Adaptive Diffusion Schemes for Heterogeneous Networks

Jesus Fernandez-Bes, Jerónimo Arenas-García, Magno T. M. Silva, *Member, IEEE*, and Luis A. Azpicueta-Ruiz, *Member, IEEE*

***Abstract*—In this paper, we deal with distributed estimation problems in diffusion networks with heterogeneous nodes, i.e., nodes that either implement different adaptive rules or differ in some other aspect such as the filter structure or length, or step size. Although such heterogeneous networks have been considered from the first works on diffusion networks, obtaining practical and robust schemes to adaptively adjust the combiners in different scenarios is still an open problem. In this paper, we study a diffusion strategy specially designed and suited to heterogeneous networks. Our approach is based on two key ingredients: 1) the adaptation and combination phases are completely decoupled, so that network nodes keep purely local estimations at all times and 2) combiners are adapted to minimize estimates of the network mean-square-error. Our scheme is compared with the standard adapt-then-combine scheme and theoretically analyzed using energy conservation arguments. Several experiments involving networks with heterogeneous nodes show that the proposed decoupled adapt-then-combine approach with adaptive combiners outperforms other state-of-the-art techniques, becoming a competitive approach in these scenarios.**

***Index Terms*—Adaptive networks, diffusion networks, distributed estimation, least-squares, mean-square performance.**

Manuscript received November 18, 2016; revised April 29, 2017; accepted August 6, 2017. Date of publication August 15, 2017; date of current version August 31, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kostas Berberidis. The work of J. Fernandez-Bes was supported in part by Project TIN2013-41998-R and Project DPI2016-75458-R from the Spanish Ministry of Economy and Competitiveness (MINECO), Spain, in part by MULTITOOLS2HEART from CIBER-BBN through Instituto de Salud Carlos III, Spain, in part by the European Social Fund (EU) and Aragón Government through BSICoS group (T96), and in part by the European Research Council (ERC) through Project ERC-2014-StG 638284. The work of J. Arenas-García was supported in part by MINECO Project TEC2014-52289-R and in part by Comunidad de Madrid Project PRICAM S2013/ICE-2933. The work of M. T. M. Silva was supported in part by CNPq under Grant 304275/2014-0, and in part by FAPESP under Grant 2012/24835-1. The work of L. A. Azpicueta-Ruiz was supported in part by Comunidad de Madrid under Grant CASI-CAM-CM (id. S2013/ICE-2845), in part by the Spanish Ministry of Economy and Competitiveness under Grants DAMA TIN2015-70308-REDT and TEC2014-52289-R, and in part by the European Union. *(Corresponding author: Jesus Fernandez-Bes.)*

J. Fernandez-Bes is with BSICoS Group, I3AIIS Aragón, and CIBER-BBN, University of Zaragoza, Zaragoza 50018, Spain (e-mail: jfbes@unizar.es).

J. Arenas-García and L. A. Azpicueta-Ruiz are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Leganés 28911, Spain (e-mail: jarenas@tsc.uc3m.es; azpicueta@tsc.uc3m.es).

M. T. M. Silva is with the Department of Electronic Systems Engineering, Escola Politécnica,Universidade de São Paulo, São Paulo 05508-010, Brazil (e-mail: magno.silva@usp.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2740199

I. Introduction

OVER the last years, adaptive diffusion networks have become an attractive and robust approach to estimate a set of parameters of interest in a distributed manner (see, e.g., [1]–[15] and their references). Compared to other distributed schemes, such as incremental and consensus strategies, diffusion techniques present some advantages, e.g., they are more robust to link failures or they do not require the definition of a cyclic path that runs across the nodes as in incremental solutions [16]. Furthermore, they perform better than consensus techniques in terms of stability, convergence rate, and tracking ability [5]. For these reasons, adaptive diffusion networks are considered an efficient solution in applications such as target localization and tracking [4], environment monitoring [5], and spectrum sensing in mobile networks [4], [17], among others. Moreover, they are also suited to model complex behaviors exhibited by biological or socioeconomic networks [5].

Diffusion networks consist of a collection of connected nodes, linked according to a certain topology, that cooperate with each other through local interactions to solve a distributed inference or optimization problem in real time. Each node is able to extract information from its local measurements and combine it with the ones received from its neighbors [4], [5]. This is typically performed in two stages: adaptation and combination. The order in which these stages are performed leads to two possible schemes: Adapt-then-Combine (ATC) and Combine-then-Adapt (CTA) [1], [18]. In both cases, the adaptation and combination steps are interleaved with the communication of the intermediate estimates among neighbors. In general, it is assumed that this communication among the nodes is synchronous, though an analysis of asynchronous diffusion strategies is available in [19].

In this paper, we focus on *heterogeneous* diffusion networks.[1] We refer as *heterogeneous* to networks whose nodes implement diverse update functions, i.e., they can differ in the filter length or structure, step sizes, or even in the implemented learning rule. This is different to other popular scenarios such as multitask or node-specific diffusion networks [23], where the heterogeneity is in the input that nodes receive or/and in the task they solve. Heterogeneous nodes are an interesting choice to improve the tracking performance of the network, or simply to achieve a better tradeoff between computational cost and convergence

[1] Although there exist many studies involving heterogeneous networks in the literature (sensor networks [20], epidemiology [21] or cellular networks [22]), here we restrict ourselves to the particular case of diffusion networks.

