{\psi} = \widetilde{\psi}_{i-1}$, $\widetilde{\psi}_n = \widetilde{\psi}_{n,i-1}$ and $\widetilde{w} = \widetilde{w}_{i-1}$: some iteration indexes will be omitted in the sequel whenever necessary, and

the brackets $[a_i] = [a_i]_{-a_+}^{a_+}$ in (31) constrain each entry of vector $a_i$ to the interval $[-a_+, a_+]$. To obtain (26), we used the facts that $q_i$ is the same across the network, and that $\sum_{\ell \in \overline{\mathcal{N}}_n} c_{n\ell} = 1$.

### *B. Analysis in the mean*

Assume that the variance of $a_n(i-1)$ is small enough so that

$$\Lambda_i \approx \bar{\Lambda}_i = \operatorname{diag}\left(\bar{\lambda}_n(i)\right) \triangleq \operatorname{diag}\left(\frac{1}{1+e^{-\operatorname{E} a_n(i-1)}}\right). \quad (32)$$

This assumption is reasonable if $\mu_a$ is small (Assumption A.7), or when the $\{a_n(i-1)\}$ are close to their limits at $\pm a_+$ [49]. Under Assumption A.8, we can approximate $p_i \approx \operatorname{E}\{p_i\}$, and the evolution of $\bar{a}_i \triangleq \operatorname{E}\{a_i\}$ is given by (to simplify the following argument we introduce the auxiliary variable $\check{a}_i$ as below)

$$\begin{aligned}\check{a}_i &= \bar{a} + \overline{\mathcal{M}}_a \bar{\Lambda}\left(I-\bar{\Lambda}\right) \operatorname{E}\left\{\operatorname{diag}\left(u_{n,i}\left(G_n\widetilde{w}-\widetilde{\psi}_n\right)\right)^T e_i\right\},\\ \bar{a}_i &= [\check{a}_i]_{-a_+}^{a_+},\end{aligned} \quad (33)$$

where $\bar{\Lambda} = \bar{\Lambda}_i$ and $\overline{\mathcal{M}}_a = \overline{\mathcal{M}}_{a,i} \triangleq \operatorname{E}\{\mathcal{M}_{a,i}\}$ will be evaluated further on.

Replacing

$$e_i = U_i\left(\mathcal{L}_i\widetilde{\psi}_{i-1} + (I_{MN}-\mathcal{L}_i)\,G\widetilde{w}_{i-1}\right) + v_i - U_i\xi_i$$

into (33), and recalling that $v_i$ and $\xi_i$ are independent of all other variables, we obtain

$$\begin{aligned}\check{a}_i &= \bar{a} + \overline{\mathcal{M}}_a\bar{\Lambda}\left(I-\bar{\Lambda}\right)\operatorname{E}\left\{\operatorname{col}\left[\left(G_n\widetilde{w}-\widetilde{\psi}_n\right)^T\right.\right.\\ &\quad\left.\left.\times u_{n,i}^T u_{n,i}\left(\bar{\lambda}_n(i)\widetilde{\psi}_{n,i-1} + \left(1-\bar{\lambda}_n(i)\right)G_n\widetilde{w}\right)\right]\right\}\\ &= \bar{a} + \overline{\mathcal{M}}_a\bar{\Lambda}\left(I-\bar{\Lambda}\right)\operatorname{col}\left[\operatorname{Tr}\left(R_n\left(\bar{\lambda}_n(i)\operatorname{E}\{\widetilde{\psi}_n\widetilde{w}^T\}G_n^T\right.\right.\right.\\ &+ \left(1-\bar{\lambda}_n(i)\right)G_n\operatorname{E}\{\widetilde{w}\widetilde{w}^T\}G_n^T - \bar{\lambda}_n(i)\operatorname{E}\{\widetilde{\psi}_n\widetilde{\psi}_n^T\}\\ &\left.\left.\left. - \left(1-\bar{\lambda}_n(i)\right)G_n\operatorname{E}\{\widetilde{w}\widetilde{\psi}_n^T\}\right)\right)\right].\end{aligned} \quad (34)$$

We see that for the evaluation of $\bar{a}_i$ we must find

$$T_i \triangleq \operatorname{E}\{\widetilde{w}_i\widetilde{w}_i^T\}, \quad S_i \triangleq \operatorname{E}\{\widetilde{w}_i\widetilde{\psi}_i^T\}, \quad K_i \triangleq \operatorname{E}\{\widetilde{\psi}_i\widetilde{\psi}_i^T\}, \quad (35)$$

from which all expected values in (33) can be obtained directly as follows. Note that $\widetilde{\psi}_{n,i-1} = \left(b_{(n)}^T \otimes I_M\right)\widetilde{\psi}_{i-1}$, where $b_{(n)}$ is the $n$-th $N\times 1$ canonical basis vector. Denoting $B_n = b_{(n)}^T \otimes I_M$, we can rewrite (33) as

$$\begin{aligned}\bar{a}_i = \bar{a} + \overline{\mathcal{M}}_a\bar{\Lambda}\left(I-\bar{\Lambda}\right)\operatorname{col}\left[\operatorname{Tr}\left(R_n\left(\bar{\lambda}_n B_n S^T G_n^T + \right.\right.\right.\\ \left.\left.\left.\left(1-\bar{\lambda}_n\right)G_nTG_n^T - \bar{\lambda}_n B_n K B_n^T - \left(1-\bar{\lambda}_n\right)G_nSB_n^T\right)\right)\right],\end{aligned} \quad (36)$$

with $\bar{\lambda}_n = \bar{\lambda}_n(i)$, $K = K_{i-1}$, $S = S_{i-1}$ and $T = T_{i-1}$. Note that a recursion for $\overline{\mathcal{M}}_{a,i}$ can be also obtained from $T_{i-1}$, $S_{i-1}$ and $K_{i-1}$ as

$$\overline{\mathcal{M}}_{a,i} = \operatorname{E}\{\mathcal{M}_{a,i}\} \approx \mathcal{M}_a \operatorname{diag}\left(\frac{1}{\operatorname{E}\{p_n(i)\}+\epsilon_p}\right), \quad (37)$$

with $\bar{p}_n(i) \triangleq \operatorname{E}\{p_n(i)\}$ and

$$\begin{aligned}\bar{p}_n(i) &= \nu_n\bar{p}_n(i-1) + (1-\nu_n)\operatorname{E}\left\{\left|u_{n,i}\left(G_n\widetilde{w}-\widetilde{\psi}_n\right)\right|^2\right\}\\ &= \nu_n\bar{p}_n(i-1) + (1-\nu_n)\operatorname{Tr}\left\{R_n\left(G_nT_{i-1}G_n^T\right.\right.\\ &\quad\left.\left. - G_nS_{i-1}B_n^T - B_nS_{i-1}^TG_n^T + B_nK_{i-1}B_n^T\right)\right\}.\end{aligned} \quad (38)$$