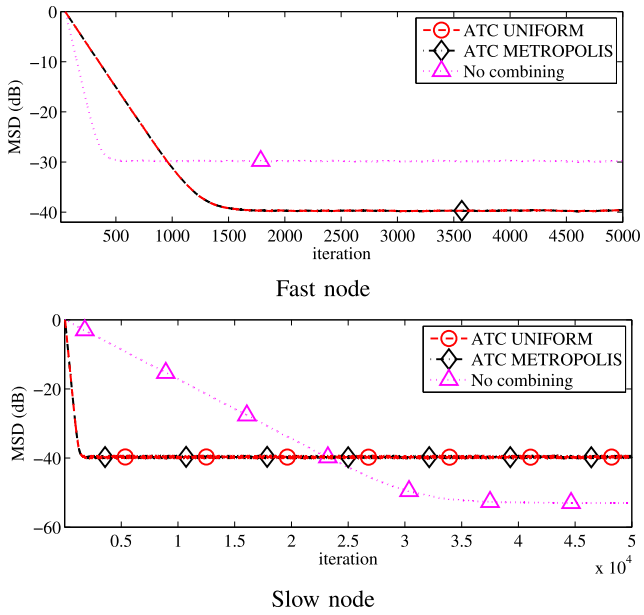


Fig. 1. Estimation error for two nodes of a fully-connected ATC network with 5 nodes. All nodes implement LMS rules, but node 1 (fast node) uses a larger step size than the rest of nodes.

rate by updating the nodes with different algorithms. Thus, it is not surprising that such heterogeneous networks have been considered in the literature from the first works on diffusion networks. For instance, [4] and [5] already considered in the analysis of the ATC and CTA schemes the case of least-mean-squares (LMS) nodes with different step sizes, and [24] used the term *heterogeneous* to refer to informed and uninformed (i.e., without access to local measurements) nodes in a diffusion network.

Compared to multitask scenarios, experimental studies of this kind of networks have been quite limited up to now and we think that they deserve more attention. One possible explanation is that overall network performance can be very sensitive to an inappropriate selection of the network combiners. To illustrate this, Fig. 1 shows the performance of the ATC scheme of [18] for a fully-connected network of 5 nodes implementing LMS updates, using static combiners. In this example, four nodes have a common step size, whereas the last node uses a larger value that can provide faster convergence. As it can be seen, in this case the diffusion strategy actually degrades the convergence of the fast node and the steady-state performance of the slow ones with respect to the operation of these nodes in the non-cooperation case. This is due to the use of fixed combiners that 'contaminate' the estimation of the fast node during convergence, whereas during the steady-state regime slow nodes fuse their estimations with that coming from the fast node, resulting in larger rather than smaller estimation error.

This simple example illustrates the importance of adequate rules for setting and adjusting the combination parameters in heterogeneous networks. Indeed, combination weights play an essential role in the overall performance of the network. For instance, diffusion least-mean-squares (LMS) strategies can perform similarly to classical centralized solutions when the weights used to combine the neighbors estimates are optimally adjusted [4], [25], [26]. Initially, different static combination rules were proposed such as Uniform [27], Laplacian [28], Metropolis [28], and Relative Degree [29]. Some adaptive schemes for adjusting the combination weights (e.g., [25], [26], [30], [31]) have also been proposed in order to optimize the network performance under spatially varying signal-to-noise ratio (SNR). Although these adaptive rules can reduce the steady-state error with respect to static combiners, some experiments show a deterioration in the convergence behavior [31]. Consequently, some schemes propose the use of different rules for transitory and steady-state regimes and include mechanisms to switch from one rule to the other in an online manner [31], [32]. Since all these works are generally optimized assuming homogeneous nodes, an important challenge is related with the fact that all the above-mentioned combination rules result in degraded performance when used with heterogeneous networks, and there are presently no alternatives for dealing with such problem in a general case.

In this work, we focus on an alternative diffusion scheme especifically designed for heterogeneous networks. In our approach, firstly proposed in [33], [34], and which will be called Decoupled ATC (D-ATC), the adaptation phase is kept decoupled from the combination phase, i.e., the local estimation of each node is combined with the estimates received from its neighbors, as in standard ATC, but the resulting combined estimation is not fed back into the next adaptation step. This scheme presents a more clear separation between the adaptation and combination phases. As it will be shown later, this allows us to implement mean-square-error (MSE) based rules for the combination phase which offer an adequate behavior for heterogeneous networks. With these rules we obtain a significant improvement in convergence and steady-state performance with respect to previous approaches, both in tracking and stationary scenarios. In addition, our proposal seems to be a more natural scheme for asynchronous networks, which are receiving increasing attention [19].

This paper extends our previous works [33], [34] in different ways:

1) We analyze the mean behavior of our diffusion strategy and derive sufficient conditions for the network combiners that guarantee the mean stability of the algorithm.
2) Using energy conservation arguments [35], we derive closed-form expressions for the steady-state mean-square deviation (MSD) of the network and of its individual nodes in a non stationary environment.
3) We propose two new rules for adjusting the combination weights: One following a Least-Squares (LS) approach, in the same vein as the one introduced in [33], [34], and one based on the Affine Projection Algorithm (APA).
4) Finally, we include detailed simulation work, both for stationary and tracking scenarios, to illustrate the performance of the proposed schemes and to corroborate the theoretical results.

The paper is organized as follows. The general formulation of ATC diffusion strategies for heterogeneous networks, together

TABLE I
SUMMARY OF THE NOTATION USED IN THE PAPER

N	Number of nodes in the network
$\mathcal{N}_k$	Neighborhood of node k, including itself
N_k	Cardinality of $\mathcal{N}_k$
$\bar{\mathcal{N}}_k$	Neighborhood of node k, excluding itself
$\bar{N}_k$	Cardinality of $\bar{\mathcal{N}}_k$
$\bar{\mathbf{b}}_k$	Vector with the indexes of all nodes in $\bar{\mathcal{N}}_k$
$\bar{b}_k^{(m)}$	Index of the m^{th} node connected to node k
$\mathbf{w}_{\text{o}}(n)$	Unknown time-varying parameter vector
$\boldsymbol{\psi}_k(n)$	Local estimate of $\mathbf{w}_{\text{o}}(n)$ (based only on local data at node k)
$\mathbf{w}_k(n)$	Combined estimate of $\mathbf{w}_{\text{o}}(n)$ at node k
$\{d_k(n), \mathbf{u}_k(n)\}$	Local desired value and regression vector at node k
$v_k(n)$	Local noise at node k
$y_k(n)$	Local output of node k
$e_k(n)$	Local error of node k
$c_{\ell k}(n)$	Combination weight assigned by node k to the estimate received from node $\ell \in \mathcal{N}_k$
$\mathbf{c}_k(n)$	Vector with combination weights associated to node k
$\bar{\mathbf{c}}_k(n)$	Vector with the same entries of $\mathbf{c}_k(n)$, excluding $c_{kk}(n)$

with the introduction of adaptive combiners, is presented in Section II. In Section III, the Decoupled ATC strategy is proposed, and we theoretically analyze it in Section IV. APA and LS-based rules for adapting network combiners are derived in Section V. Experimental results are provided in Section VI, and we close the paper with our main conclusions and some possibilities for future works in Section VII.

A. Notation

We use boldface lowercase letters to denote vectors and boldface uppercase letters to denote matrices. The superscript T represents the transpose of a matrix or a vector. Depending on the context, $\mathbf{0}_N$ represents an $N \times N$ matrix or a length-N column vector with all elements equal to zero, and $\mathbb{1}_N$ is an all-ones column vector with length N. In addition, to simplify the arguments, we assume that all the variables are real. Table I summarizes the notation that is used throughout the paper.

II. HETEROGENEOUS DIFFUSION NETWORKS WITH MSE-BASED ADAPTIVE COMBINERS

A. ATC and CTA Diffusion Strategies

Consider a collection of N nodes connected according to a certain topology, as depicted in Fig. 2. Each node k shares information with its neighbors and we denote this neighborhood of k, excluding the node itself, $\bar{\mathcal{N}}_k$, while $\mathcal{N}_k = \bar{\mathcal{N}}_k \cup \{k\}$. The network objective at every time instant n is to obtain, in a distributed manner, the solution that minimizes a certain global cost function $J[\mathbf{w}(n)]$.

In this work, we consider that the global cost and the individual cost of each agent are the mean-square error (MSE). In particular, we consider a linear estimation setting: At every time instant n, each node k has access to a scalar measurement $d_k(n)$ and a regression column vector $\mathbf{u}_k(n)$ of length M, both realizations of zero-mean random processes. We assume

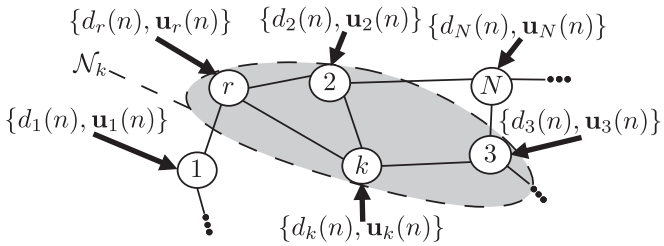


Fig. 2. Example of diffusion network: At every time step n, each node k takes a measurement $\{d_k(n), \mathbf{u}_k(n)\}$. In this example, the neighborhood of node k is $\mathcal{N}_k = \{2, 3, r, k\}$ and its cardinality is $N_k = 4$.

that these measurements are related via some unknown column vector $\mathbf{w}_{\text{o}}(n)$ of length M through a linear model

$$d_k(n) = \mathbf{u}_k^T(n)\mathbf{w}_{\text{o}}(n) + v_k(n), \tag{1}$$

where $v_k(n)$ denotes measurement noise and is assumed to be a realization of a zero-mean white random process with power $\sigma_{v,k}^2$ and independent of all other variables across the network. The objective of the network is estimating the (possibly) time-varying parameter vector $\mathbf{w}_{\text{o}}(n)$.

In standard ATC or CTA diffusion strategies, adaptation and combination phases are iterated to solve this estimation problem in an adaptive and distributed manner. In particular, the ATC scheme has the following two steps

$$\boldsymbol{\phi}_k(n) = f_k(\mathbf{w}_k(n-1), \mathbf{u}_k(n), d_k(n), \boldsymbol{\eta}_k), \tag{2}$$

$$\mathbf{w}_k(n) = \sum_{\ell \in \mathcal{N}_k} c_{\ell k}(n)\boldsymbol{\phi}_\ell(n), \tag{3}$$

where an intermediate estimation $\boldsymbol{\phi}_k(n)$ is calculated as a function of these elements: the previous estimation $\mathbf{w}_k(n-1)$, current local data $\{\mathbf{u}_k(n), d_k(n)\}$ and a state vector η_k that incorporates any other information needed for filter adaptation. Some typical choices for the adaptation stage (2) are least-mean-squares (LMS), normalized least-mean-squares (NLMS), Affine Projection Algorithm (APA) [35], etc. $\boldsymbol{\phi}_k(n)$ is then shared with the neighbors and combined by means of the coefficients $c_{\ell k}(n)$, $\ell \in \mathcal{N}_k$, to calculate $\mathbf{w}_k(n)$.

Note that we assume that the update rules on (2) can be different among the nodes but all of them try to solve the same estimation task. This is complementary to other approaches such us multitask networks [36], where each node solves a different but related task.

B. Adaptive Combiners for Heterogeneous Networks

Different update rules to adapt the combiners in diffusion schemes have been proposed in the literature [25], [26], [30], all of them based in the approximated minimization of the network MSD (NMSD) defined as

$$\begin{aligned} \text{NMSD}(n) &= \frac{1}{N}\sum_{k=1}^{N} \text{MSD}_k(n) \\ &= \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\{[\mathbf{w}_{\text{o}}(n) - \mathbf{w}_k(n)]^2\}, \end{aligned} \tag{4}$$

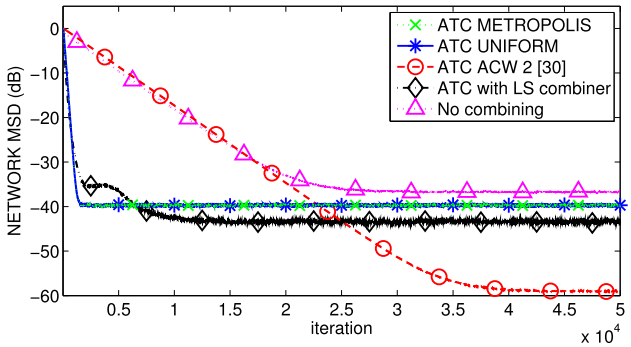


Fig. 3. Estimation error for an ATC network with different combiners. It can be observed that state-of-the-art adaptive combination rules (in red) obtain a very good steady-state behaviour, much better that the ATC with LS combiners (black) because of the numerical problems of the latter.

where $\text{MSD}_k(n)$ is the mean-square deviation of each node k in the network at iteration n.

However, the NMSD is very complicated to estimate as $\mathbf{w}_o$ is unknown, and the available approximations [25], [26], [30] are only valid for the steady-state NMSD, $\text{NMSD}(\infty) = \lim_{n\to\infty} \text{NMSD}(n)$. Moreover, there is no approach in the literature that approximately minimizes (4) in heterogeneous networks, not even for the well-studied case of diffusion LMS networks with different step sizes in the nodes.