The covariance matrices $T_i$, $S_i$ and $K_i$ can be obtained from the autocorrelation matrix of the vector $\Theta_i = \operatorname{col}\left(\widetilde{w}_i, \widetilde{\psi}_i\right)$, and will be dealt with in Section V-C. A recursion for $\Theta_i$ can be obtained from (25)–(31) as follows.

$$\Theta_i = \begin{bmatrix}(I_{MN}-\mathcal{L}_i)G & \mathcal{L}_i\\ 0 & I - U_i^*\mathcal{M}_iU_i\end{bmatrix}\Theta + \begin{bmatrix}\xi_i\\ \xi_i - U_i^T\mathcal{M}_iv_i\end{bmatrix}, \quad (39)$$

with $\Theta = \Theta_{i-1}$. Using again the assumption that the variance of $a_n(i)$ is small (A.7), the mean of (39) becomes

$$\bar{\Theta}_i \triangleq \operatorname{E}\{\Theta_i\} = \underbrace{\begin{bmatrix}(I_{MN}-\bar{\mathcal{L}}_i)G & \bar{\mathcal{L}}_i\\ 0 & I-R_\mu\end{bmatrix}}_{\triangleq F_i}\bar{\Theta}_{i-1}, \quad (40)$$

where $\bar{\mathcal{L}}_i = \bar{\Lambda}_i \otimes I_M$, and $R_\mu$ depends on the particular algorithm used at each node. Assuming all nodes use LMS, we have $R_\mu = \operatorname{diag}(\mu_nR_n)$. If the nodes use NLMS and the regressors $u_{n,i}$ are tap-delay lines, we can use the approximation $R_\mu \approx \operatorname{diag}\left(\frac{\mu_n}{M\sigma_{u,n}^2}R_n\right)$, where $\sigma_{u,n}^2 = E\{u_n^2(i)\}$ [55]. Of course, other algorithms can also be used with their appropriate models, and the algorithms need not be equal for all nodes.

We next study the stability and convergence of (40)[6]. Note first that without transfer of coefficients (i.e., for $L\to\infty$) the recursion for $\operatorname{E}\{\widetilde{\psi}_i\}$ is linear and uncoupled from that for $\operatorname{E}\{\widetilde{w}_i\}$, so $\operatorname{E}\{\widetilde{\psi}_i\}$ converges to zero if and only if $\rho(I-R_\mu) < 1$, where $\rho(\cdot)$ denotes the spectral radius, or equivalently if $0 < \mu_nR_n < 2I$ [20]. The stepsizes must then satisfy

$$0 < \mu_n < \begin{cases}2/\lambda_{\max}(R_n), & \text{for LMS, or}\\ 2, & \text{for NLMS.}\end{cases} \quad (41)$$

where $\lambda_{\max}(R_n)$ represents the largest eigenvalue of $R_n$. Therefore, for small enough stepsizes we can guarantee that $\operatorname{E}\{\widetilde{\psi}_i\} \to 0$. This should be no surprise, since the adaptive filters in each node are running independently of each other.

To prove stability and convergence in the mean of the entire scheme, note that $F_i$ in (40) is block-diagonal, and we just saw that under (41) the lower diagonal block corresponds to a stable recursion. We therefore must now show that the spectral radius of the upper diagonal block, i.e., $(I_{MN}-\bar{\mathcal{L}}_i)G$ is less than one. This can be accomplished by showing that there is an induced norm such that $\|(I_{MN}-\bar{\mathcal{L}}_i)G\| < 1$ (since any induced norm upper bounds the spectral radius of a matrix [56]).

For that we use the block-maximum norm [34]. The block-maximum norm of a length-$MN$ vector is defined as follows: partition a length-$MN$ vector into length-$M$ blocks as $x =$

[6]This recursion is *not* linear, since $\bar{\mathcal{L}}_i$ depends on the autocorrelation (not the autocovariance) of $\Theta_i$, and therefore depends also on its mean, $\bar{\Theta}_i$.

$\mathrm{col}(x_1,\ldots,x_N)\in\mathbb{R}^{MN}$. Then the block maximum norm is defined as

$$\|x\|_{b,\infty}\triangleq\max_{1\le n\le N}\|x_n\|, \tag{42}$$

where from now on $\|\cdot\|$ denotes the Euclidean norm. For block matrices an induced norm based on (42) is defined the usual way: let $A\in\mathbb{R}^{MN\times MN}$ be also partitioned into $M\times M$ blocks $A_{n,\ell}$ and define [34]

$$\|A\|_{b,\infty}\triangleq\max_{\|x\|_{b,\infty}\le 1}\|Ax\|_{b,\infty}. \tag{43}$$

Let us now evaluate $\|(I_{MN}-\bar{\mathcal{L}}_i)G\|_{b,\infty}$: From the definitions of $\bar{\mathcal{L}}_i=\bar{\Lambda}_i\otimes I_M$ and $G=C\otimes I_M$ and for any vector $x\in\mathbb{R}^{MN}$ with $\|x\|_{b,\infty}\le 1$, we have

$$\begin{aligned}(I_{MN}-\bar{\mathcal{L}}_i)G\begin{bmatrix}x_1\\ \vdots\\ x_N\end{bmatrix}&=\left[C\otimes I_M-(\bar{\Lambda}_iC)\otimes I_M\right]\begin{bmatrix}x_1\\ \vdots\\ x_N\end{bmatrix}\\&=\begin{bmatrix}\sum_{\ell=1}^N(1-\bar{\lambda}_1(i))c_{1,\ell}x_\ell\\ \vdots\\ \sum_{\ell=1}^N(1-\bar{\lambda}_N(i))c_{N,\ell}x_\ell\end{bmatrix}\\&=\mathrm{col}\left\{(1-\bar{\lambda}_n(i))\sum_{\ell=1}^N c_{n,\ell}x_\ell\right\}.\end{aligned} \tag{44}$$

Note that each one of the blocks in (44) satisfies

$$\begin{aligned}\left\|(1-\bar{\lambda}_n(i))\sum_{\ell=1}^N c_{n,\ell}x_\ell\right\|&\le\left|1-\bar{\lambda}_n(i)\right|\sum_{\ell=1}^N c_{n,\ell}\|x_\ell\|\\&\le(1-\bar{\lambda}_n(i))\sum_{\ell=1}^N c_{n,\ell}=1-\bar{\lambda}_n(i),\end{aligned}$$

where we used the facts that (a) $\|x\|_{b,\infty}\le 1\Rightarrow\|x_\ell\|\le 1$, $1\le\ell\le N$; (b) $0<\bar{\lambda}_n(i)<1$, so $\left|1-\bar{\lambda}_n(i)\right|=1-\bar{\lambda}_n(i)$; (c) $\sum_{\ell=1}^N c_{n,\ell}=1$.