In this work, we follow a different approach. We propose an update rule driven by the online minimization of network MSE (NMSE), defined as

$$\text{NMSE}(n) = \frac{1}{N}\sum_{k=1}^{N}\text{MSE}_k(n) = \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\{\check{e}_k^2(n)\}, \qquad (5)$$

where $\check{e}_k(n) = d_k(n) - \check{y}_k(n) = d_k(n) - \mathbf{u}_k^T(n)\mathbf{w}_k(n-1)$ represents the error at node k using combined estimates available at that node, while $\check{y}_k(n)$ stands for the corresponding combined output.

It is well-known that for linear regression problems both criteria, MSD and MSE, are tightly related [35]. In our approach, $\text{MSE}_k(n)$ can be easily approximated during both the convergence and the steady-state phases, and whatever the kind of update implemented by each node of the network. In addition, other advantage of using the NMSE as the optimization criterion lies in the fact that its minimization can be tackled with well known algorithms to update the combination weights, including gradient-based or LS strategies. In this respect, while using the NMSD requires a model for the network performance to overcome the lack of knowledge of the optimum weight vector, minimization of the network MSE can be seen as a model-free approach.

Fig. 3 illustrates the performance of a standard ATC diffusion in a fully connected network with 5 LMS nodes. As in the example in Fig. 1, one node has a comparably larger step size than the other nodes. The scheme of [30], with adaptive combination weights (ATC ACW 2), is not able to exploit the fast convergence of the node with large step size. This is due to the lack of an accurate NMSD approximation during convergence. When the network combiners are adjusted using an LS criterion (ATC with LS combiner), the behavior of the network is similar to the use of static combiners, what made us wonder why the network did not fully exploit the best properties of the different nodes. When analyzing this problem in detail, we observed severe numerical problems due to the coupled nature of adaptation and combination stages in standard ATC, i.e., the feedback of the combined weights in the adaptation step causes almost perfect correlation among the weight estimations at all nodes, ill-conditioning the minimization of (5).

In the next section, we propose a novel diffusion scheme that overcomes these numerical instabilities, permitting an effective use of NMSE-based adaptive combiners.

III. Decoupled Adapt-Then-Combine Diffusion

Recently, we proposed a diffusion method [33], [34] that also iterates an adaptation and a combination phase. However, differently from standard ATC or CTA diffusion [4], [5], each node in our scheme preserves and adapts a purely local estimation $\boldsymbol{\psi}_k(n)$, which is then combined with the combined estimates, $\mathbf{w}_\ell(n-1)$, received from the neighboring nodes $\ell \in \bar{\mathcal{N}}_k$ at the previous iteration. Note that, although we have selected an ATC approach as the basis of our algorithm, it could be straightforwardly extended to CTA. Consequently, the proposed diffusion scheme can be written as follows

$$\boldsymbol{\psi}_k(n) = f_k(\boldsymbol{\psi}_k(n-1), \mathbf{u}_k(n), d_k(n), \boldsymbol{\eta}_k), \qquad (6)$$

$$\mathbf{w}_k(n) = c_{kk}(n)\boldsymbol{\psi}_k(n) + \sum_{\ell\in\bar{\mathcal{N}}_k} c_{\ell k}(n)\mathbf{w}_\ell(n-1). \qquad (7)$$

with adaptive combiners $c_{\ell k}(n)$ selected to minimize (5).

In the adaptation phase (6), an updated local estimation $\boldsymbol{\psi}_k(n)$ is calculated as a function of the previous local estimation $\boldsymbol{\psi}_k(n-1)$, local data $\{d_k(n), \mathbf{u}_k(n)\}$ and a state vector $\boldsymbol{\eta}_k$. In the combination phase (7), each node calculates $\mathbf{w}_k(n)$ using time-varying combination coefficients $c_{\ell k}(n)$.

Two conditions are applied in the adaptation of $c_{\ell k}(n)$. Firstly, as most adaptive filtering schemes converge to unbiased estimations of the optimal solution in stationary scenarios, i.e., $\mathbb{E}\{\mathbf{w}_o(n) - \boldsymbol{\psi}_k(n)\} \to 0$ as $n \to \infty$, we constrain all coefficients at each node to sum up to one, in order to keep combined weights estimations unbiased in steady state. In addition to this, to guarantee mean stability of our scheme, and in contrast with our previous work [33], we also impose non-negativity constraints on such combiners,

$$c_{\ell k}(n) \geq 0, \ \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(n) = 1, \forall k. \qquad (8)$$

These conditions on combination coefficients have also been considered in other diffusion schemes available in the literature to guarantee certain stability properties.

The adaptation and combination stages on the proposed diffusion algorithm can be interpreted in the following way: with respect to the adaptation of local estimates $\boldsymbol{\psi}_k(n)$, each node could be considered as an isolated adaptive filter working independently from the rest of the network. Thus, it pursues the minimization of $\text{MSE}_k(n)$ using just its own regressors.

During the combination phase, which includes the computation of combined weight estimators $\mathbf{w}_k(n)$ and the adaptation of the combiners, minimization of the network MSE is pursued.

We should emphasize that the most significant difference of (6) and (7) with respect to standard ATC is that the weight vector resulting from the combination is not fed back to the update of each individual filter. Even though under some circumstances this feedback can be beneficial, for instance to reduce the steady-state error in homogeneous networks, it also dilutes the differences between individual filter performances, which are key to get the maximum advantage out of heterogeneous networks. Furthermore, local estimates differ more among nodes than combined estimates, what results in a better conditioning for the model-free MSE-based adjustment of network combiners.

Since this diffusion scheme keeps the updates at each node decoupled from the rest of the network, we will refer to it in the following as *Decoupled ATC* (D-ATC). There are some additional advantages of decoupling the adaptation step from the combination phase:

- Since nodes are updated as if they were working isolated from the network, the analysis of the local estimates can rely on existing models for adaptive filters.
- Decoupled adaptation simplifies the design of heterogeneous networks, thus making it easy to include nodes that use different learning rules (e.g., LMS and RLS, Recursive Least-Squares), different learning parameters (e.g., step sizes, or asymmetry parameters in sparsity-aware nodes), or different filter lengths (using zero-padding for the shorter nodes).
- Related to this, the adaptation phase of our scheme is not influenced by an erroneous selection of the combination weights. In contrast, in standard ATC, if the combination weights are suboptimal, the adaptation phase of the diffusion algorithm is also affected.
- Since the adaptation of each node is completely independent of other nodes' adaptation, we can more easily deal with synchronization issues. Furthermore, the combination stage can be modified to include the last available estimates received from the neighbors so that a delay in a particular node does not slow down the network.