Recalling that the $a_n(i)$ are restricted to the interval $[-a_+,a_+]$, the $\bar{\lambda}_n(i)$ will stay in the interval

$$\bar{\lambda}_n(i)\in\left[\frac{1}{1+e^{a_+}},\frac{1}{1+e^{-a_+}}\right],$$

and

$$0<1-\bar{\lambda}_n(i)\le 1-\frac{1}{1+e^{a_+}}\triangleq\eta<1.$$

We conclude that $\|(I-\bar{\mathcal{L}}_i)Gx\|_{b,\infty}\le\eta<1$ for all $x\in\mathbb{R}^{MN}$ with $\|x\|_{b,\infty}\le 1$. This means that $\rho((I-\bar{\mathcal{L}}_i)G)\le 1-\eta<1$, and thus $\bar{\Theta}_i$ converges exponentially fast to the origin, that is, the proposed scheme converges in the mean whenever the step-sizes $\mu_n$ are chosen so that all node filters are stable.

### C. Analysis in the mean square

Multiplying (39) by its transpose and taking expectations, we obtain recursions for $T_i$, $S_i$ and $K_i$. The recursion for $K_i$ is a standard result for LMS or NLMS filters, since the local filters are operating independently of each other. We restrict ourselves to the cases of LMS and NLMS only to keep the argument more concise — if some nodes use different algorithms (such as RLS), the corresponding models from the literature can be substituted [20], [21]. Defining $\mathcal{Q}=\mathrm{E}\,\xi_i\xi_i^T=Q\otimes\mathbb{1}\mathbb{1}^T$, for LMS we have

$$K_i=\begin{cases}K-R_\mu K-KR_\mu+\mathrm{diag}\left(\mathrm{Tr}(\mu_nR_nK)R_n\right)\\ \quad+2R_\mu KR_\mu+\mathrm{diag}\left(\mu_n^2\sigma_{v,n}^2R_n\right)+\mathcal{Q},\ \text{if }\delta_L(i)=1\\ T_{i-1},\qquad\qquad\text{if }\delta_L(i)=0.\end{cases} \tag{45}$$

whereas an approximate model for NLMS is the recursion [55]

$$K_i=\begin{cases}K\ -\ R_\mu K\ -\ KR_\mu\ +\ R_\mu KR_\mu\\ +\ \mathrm{diag}\left(\frac{\mu_n^2}{M(M-2)\sigma_{u,n}^4}\sigma_{v,n}^2R_n\right)+\mathcal{Q},\quad\text{if }\delta_L(i)=1\\ T_{i-1},\qquad\qquad\text{if }\delta_L(i)=0\end{cases} \tag{46}$$

recalling that here we denote $K=K_{i-1}$.

For $T_i$ and $S_i$, we obtain

$$\begin{aligned}T_i=&\left(I_{MN}-\bar{\mathcal{L}}\right)GTG^T\left(I_{MN}-\bar{\mathcal{L}}\right)+\bar{\mathcal{L}}S^TG^T\left(I_{MN}-\bar{\mathcal{L}}\right)\\&+\left(I_{MN}-\bar{\mathcal{L}}\right)GS\bar{\mathcal{L}}+\bar{\mathcal{L}}K\bar{\mathcal{L}}+\mathcal{Q},\end{aligned} \tag{47}$$

$$S_i=\begin{cases}\left(I_{MN}-\bar{\mathcal{L}}\right)GS\left(I_{MN}-R_\mu\right)\\ \quad+\bar{\mathcal{L}}K\left(I_{MN}-R_\mu\right)+\mathcal{Q}\qquad\text{if }\delta_L(i)=1\\ T_{i-1}\qquad\text{if }\delta_L(i)=0.\end{cases} \tag{48}$$

where $\bar{\mathcal{L}}=\bar{\mathcal{L}}_i$.

A model for the overall algorithm is obtained running (45)–(48) and (36)–(38) sequentially. Given the highly nonlinear nature of the problem, we leave a stability analysis of the recursion for a future work.

## VI. Simulations

In this section we study the three adaptive combiners described earlier: the MSD-alg algorithm, given by (10); the LS-alg algorithm, defined in (11); and the proposed U-sup algorithm, collected in (17).

For ease of reference, we repeat some definitions here, and introduce others. The input signals $\{u_n(i)\}$ are zero-mean Gaussian sequences generated according to

$$u_n(i)=\beta_nu_n(i-1)+\sqrt{1-\beta_n^2}x_n(i), \tag{49}$$

in which $x_n(i)$ is a Gaussian i.i.d. zero-mean signal with unit variance $\sigma_{x,n}^2=1$, and $-1<\beta_n<1$ is the correlation factor. The measured signal $d_n(i)$ is generated according to the data model (22). The measurement noise $v_n(i)$ is a Gaussian i.i.d. sequence whose variance $\sigma_{v,n}^2$ is adjusted at the nodes to achieve the SNR profiles, randomly selected, that are presented in each simulation example.

Both stationary and non-stationary scenarios are considered, so that the $M\times 1$ time-varying unknown plant $w_i^o$ follows the random walk model defined in (23), with initial condition $w_{-1}^o=\frac{1}{\sqrt{M}}\mathbb{1}$, and $q_i$ is a zero-mean, i.i.d. Gaussian vector process, with covariance matrix $Q=\sigma_q^2I_M$. For stationary plants, $\sigma_q^2=0$ and $w_i^o=w^o$. The adaptive network in charge of tracking such a plant is comprised of $N=15$ nodes equipped with local NLMS filters also with order $M=50$, and step-sizes randomly selected as either $\mu_n=0.1$, or $\mu_n=0.01$ (except for Example 5), with regularization $\epsilon=10^{-6}$.

We design scenarios so as to explore the desired properties for ANs discussed in Section III-A, and to conclude on the

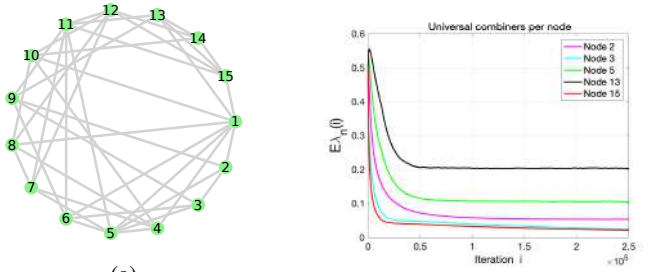
universality of the algorithms. The adopted metrics are the local $\mathrm{MSD}_n(i)$ and network $\mathrm{MSD}(i)$ mean-square deviations (check (4) and (13)), defined in terms of $\psi_{n,i-1}$ for the non-cooperative case, in terms of $\phi_{n,i-1}$ for the MSD-alg case, and in terms of $w_{n,i}$ for the U-sup and LS-alg. We only present a few U-sup combiners $\{\mathrm{E}\,\lambda_n(i)\}$, in order to promote picture clarity; information on combiners for the other algorithms may be obtained directly in [36] and [43].

There are three different error figures: the transient curves, represented by $\mathrm{MSD}(i)$ as a function of the iterations; the steady-state curves, given by $\mathrm{MSD}_n(\infty)$ versus the node index; and a robustness curve, which depicts global MSD quantities versus the non-stationarity parameter $\sigma_q^2$, obtained as follows. The minimum local MSD across the network, $\min_n \mathrm{MSD}_n(\infty)$, is considered for the non-cooperative case; the distributed algorithms are represented by their maximum local MSD, i.e., $\max_n \mathrm{MSD}_n(\infty)$; in other words, the worst node in each algorithm must be equal or better than the best non-cooperative node. Such pictures show how sensitive the distributed algorithms are with respect to how rapidly the plant evolves. Notice that universality may be inferred from both the steady-state and the robustness MSD curves.

The algorithms are compared in fair scenarios, with their parameters optimized for the set of studied cases. Namely, U-sup uses $\mu_{a,n} = \mu_a = 0.005$ (in Example 4, $\mu_a = 5{\cdot}10^{-4}$) and feedback cycle $L_n = L$, with either $L = 800$, or no transfer of coefficients ($L \to \infty$), $\nu_n = 0.9$ and $\epsilon_p = 0.01$. The LS-alg uses $\epsilon_{LS} = 10^{-8}$ and $\gamma_n = \gamma = 0.9999$; in Example 1 we add one LS-alg curve with $\gamma = 0.99$ for comparison, as suggested by the authors [43].[7] The MSD-alg step-size $\mu_M$ is implemented with $\kappa_M = 10^{-5}$ (this choice resulted in better performance than the value $\kappa_M = 0.8$ used in [36]) and $\epsilon_M = 10^{-3}$ (See (14) and (18) in [36]).

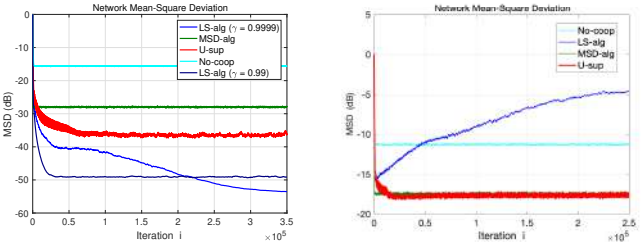
In **Example 1**, we test the ability of the algorithms in rejecting a low-SNR at a well connected node. Node 1 has SNR = $-4.9$ $dB$ and a strict node degree $\mathcal{N}_1 = 8$ (the most connected). The $N$ inputs $\{u_n(i)\}$ are white, the unknown vector $w^o$ is stationary at first and the SNR and the stepsizes across the nodes are SNR=$[-4.9, 11.8, 19.1, 15.7, 16.4, 14.5, 13.8, 15.9, 11.7, 12.1, 11.6, 11.1, 18.9, 14.6, 18.1]$ and $\mu_k = 0.1 \cdot [1, 10, 1, 1, 1, 10, 1, 10, 1, 10, 1, 1, 1, 10, 1, 1]$. The network topology and the adaptive combiner mean evolution $\mathrm{E}\lambda_n(i)$ for our universal AN are presented in Figs. 2-(a) and (b). This is a worst case scenario for the AN: a node subject to a high noise level will produce poor estimates, that will rapidly spread since Node 1 is well-connected. Fig. 3-(a) shows the network $\mathrm{MSD}(i)$ evolution for a stationary plant and Fig. 3-(b) depicts the algorithms tracking a non-stationary plant following model (23), with $\sigma_q^2 = 5 \cdot 10^{-6}$. Figure 3-(a) shows how the LS-alg can perform very well in some cases where the plant is either stationary, or varies very slowly, and can be an option if its extra computational complexity is not a problem; the

[7]We tested $\gamma \in \{0.99, 0.999, 0.9999\}$ in order to obtain the best LS-alg performance. For all the examples included here, $\gamma = 0.9999$ resulted in the best error levels at steady-state, while $\gamma = 0.99$ presented better transient, but with a considerable degradation after convergence. Different levels for $\epsilon_{LS}$ were also considered, although without relevant impact on performance.



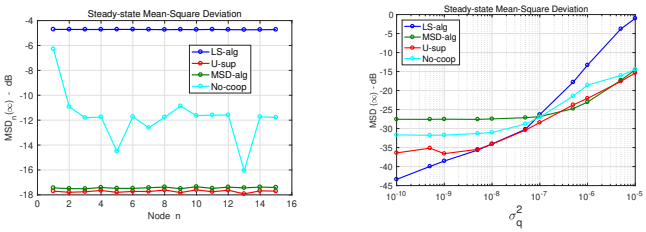

(a) (b)

Fig. 2. **Example 1 (White inputs)**: (a) Network topology; (b) The mean adaptive combiners $\mathrm{E}\lambda_n(i)$ corresponding to Fig. 3-(a).




(a) (b)

Fig. 3. **Example 1 (White inputs)**: (a) Network $\mathrm{MSD}(i)$ for stationary plant; (b) Network $\mathrm{MSD}(i)$ for a random walk plant with $\sigma_q^2 = 5 \cdot 10^{-6}$.

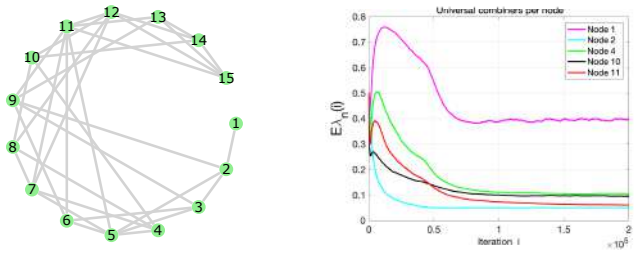



(a) (b)

Fig. 4. **Example 1 (White inputs)**: (a) Steady-state $\mathrm{MSD}_n(\infty)$ for $\sigma_q^2 = 5 \cdot 10^{-6}$ corresponding to Fig. 3-(b); (b) Tracking robustness in terms of $\mathrm{MSD}(\infty)$ versus $\sigma_q^2$: $\min_n \mathrm{MSD}_n(\infty)$ for the non-cooperative case and $\max_n \mathrm{MSD}_n(\infty)$ for the cooperative algorithms.