IV. Theoretical Analysis of D-ATC

In this section, we analyze the performance of the D-ATC diffusion strategy in the mean and mean-square sense and derive expressions for the steady-state NMSD in stationary and nonstationary environments. Different from [33], [37] and thanks to the energy conservation method [35], we directly obtain steady-state results, bypassing several of the difficulties encountered when obtaining them as a limiting case of a transient analysis. In order to simplify the analysis, the combiners $c_{\ell k}(n)$ are assumed to be static. Finally, for this analysis we consider LMS and NLMS adaptations and consequently the general equation (6) becomes

$$\boldsymbol{\psi}_k(n) = \boldsymbol{\psi}_k(n-1) + \mu_k(n)\mathbf{u}_k(n)e_k(n), \tag{9}$$

where the local estimation error signals are

$$e_k(n) = d_k(n) - \mathbf{u}_k^T(n)\boldsymbol{\psi}_k(n-1) \triangleq d_k(n) - y_k(n), \tag{10}$$

with $y_k(n)$ being the local output, and $\mu_k(n)$ a step size. For LMS, we have a constant step size $\mu_k(n) = \mu_k$, whereas for NLMS $\mu_k(n) = \tilde{\mu}_k/[\delta + \|\mathbf{u}_k(n)\|^2]$, with $0 < \tilde{\mu}_k < 2$, with δ a regularization factor to prevent division by zero.

A. Data Model and Definitions

We start by introducing several assumptions to make the analysis more tractable. In order to obtain the most general results, during our analysis, we will delay the application of the different assumptions as much as possible.

A1) The unknown parameter vector $\mathbf{w}_{\mathrm{o}}(n)$ follows a *random-walk model* [35]. According to this widespread model, the optimal solution varies in a nonstationary environment as

$$\mathbf{w}_{\mathrm{o}}(n) = \mathbf{w}_{\mathrm{o}}(n-1) + \mathbf{q}(n), \tag{11}$$

where $\mathbf{q}(n)$ is a zero-mean, independent and identically distributed (i.i.d.) vector with autocorrelation matrix $\mathbf{Q} = \mathbb{E}\{\mathbf{q}(n)\mathbf{q}^T(n)\}$, independent of the initial conditions $\boldsymbol{\psi}_k(0)$, $\mathbf{w}_k(0)$, and of $\{\mathbf{u}_k(n'), v_k(n')\}$ for all k and n'. Although this model implies that the covariance matrix of $\mathbf{w}_{\mathrm{o}}(n)$ diverges as $n \to \infty$, it has been commonly used in the literature to keep the analysis of adaptive systems simpler [35]. For $\mathbf{Q} = \mathbf{0}_M$ all expressions in the subsequent analysis are particularized for the stationary case.

A2) Input regressors are zero-mean and have covariance matrix $\mathbf{R}_k = \mathbb{E}\{\mathbf{u}_k(n)\mathbf{u}_k^T(n)\}$. Furthermore, they are spatially independent, i.e.,

$$\mathbb{E}\{\mathbf{u}_k(n)\mathbf{u}_\ell^T(n)\} = \mathbf{0}_M, \quad k \neq \ell.$$

This assumption is widely employed in the analysis of diffusion algorithms and is realistic in many practical applications [4]. Furthermore, the noise processes $\{v_k(n)\}$ are assumed to be temporally white and spatially independent,

$$\mathbb{E}\{v_k(n)v_k(n')\} = 0, \quad \text{for all } \; n \neq n',$$
$$\mathbb{E}\{v_k(n)v_\ell(n')\} = 0, \quad \text{for all } \; n, n' \; \text{ whenever } \; k \neq \ell.$$

Additionally, noise is assumed to be independent (not only uncorrelated) of the regression data $\mathbf{u}_\ell(n')$, so that $\mathrm{E}\{v_k(n)\mathbf{u}_\ell(n')\} = \mathbf{0}_M$, for all k, ℓ, n, and n'. As a result, $\boldsymbol{\psi}_k(n-1)$ is independent of $v_\ell(n)$ for all k and ℓ. Since the regressors are assumed spatially independent, $\boldsymbol{\psi}_k(n-1)$ is also independent of $\mathbf{u}_\ell(n)$ for $k \neq \ell$. For $k = \ell$, this independence condition also holds if the regressors are temporally uncorrelated.

A3) We will finally assume sufficiently small step sizes to neglect the effects of the statistical dependence of $\boldsymbol{\psi}_k(n-1)$ and $\mathbf{u}_k(n)$ for colored regressors. This assumption has also been widely used in analyses of diffusion schemes [4], [5], [25], [26], [30]. Furthermore, results obtained from the independence assumption between $\boldsymbol{\psi}_k(n-1)$ and $\mathbf{u}_k(n)$ tend to match reasonably well the real filter performance for sufficiently small step sizes, even when the temporal whiteness condition on the regression data does not hold (see e.g., [35]).

To analyze adaptive diffusion strategies, it is usual to define weight-error vectors, taking into account the local and combined estimates of each node, i.e.,

$$\widetilde{\boldsymbol{\psi}}_k(n) \triangleq \mathbf{w}_{\mathrm{o}}(n) - \boldsymbol{\psi}_k(n), \tag{12}$$

$$\widetilde{\mathbf{w}}_k(n) \triangleq \mathbf{w}_{\mathrm{o}}(n) - \mathbf{w}_k(n), \tag{13}$$

with $k = 1, \ldots, N$.