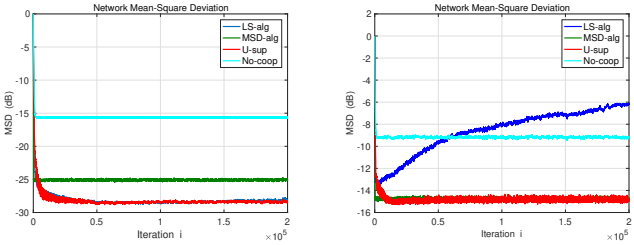
dark blue LS-alg curve uses $\gamma = 0.99$, presenting a faster transient, while the blue LS-alg curve uses $\gamma = 0.9999$ and attains a better steady-state performance. However, when the plant varies faster, as happens in Fig. 3-(b), the LS-alg cannot keep up, presenting a major drop in performance, while the U-sup and the MSD-alg perform much better. Figure 4-(a) presents the steady-state performance $\mathrm{MSD}_n(\infty)$ at the node level for $\sigma_q^2 = 5 \cdot 10^{-6}$, and one can see that both U-sup and MSD-alg are universal, while the LS-alg is not. Figure 4-(b) depicts the robustness of all three methods: for slowly varying plants, up to a certain degree, the LS-alg performs better than the other two algorithms. However, as the plant starts evolving faster, LS-alg degrades more rapidly than the other algorithms, becoming worse than the non-cooperative case beyond $\sigma_q^2 = 5{\cdot}10^{-8}$, therefore losing universality. The U-sup is the only algorithm that outperforms the non-cooperative case across the entire $\sigma_q^2$ test range, also when $\sigma_q^2 \to 0$ (tested, but not shown).

In **Example 2**, we test the network ability to recruit and propagate the exceptional Node 1, which is poorly connected (one connection only). This is the opposite scenario of Example 1. Figs 5-(a) and (b) respectively depict the network topology and the mean combiners



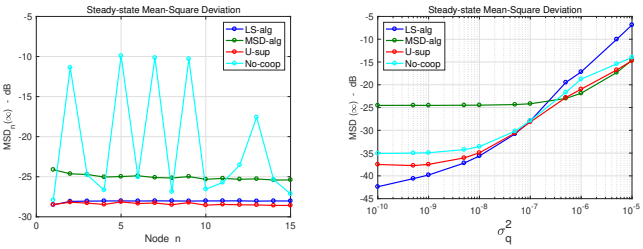

(a) (b)

Fig. 5. **Example 2 (White inputs)**: (a) Network topology; (b) The mean adaptive combiners $\mathrm{E}\lambda_n(i)$ corresponding to Fig.6-(a).




(a) (b)

Fig. 6. **Example 2 (White inputs)**: (a) Network $\mathrm{MSD}(i)$ for a non-stationary plant with $\sigma_q^2 = 10^{-7}$; (b) Network $\mathrm{MSD}(i)$ when $\sigma_q^2 = 10^{-5}$.

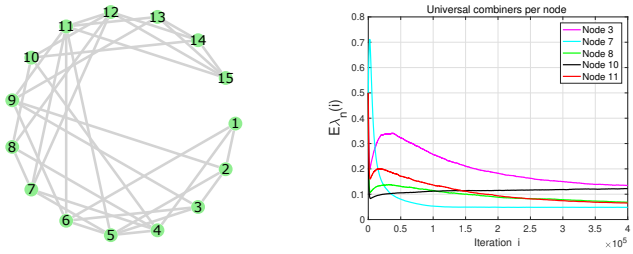



(a) (b)

Fig. 7. **Example 2 (White inputs)**: (a) Steady-state $\mathrm{MSD}_n(\infty)$ for Fig. 6-(a); (b) Tracking robustness in terms of $\mathrm{MSD}(\infty)$ versus $\sigma_q^2$: $\min_n$ $\mathrm{MSD}_n(\infty)$ for the non-cooperative case and $\max_n$ $\mathrm{MSD}_n(\infty)$ for the cooperative algorithms.

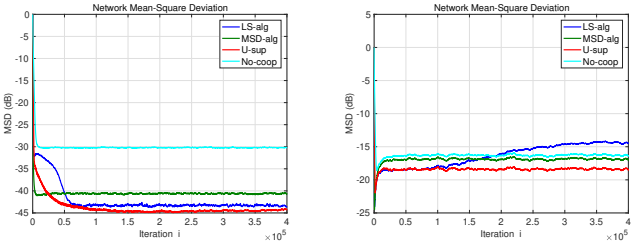
$\mathrm{E}\lambda_n(i)$ for the U-sup algorithm corresponding to the network MSD depicted in Fig.6-(a). The input data $\{u_n(i)\}$ is white and the plant $w_i^o$ again evolves according to (23). The network SNR and step-sizes are respectively SNR=$[22.5, 11.6, 14.5, 17.7, 10.1, 14.7, 10.3, 18.7, 10.5, 17.8, 15.9, 12.4, 17.8, 15.2, 19.5]$ and $\mu_k = 0.1 \cdot [1, 10, 1, 1, 10, 1, 10, 1, 10, 1, 1, 1, 10, 1, 1]$. Figure 6-(a) shows how U-sup and LS-alg algorithms perform (much) better than the non-cooperative case, and better than the MSD-alg, when the time-varying plant evolves under $\sigma_q^2 = 10^{-7}$. Increasing the plant velocity to $\sigma_q^2 = 10^{-5}$, in Fig. 6-(b), the LS-alg is nearly 10 dB worse (and still worsening: it has not converged after $2 \cdot 10^5$ iterations) than the U-sup and MSD-alg, and worse than the non-cooperative case. Figure 7-(a) corresponds to the steady-state for Fig. 6-(a), and shows that both U-sup and LS-alg are universal for $\sigma_q^2 = 10^{-7}$, while the MSD-alg is not. The algorithm robustness is depicted by Fig. 7-(b), again confirming that the U-sup is the only algorithm that is universal across the entire $\sigma_q^2$ test range, also depicting the LS-alg sensitivity when tracking fast varying plants.

**In Example 3**, the scenario is what is more likely to take place in practice, where nodes have approximately



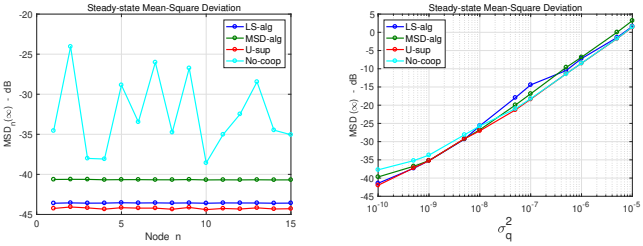

(a) (b)

Fig. 8. **Example 3 (Correlated inputs)**: (a) Network topology; (b) The mean adaptive combiners $\mathrm{E}\lambda_n(i)$ corresponding to Figs. 9-(a) and 10-(a).