For notational convenience, we collect all weight-error vectors and products $v_k(n)\mathbf{u}_k(n)$ across the network into column vectors:

$$\widetilde{\boldsymbol{w}}(n) = \mathrm{col}\{\widetilde{\boldsymbol{\psi}}_1(n), \cdots, \widetilde{\boldsymbol{\psi}}_N(n), \widetilde{\mathbf{w}}_1(n), \cdots, \widetilde{\mathbf{w}}_N(n)\}, \tag{14}$$

$$\boldsymbol{s}(n) = \mathrm{col}\{v_1(n)\mathbf{u}_1(n), v_2(n)\mathbf{u}_2(n), \cdots, v_N(n)\mathbf{u}_N(n)\}, \tag{15}$$

where $\mathrm{col}\{\cdot\}$ represents the vector obtained by stacking its entries on top of each other. Note that the length of $\widetilde{\boldsymbol{w}}(n)$ is equal to $2MN$, whereas the length of $\boldsymbol{s}(n)$ is MN. We also define the $(2MN)$-length column vector

$$\mathbf{q}_a(n) = \mathrm{col}\{\mathbf{q}(n), \mathbf{q}(n), \cdots, \mathbf{q}(n)\}, \tag{16}$$

and the following $MN \times MN$ block-diagonal matrices containing the step sizes and information related to the autocorrelation matrices of the regressors:

$$\boldsymbol{\mathcal{M}}(n) = \mathrm{diag}\{\mu_1(n)\mathbf{I}_M, \mu_2(n)\mathbf{I}_M, \cdots, \mu_N(n)\mathbf{I}_M\}, \tag{17}$$

$$\boldsymbol{\mathcal{R}}(n) = \mathrm{diag}\{\mathbf{u}_1(n)\mathbf{u}_1^T(n), \cdots, \mathbf{u}_N(n)\mathbf{u}_N^T(n)\}, \tag{18}$$

where $\mathrm{diag}\{\cdot\}$ generates a block-diagonal matrix from its arguments and $\mathbf{I}_M$ is the $M \times M$ identity matrix. Finally, we also define the following matrices containing the combination weights:

$$\mathbf{C}_1 = \mathrm{diag}\{c_{11}, c_{22}, \cdots, c_{NN}\}, \tag{19}$$

$$\mathbf{C}_2 = \begin{bmatrix} 0 & c_{12} & \cdots & c_{1N} \\ c_{21} & 0 & \cdots & c_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \cdots & 0 \end{bmatrix}, \tag{20}$$

and their extended versions

$$\boldsymbol{\mathcal{C}}_i \triangleq \mathbf{C}_i \otimes \mathbf{I}_M, \quad i = 1, 2, \tag{21}$$

where $\otimes$ represents the Kronecker product of two matrices.

As a measure of performance, we consider the steady-state MSD_k at each node and the steady-state NMSD, as defined in (4).

B. Mean Stability Analysis

First, we present the mean convergence and stability analysis of our scheme. To do so, we start subtracting both sides of (7) and (9) from $\mathbf{w}_{\mathrm{o}}(n)$. Under Assumption **A1**, using (1) and recalling that $c_{kk} + \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k} = 1$, we obtain

$$\widetilde{\boldsymbol{\psi}}_k(n) - \mathbf{q}(n) = \mathbf{A}_k(n)\widetilde{\boldsymbol{\psi}}_k(n-1) - \mu_k(n)v_k(n)\mathbf{u}_k(n), \tag{22}$$

$$\widetilde{\mathbf{w}}_k(n) - \mathbf{q}(n) = c_{kk}\mathbf{A}_k(n)\widetilde{\boldsymbol{\psi}}_k(n-1) + \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k}\widetilde{\mathbf{w}}_\ell(n-1) - c_{kk}\mu_k(n)v_k(n)\mathbf{u}_k(n), \tag{23}$$

where $\mathbf{A}_k(n) \triangleq \mathbf{I}_M - \mu_k(n)\mathbf{u}_k(n)\mathbf{u}_k^T(n)$.

From (22) and (23), using definitions (14)–(21), and following algebraic manipulations similar to those of [4], we obtain the following equation characterizing the evolution of the weight-error vectors:

$$\widetilde{\boldsymbol{w}}(n) - \mathbf{q}_a(n) = \boldsymbol{\mathcal{B}}(n)\widetilde{\boldsymbol{w}}(n-1) - \boldsymbol{z}(n), \tag{24}$$

where

$$\boldsymbol{\mathcal{B}}(n) \triangleq \begin{bmatrix} \boldsymbol{\mathcal{B}}_{11}(n) & \mathbf{0}_{(MN)} \\ \boldsymbol{\mathcal{B}}_{21}(n) & \boldsymbol{\mathcal{B}}_{22} \end{bmatrix},$$

$$\boldsymbol{\mathcal{B}}_{11}(n) = \mathbf{I}_{(MN)} - \boldsymbol{\mathcal{M}}(n)\boldsymbol{\mathcal{R}}(n),$$

$$\boldsymbol{\mathcal{B}}_{21}(n) = \boldsymbol{\mathcal{C}}_1^T[\mathbf{I}_{(MN)} - \boldsymbol{\mathcal{M}}(n)\boldsymbol{\mathcal{R}}(n)],$$

$$\boldsymbol{\mathcal{B}}_{22} = \boldsymbol{\mathcal{C}}_2^T,$$

$$\boldsymbol{z}(n) \triangleq [\boldsymbol{\mathcal{M}}(n)\boldsymbol{s}(n) \quad \boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{M}}(n)\boldsymbol{s}(n)]^T.$$

Under Assumptions **A2** and **A3**, all regressor vectors $\mathbf{u}_k(n)$ are independent of $\widetilde{\boldsymbol{\psi}}_\ell(n-1)$ and $\widetilde{\mathbf{w}}_\ell(n-1)$ for $k, \ell = 1, 2, \cdots, N$. Furthermore, independence of the noise w.r.t. the rest of variables implies that $\mathbb{E}\{\boldsymbol{s}(n)\} = \mathbf{0}_M$ and $\mathbb{E}\{\boldsymbol{z}(n)\} = \mathbf{0}_M$. Thus, taking expectations on both sides of (24) and recalling that $\mathbb{E}\{\mathbf{q}_a(n)\} = \mathbf{0}_{2MN}$, we obtain

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}(n)\} = \mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}\mathbb{E}\{\widetilde{\boldsymbol{w}}(n-1)\}. \tag{25}$$

A necessary and sufficient condition for the mean stability of (25) is that the spectral radius of $\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$ is less than one, i.e.,

$$\rho(\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}) = \max_i\{\lambda_i\} < 1,$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument and λ_i, with $i = 1, 2, \ldots, 2MN$, are the eigenvalues of $\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$ [4]. Since $\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$ is a block-triangular matrix, its eigenvalues are the eigenvalues of the blocks of its main diagonal, i.e., the eigenvalues of $\mathbb{E}\{\boldsymbol{\mathcal{B}}_{11}(n)\}$ and $\mathbb{E}\{\boldsymbol{\mathcal{B}}_{22}\}$ [38].

Focusing first on matrix $\mathbb{E}\{\boldsymbol{\mathcal{B}}_{11}(n)\}$, we notice that it is also a block-diagonal matrix, so the step sizes need to be selected to guarantee

$$\rho(\mathbb{E}\{\boldsymbol{\mathcal{B}}_{11}(n)\}) = \max_{1 \le k \le N} \rho\left(\mathbf{I}_M - \overline{\mathbf{R}}_k\right) < 1, \tag{26}$$

where

$$\overline{\mathbf{R}}_k \triangleq \mathbb{E}\left\{\mu_k(n)\mathbf{u}_k(n)\mathbf{u}_k^T(n)\right\}. \tag{27}$$

For LMS, this matrix reduces to

$$\overline{\mathbf{R}}_k = \mu_k\mathbb{E}\left\{\mathbf{u}_k(n)\mathbf{u}_k^T(n)\right\} = \mu_k\mathbf{R}_k \tag{28}$$

and for NLMS, we have

$$\overline{\mathbf{R}}_k = \tilde{\mu}_k \mathbb{E}\left\{\frac{\mathbf{u}_k(n)\mathbf{u}_k^T(n)}{\delta + \|\mathbf{u}_k(n)\|^2}\right\}. \tag{29}$$

Condition (26) will be ensured for LMS if the step sizes μ_k satisfy [35]

$$0 < \mu_k < \frac{2}{\lambda_{\max}(\mathbf{R}_k)}, \quad \text{for } k = 1, 2, \cdots, N, \tag{30}$$

in terms of the largest eigenvalue of $\mathbf{R}_k$. Similarly, Condition (26) will be ensured for NLMS if the step sizes $\tilde{\mu}_k$ satisfy [35]

$$0 < \tilde{\mu}_k < 2, \quad \text{for } k = 1, 2, \cdots, N. \tag{31}$$

Conditions (30) and (31), which are well-known results for the LMS and NLMS algorithms, respectively, guarantee that the local estimators $\{\boldsymbol{\psi}_k(n)\}$ are asymptotically unbiased, i.e., $\mathbb{E}\{\widetilde{\boldsymbol{\psi}}_k(n)\} \to \mathbf{0}_M$ as $n \to \infty$ for all nodes of the network.

For the spectral radius of $\boldsymbol{\mathcal{B}}_{22} = \boldsymbol{\mathcal{C}}_2^T$, we can rely on the following bound from [38]:

$$\rho(\boldsymbol{\mathcal{B}}_{22}) \leq \|\boldsymbol{\mathcal{B}}_{22}\|_\infty = \max_k \sum_{\ell \in \mathcal{N}_k} |c_{\ell k}|. \tag{32}$$

A sufficient (but not necessary) condition to guarantee $\rho(\boldsymbol{\mathcal{B}}_{22}) \leq 1$ is to keep all combination weights non-negative. In effect, since the sum of all combiners associated to a node is one, using non-negative weights we have

$$\rho(\boldsymbol{\mathcal{B}}_{22}) \leq \max_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} = \max_k (1 - c_{kk}) \leq 1. \tag{33}$$

When combiners are learned by the network, non-negativity constraints can be applied at every iteration to ensure mean stability. Although our derivations show that this is just a sufficient condition, we should mention that in our previous simulation work of [33], [34], where we allowed combination weights to become negative, the network showed some instability problems that have been removed thanks to the application of these constraints.

C. Mean-Square Performance

We present next a mean-square performance analysis, following the energy conservation framework of [35]. First, let $\boldsymbol{\Sigma}$ denote an arbitrary nonnegative definite $2MN \times 2MN$ matrix. Different choices of $\boldsymbol{\Sigma}$ allow us to obtain different performance measurements of the network [5].

Thus, computing the weighted squared norm on both sides of (24) using $\boldsymbol{\Sigma}$ as a weighting matrix, we arrive at

$$\begin{aligned}\widetilde{\boldsymbol{w}}^T(n)\boldsymbol{\Sigma}\widetilde{\boldsymbol{w}}(n) - \widetilde{\boldsymbol{w}}^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n) - \mathbf{q}_a^T(n)\boldsymbol{\Sigma}\widetilde{\boldsymbol{w}}(n) + \mathbf{q}_a^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n)\\ = \widetilde{\boldsymbol{w}}^T(n-1)\boldsymbol{\mathcal{B}}^T(n)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(n)\widetilde{\boldsymbol{w}}(n-1) + \boldsymbol{z}^T(n)\boldsymbol{\Sigma}\boldsymbol{z}(n)\\ - 2\boldsymbol{z}^T(n)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(n)\widetilde{\boldsymbol{w}}(n-1).\end{aligned} \tag{34}$$

As before, independence of the noise terms in $\boldsymbol{z}(n)$ with respect to all other variables implies that the last element in (34) vanishes under expectation. Furthermore, under Assumption **A1**, we can verify that

$$\begin{aligned}\mathbb{E}\{\widetilde{\boldsymbol{w}}^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n)\} = \mathbb{E}\{\mathbf{q}_a^T(n)\boldsymbol{\Sigma}\widetilde{\boldsymbol{w}}(n)\} = \mathbb{E}\{\mathbf{q}_a^T(n)\boldsymbol{\Sigma}\mathbf{q}_a(n)\}\\ = \mathrm{Tr}(\boldsymbol{\Sigma}\mathbf{Q}_a),\end{aligned} \tag{35}$$

where $\mathrm{Tr}(\cdot)$ stands for the trace of a matrix and

$$\mathbf{Q}_a \triangleq \mathbb{E}\{\mathbf{q}_a(n)\mathbf{q}_a^T(n)\} = \mathbf{J}_{(2N)} \otimes \mathbf{Q},$$