(a) (b)

Fig. 9. **Example 3 (Correlated inputs)**: (a) Network $\mathrm{MSD}(i)$ for correlated inputs and a stationary plant; (b) Network $\mathrm{MSD}(i)$ for the same inputs and a time-varying plant with $\sigma_q^2 = 10^{-7}$.

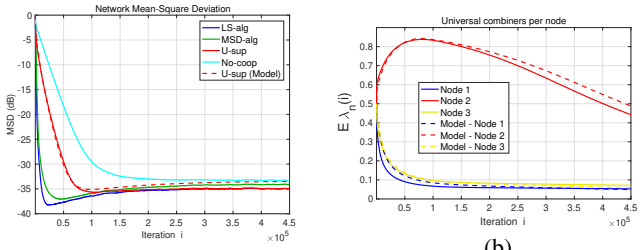



(a) (b)

Fig. 10. **Example 3 (Correlated inputs)**: (a) Steady-state $\mathrm{MSD}_n(\infty)$; (b) Tracking robustness in terms of MSD versus $\sigma_q^2$: $\min_n$ $\mathrm{MSD}_n(\infty)$ for the non-cooperative case and $\max_n$ $\mathrm{MSD}_n(\infty)$ for the cooperative algorithms.

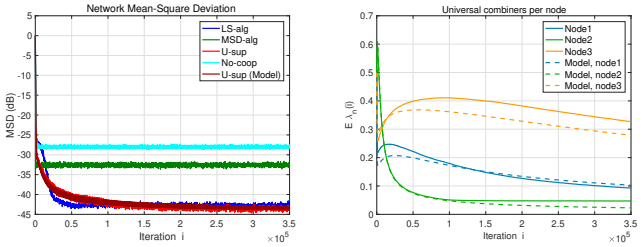
the same node degree, without a clearly exceptional or poor node in terms of SNR, and considering correlated inputs $\{u_n(i)\}$ obtained from (49), with correlation factors $\{\beta_n\}$ randomly selected from the two values 0.63 and 0.95 and captured by the vector $\beta = [0.63, 0.95, 0.63, 0.63, 0.95, 0.63, 0.95, 0.63, 0.95, 0.63, 0.63, 0.63, 0.95, 0.63, 0.63]$. Figures 8-(a) and (b) depict network topology and the mean combiners $\mathrm{E}\lambda_n(i)$ corresponding to Figs. 9-(a) and 10-(a); the former presenting the performance for a stationary plant, the latter of a non-stationary plant. The network SNR and step-sizes are SNR $= [12.2, 15.2, 15.5, 15.5, 20, 11.2, 17.2, 12.4, 17.8, 16.1, 12.6, 10.1, 19.5, 12.1, 12.6]$ and $\mu_k = 0.1 \cdot [1, 10, 1, 1, 10, 1, 10, 1, 10, 1, 1, 1, 10, 1, 1]$. We then test the algorithms when tracking a non-stationary plant with $\sigma_q^2 = 10^{-7}$ in Fig. 9-(b). Observe in Figs. 9-(a) and (b) how the U-sup algorithm outperforms both the MSD-alg and LS-alg algorithms. As depicted in Figs. 10-(a) and (b), the three algorithms present a similar performance, however once more only the U-sup attains universality in the entire test range for $\sigma_q^2$.

In **Example 4**, we test the theoretical model developed in Section V. For that, we revisit the scenario from Example 3,




(a) (b)

Fig. 11. **Example 4 (Correlated inputs)**: (a) Network MSD($i$) and the theoretical model for U-sup (dashed red); (b) The mean adaptive combiners E$\lambda_n(i)$ for a few nodes, corresponding to Fig 11-(a).




(a) (b)

Fig. 12. **Example 5 (White inputs)**: (a) Network MSD($i$) and the theoretical model for U-sup (dark red); (b) The mean adaptive combiners E $\lambda_n(i)$ for a few nodes and their theoretical model (dashed lines), all corresponding to Fig. 12-(a).

changing the non-stationarity level for $\sigma_q^2 = 10^{-10}$, and set the feedback cycle $L \to \infty$. Figure 11-(a) shows the network MSD evolution, and Fig. 11-(b) presents the corresponding mean combiner evolution E$\lambda_n(i)$. Note how the theoretical model is able to capture the general tendencies correctly.

In **Example 5**, we test the theoretical model from Section V for white inputs in a network with $N = 8$, NLMS AFs with order $M = 6$, with the U-sup using $L = 800$, all identifying a stationary plant. The stepsizes for this example are captured by the vector $\mu = 0.01 \cdot [1, 10, 1, 10, 1, 1, 1, 1]$. The network topology is described by the reduced undirected edge set $E' = [12, 13, 18, 24, 25, 34, 56, 58, 67, 78]$, in which, for example, the pair 12 means there are edges between nodes 1 and 2, between node 1 and itself and between 2 and itself. The SNR across the network is SNR $= [11.2, 10.6, 18.4, 13.4, 17.8, 11.2, 16.8, 10.9]$. The theoretical model is shown in Figs. 12(a) and (b) to describe well the network MSD performance and the combiner evolution.

The last example, **Example 6**, shows the effect of $L$ in a network with $N = 8$ and $M = 6$ as in Example 5. Figure 13 shows the worst node MSD at various iterations during convergence, for different values of $L$. One can see how intermediate values of $L$ (between 10 and 1,000) accelerate convergence, without affecting the steady-state performance. In other words, the U-sup algorithm is relatively insensitive to the choice of $L$ over a wide range.

## VII. Conclusion

This work has established the concept of universal estimation in the context of distributed adaptive estimation, also proposing a distributed adaptive protocol capable of attaining distributed global universality, as proved by Theorem 2

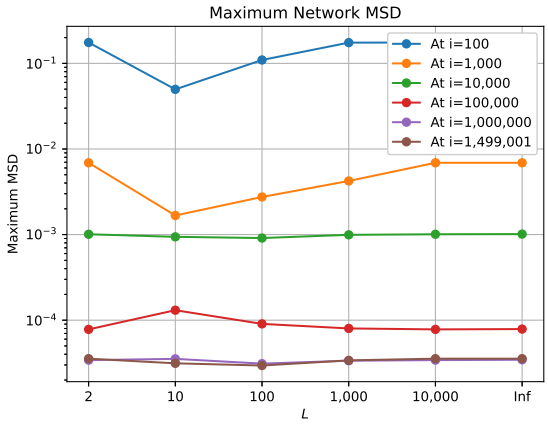



Fig. 13. Performance comparison with different values of $L$ for a 8-node network: worst-case node MSD at various points during convergence. The values were obtained by averaging the MSD values at each node separately between instants $i$ and $i + 999$, and taking the maximum value between the nodes.

and shown by several simulations: the distributed supervisor drives the entire network to the best node performance, also efficiently rejecting poorly performing nodes, while promoting performance uniformity across the nodes.

Fair comparisons were carried out with two other directly competing algorithms, namely the MSD-alg (10) [36], and the LS-alg (11) [43]. The proposed U-sup algorithm (17) is the simplest and was the only method to consistently achieve universality in all tested scenarios, under white and correlated data, for stationary and fast-varying plants. Algorithm (11) performs very well for some stationary and slowly varying plants, however its computational complexity may limit its use in some applications.

Theoretical mean and mean-square error models were developed with a reasonable agreement with simulations, capturing the general tendencies of network MSD and local combiners E$\lambda_n(i)$, and proving convergence of the algorithm in the mean. The agreement between simulated and analytical curves improves as the local AF step-size decreases ($\mu_n \to 0$); in the distributed case the same effect is further noticed when the universal combiner stepsize is also decreased, i.e., $\mu_a \to 0$.

## References

[1] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[2] F. Cattivelli and A. Sayed, "Distributed detection over adaptive networks using diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 59[5], pp. 1917–1932, 2011.

[3] R. V. Kulkarni, A. Förster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 68–96, 2011.

[4] Y. P. Bergamo and C. G. Lopes, "Scalar field estimation using adaptive networks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3565–3568.

[5] A. Scaglione, R. Pagliari, and H. Krim, "Non-cooperative versus cooperative approaches for distributed network synchronization," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07).*, 2007, pp. pp. 537–541.

[6] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Distributed spectrum estimation for small cell networks based on sparse diffusion adaptation," *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1261–1265, 2013.

[7] X. Zhai, H. Jing, and T. Vladimirova, "Multi-sensor data fusion in wireless sensor networks for planetary exploration," in *2014 NASA/ESA Conf. on Adaptive Hardware and Systems (AHS)*, 2014, pp. 188–195.

[8] G. S. Vicinansa, Y. P. Bergamo, and C. G. Lopes, "Position estimation from range measurements using adaptive networks," in *2016 IEEE Sensor Array and Multichannel Sig. Proc. Workshop*, 2016, pp. 1–5.

[9] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 13, no. 1, pp. 65–78, 2004.

[10] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. Asilomar Conf. Signal, Syst., Comput.*, 2008, p. pp. 1722–1726.

[11] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed lms for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[12] A. Sandryhaila, S. Kar, and J. M. F. Moura, "Finite-time distributed consensus through graph filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, 2014, pp. pp. 1080–1084.

[13] S. Kar and J. M. F. Moura, "Convergence Rate Analysis of Distributed Gossip (Linear Parameter) Estimation: Fundamental Limits and Tradeoffs," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[14] P. C. Chen and P. P. Vaidyanathan, "Distributed algorithms for array signal processing," *IEEE Trans. Signal Process.*, vol. 69, pp. 4607–4622, 2021.

[15] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *2007 IEEE International Conference on Acoustics, Speech and Sig. Proc. - ICASSP '07*, vol. 3, 2007, pp. III–917–III–920.

[16] ——, "Incremental adaptive strategies over distributed networks," *IEEE Trans. on Sig. Proc.*, vol. 55[8], pp. 4064–4077, 2007.

[17] ——, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56[7], pp. 3122–3136, 2008.

[18] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56[5], pp. 1865–1877, 2008.

[19] S.-Y. Tu and A. H. Sayed, "Diffusion Strategies Outperform Consensus Strategies for Distributed Estimation Over Adaptive Networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[20] A. Sayed, *Adaptive filters*. Wiley-IEEE Press, 2008.

[21] P. Diniz, *Adaptive filtering: Algorithms and practical implementation*, 4th ed. Springer, 2013.

[22] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, 2006, pp. 233–237.

[23] C. G. Lopes and A. H. Sayed, "Randomized incremental protocols over adaptive networks," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 3514–3517.

[24] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based lms for node-specific adaptive parameter estimation," *IEEE Trans. on Sig. Proc.*, vol. 63, no. 13, pp. 3448–3460, 2015.

[25] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3285–3288.

[26] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, 2011.

[27] I. Harrane, R. Flamary, and C. Richard, "On reducing the communication cost of the diffusion lms algorithm," *IEEE Trans. on Sig. and Information Proc. over Networks*, vol. 5, no. 1, pp. 100–112, 2019.

[28] D. Jin, J. Chen, C. Richard, J. Chen, and A. H. Sayed, "Affine combination of diffusion strategies over networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2087–2104, 2020.

[29] ——, "Convex combination of diffusion strategies over networks," *IEEE Trans. on Sig. and Inf. Proc. over Networks*, vol. 6, pp. 714–731, 2020.

[30] B. Bollobás, *Modern Graph Theory*. Springer, 1998.

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

[32] M. O. Sayin and S. S. Kozat, "Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5308–5323, 2014.

[33] S.-Y. Tu and A. Sayed, "Optimal combination rules for adaptation and learning over networks," in *Int. Workshop on Computational Advances in Multi-Sensor Adaptive Proc.* IEEE, 2011, pp. 317–320.

[34] A. Sayed, *Adaptation, Learning, and Optimization over Networks*. NOW Publishers, 2014.

[35] S. Werner, Y.-F. Huang, M. L. R. de Campos, and V. Koivunen, "Distributed parameter estimation with selective cooperation," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2009, pp. 2849–2852, iSSN: 2379-190X.

[36] N. Takahashi, I. Yamada, and A. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58[9], pp. 4795–4810, 2010.

[37] J. Fernandez-Bes, L. Azpicueta-Ruiz, M. Silva, and J. Arenas-Garcia, "A novel scheme for diffusion networks with least-squares adaptive combiners," in *International Workshop on Machine Learning for Signal Processing*. IEEE, 2012, pp. 1–6.

[38] C. G. Lopes, V. H. Nascimento, and L. F. O. Chamon, "Towards spatially universal adaptive diffusion networks," in *2014 IEEE Global Conference on Signal and Information Processing*. IEEE, 2014, pp. 803–807.

[39] J. Arenas-García, A. Figueiras-Vidal, and A. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54[3], pp. 1078–1090, 2006.

[40] R. Candido, M. T. M. Silva, and V. H. Nascimento, "Transient and steady-state analysis of the affine combination of two adaptive filters," *IEEE Trans. on Sig. Proc.*, vol. 58[10], no. 10, pp. 4064–4078, 2010.

[41] N. Merhav and M. Feder, "Universal prediction," *IEEE Trans. Inf. Theory*, vol. 44[6], pp. 2124–2147, 1998.

[42] A. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Trans. on Sig. Proc.*, vol. 47[10], pp. 2685–2699, 1999.

[43] J. Fernandez-Bes, J. Arenas-Garcia, M. T. M. Silva, and L. A. Azpicueta-Ruiz, "Adaptive diffusion schemes for heterogeneous networks," *IEEE Trans. Signal Process.*, vol. 65[21], pp. 5661–5674, 2017.

[44] V. H. Nascimento and M. T. M. Silva, "Adaptive filters," in *Academic Press Library in Signal Processing:*, R. Chellappa and S. Theodoridis, Eds. Chennai: Academic Press, 2014, vol. 1, Signal Processing Theory and Machine Learning, pp. 619—761.

[45] R. G. Gallager, *Discrete Stochastic Processes*. Kluwer Academic Publishers, 1996.

[46] Y. Zhang and J. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *IEEE Signal Process. Lett.*, vol. 13[10], pp. 628–631, 2006.

[47] L. Azpicueta-Ruiz, A. Figueiras-Vidal, and J. Arenas-García, "A normalized adaptation scheme for the convex combination of two adaptive filters," in *ICASSP 2008*, 2008, pp. 3301–3304.

[48] M. Silva and V. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56[7], pp. 3137—-3149, 2008.

[49] V. Nascimento, M. Silva, R. Candido, and J. Arenas-García, "A transient analysis for the convex combination of adaptive filters," in *IEEE Statistical Signal Processing Workshop (SSP)*, 2009, pp. 53–56.

[50] C. G. Lopes, E. H. Satorius, P. Estabrook, and A. H. Sayed, "Adaptive carrier tracking for mars to earth communications during entry, descent, and landing," *IEEE Trans. Aer. and Elec. Systems*, vol. 46[4], pp. 1865–1879, 2010.

[51] L. Chamon, W. Lopes, and C. Lopes, "Combination of adaptive filters with coefficients feedback," in *International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 3785–3788.

[52] A. E. Feitosa, V. H. Nascimento, and C. G. Lopes, "Adaptive detection in distributed networks using maximum likelihood detector," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 974–978, 2018.

[53] N. Cesa-Bianchi and G. Lugosi, "On prediction of individual sequences," *Annals of Statistics*, vol. 27[6], pp. 1865–1895, 1999.

[54] J. Arenas-Garcia, L. Azpicueta-Ruiz, M. Silva, V. Nascimento, and A. Sayed, "Combinations of Adaptive Filters: Performance and convergence properties," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 120–140, Jan. 2016.

[55] M. Tarrab and A. Feuer, "Convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, pp. 680–691, Jul. 1988.

[56] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge Univ. Press, 2013.

**Cássio G. Lopes** (S'06–M'08–SM'16) received his B.S. degree and M.S. degrees in electrical engineering from the Federal University of Santa Catarina, Brazil, in 1997 and 1999, respectively, and M.S. and Ph.D. degrees in electrical engineering from the University of California, Los Angeles (UCLA) in 2004 and 2008, respectively. From 2005 to 2007 he worked in a joint UCLA/NASA Jet Propulsion Laboratory project to develop frequency tracking algorithms to support direct-to-Earth Mars communications during entry, descent, and landing. In 2008 he worked at the Instituto Tecnologico de Aeronautica (ITA) Sao Jose dos Campos, Brazil as a postdoctoral researcher, developing distributed estimation algorithms for inertial navigation. In 2008 he joined the Department of Electronic Systems of the University of Sao Paulo (USP), Polytechnic School, where he is an associate professor of electrical engineering since 2014. From 2014 to 2018 he worked in a joint USP/EMBRAER project to enhance acoustic emission techniques for Structural Health Monitoring (SHM) of aircrafts. His current research interests are theory and methods for adaptive and statistical signal processing, distributed adaptive estimation, geometric algebra and tensor adaptive processing, as well as SHM.

**Vítor H. Nascimento** obtained the B.S. and M.S. degrees in Electrical Engineering from Escola Politécnica, University of São Paulo, Brazil, in 1989 and 1992, respectively, and the Ph.D. degree from the University of California, Los Angeles, in 1999. From 1990 to 1994 he was a Lecturer at the Univ. of São Paulo, and in 1999 he joined the faculty at the same school where his now Professor and chair of the Dept. of Electronic Systems Engineering. One of his papers received the 2002 IEEE SPS Best Paper Award. He served as an Associate Editor for the IEEE Signal Processing Letters from 2003 to 2005, for the IEEE Transactions on Signal Processing from 2005 to 2008 and for the EURASIP Journal on Advances in Signal Processing from 2006 to 2009, and as a Senior Area Editor for the IEEE Trans. on Signal Processing (2018-2021). He was a member of the IEEE-SPS Signal Processing Theory and Methods Technical Committee (2007 -2012 and 2016-2021). From 2010 to 2014 he was chair of the São Paulo IEEE-SPS Chapter, and between 2012 and 2016 he served as area editor for the Journal of Communication and Information Systems. He was Technical Chair of the 2014 International Telecommunications Symposium, organized in São Paulo by the Brazilian Telecommunications Society (SBrT), of the 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (Rio de Janeiro, Brazil), and of the 2021 IEEE Statistical Signal Processing Workshop (Rio de Janeiro). His research interests include signal processing theory and applications, statistical methods for epidemiology, adaptive and sparse estimation, distributed learning, structural health monitoring, array signal processing, and applied linear algebra.

**Luiz F. O. Chamon** received the B.Sc. and M.Sc. degrees in electrical engineering from the University of São Paulo, São Paulo, Brazil, in 2011 and 2015 and the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania (Penn), Philadelphia, in 2020. Until 2022, he was a postdoctoral fellow at the Simons Institute of the University of California, Berkeley. He is currently an independent research group leader at the University of Stuttgart, Germany. In 2009, he was an undergraduate exchange student of the Masters in Acoustics of the École Centrale de Lyon, Lyon, France, and worked as an Assistant Instructor and Consultant on nondestructive testing at INSACAST Formation Continue. From 2010 to 2014, he worked as a Signal Processing and Statistics Consultant on a research project with EMBRAER. He received both the best student paper and the best paper awards at IEEE ICASSP 2020 and was recognized by the IEEE Signal Processing Society for his distinguished work for the editorial board of the IEEE Transactions on Signal Processing in 2018. His research interests include optimization, signal processing, machine learning, statistics, and control.