being $\mathbf{J}_{(2N)}$ a $2N \times 2N$ matrix with all entries equal to one. Defining the matrices

$$\begin{aligned}\boldsymbol{\mathcal{S}} \triangleq \mathrm{diag}\Big\{&\sigma_{v_1}^2 \mathbb{E}\{\mu_1^2(n)\mathbf{u}_1(n)\mathbf{u}_1^T(n)\},\\ &\sigma_{v_2}^2 \mathbb{E}\{\mu_2^2(n)\mathbf{u}_2(n)\mathbf{u}_2^T(n)\}, \ \cdots,\\ &\sigma_{v_N}^2 \mathbb{E}\{\mu_N^2(n)\mathbf{u}_N(n)\mathbf{u}_N^T(n)\}\Big\},\end{aligned} \tag{36}$$

$$\boldsymbol{\mathcal{Z}} \triangleq \mathbb{E}\{\boldsymbol{z}(n)\boldsymbol{z}^T(n)\} = \begin{bmatrix}\boldsymbol{\mathcal{S}} & \boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1\\ \boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}} & \boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1\end{bmatrix}, \tag{37}$$

using (35), and taking expectations of both sides of (34), we obtain

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|_{\boldsymbol{\Sigma}}^2\} = \ &\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}(n-1)\|_{\boldsymbol{\mathcal{B}}^T(n)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(n)}^2\right\}\\ &+ \mathrm{Tr}(\boldsymbol{\Sigma}\boldsymbol{\mathcal{Z}}) + \mathrm{Tr}(\boldsymbol{\Sigma}\mathbf{Q}_a),\end{aligned} \tag{38}$$

where $\|\mathbf{x}\|_{\boldsymbol{\Sigma}}^2$ denotes the weighted squared norm $\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}$.

Using Assumption **A3**, we can replace the random matrix $\boldsymbol{\mathcal{B}}(n)$ by its steady-state mean value $\overline{\boldsymbol{\mathcal{B}}} = \lim_{n\to\infty}\mathbb{E}\{\boldsymbol{\mathcal{B}}(n)\}$, which is equivalent to replacing the matrix $\mu_k(n)\mathbf{u}_k(n)\mathbf{u}_k^T(n)$ by its mean $\overline{\mathbf{R}}_k$, given by (28) for LMS or by (29) for NLMS. Using this approximation, (38) reduces to

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|_{\boldsymbol{\Sigma}}^2\} \approx \ &\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}(n-1)\|_{\overline{\boldsymbol{\mathcal{B}}}^T\boldsymbol{\Sigma}\overline{\boldsymbol{\mathcal{B}}}}^2\right\}\\ &+ \mathrm{Tr}(\boldsymbol{\Sigma}\boldsymbol{\mathcal{Z}}) + \mathrm{Tr}(\boldsymbol{\Sigma}\mathbf{Q}_a).\end{aligned} \tag{39}$$

Mean-Square Convergence: As in [5], the convergence rate of the series is governed by $[\rho(\overline{\boldsymbol{\mathcal{B}}})]^2$, in terms of the spectral radius of $\overline{\boldsymbol{\mathcal{B}}}$. From Section IV-B, we can obtain a superior limit for $\rho(\overline{\boldsymbol{\mathcal{B}}})$, which is given by

$$\rho(\overline{\boldsymbol{\mathcal{B}}}) \leq \max\left\{\max_{k,i}\left[1 - \lambda_i(\overline{\mathbf{R}}_k)\right], \ \max_k(1 - c_{kk})\right\}. \tag{40}$$

Choosing the step size of the LMS (resp., NLMS) algorithm into the interval (30) [resp., (31)] and imposing non-negativity constraints to the combiners, $\rho(\overline{\boldsymbol{\mathcal{B}}}) \leq 1$ and the convergence of $\lim_{n\to\infty}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|_{\boldsymbol{\Sigma}}^2\}$ is ensured. Furthermore, from the superior limit (40), we can see that, in the worst case, our diffusion scheme can converge with the same convergence rate of the noncooperative solution, whose spectral radius is $\max_{k,i}\{1 - \lambda_i(\overline{\mathbf{R}}_k)\}$ (considering that all the nodes are adapted using LMS or NLMS). However, we show by means of simulations that in practice this limit is very conservative and the proposed diffusion scheme converges much faster than the noncooperative solution.

Steady-State MSD Performance: It it important to notice that variance relations similar to (39) have often appeared in the performance analysis of diffusion schemes [5]. Iterating (39)

and taking the limit as $n \to \infty$, we conclude that (see, e.g., [24])

$$\lim_{n\to\infty} \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(n)\|^2_{\boldsymbol{\Sigma}}\} \approx \sum_{j=0}^{\infty} \mathrm{Tr}[\overline{\boldsymbol{\mathcal{B}}}^j(\boldsymbol{\mathcal{Z}}+\mathbf{Q}_a)(\overline{\boldsymbol{\mathcal{B}}}^T)^j\boldsymbol{\Sigma}]. \quad (41)$$

To obtain analytical expressions for the steady-state MSD of the network and of its individual nodes, we will replace $\boldsymbol{\Sigma}$ by the following matrices

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} \mathbf{0}_{NM} & \mathbf{0}_{NM} \\ \mathbf{0}_{NM} & \frac{1}{N}\mathbf{I}_{NM} \end{bmatrix}, \quad (42)$$

$$\boldsymbol{\Upsilon}_k \triangleq \begin{bmatrix} \mathbf{0}_{NM} & \mathbf{0}_{NM} \\ \mathbf{0}_{NM} & \mathbf{E}_k \otimes \mathbf{I}_M \end{bmatrix}, \quad (43)$$

where $\mathbf{E}_k$ is an $N \times N$ zero matrix, except in the element (k,k), that is equal to one. Replacing $\boldsymbol{\Sigma}$ in (41) by either $\boldsymbol{\Gamma}$ or $\boldsymbol{\Upsilon}_k$, the MSD performance of the network and of its individual nodes can be expressed, respectively, by

$$\mathrm{NMSD}(\infty) \approx \sum_{j=0}^{\infty} \mathrm{Tr}[\overline{\boldsymbol{\mathcal{B}}}^j(\boldsymbol{\mathcal{Z}}+\mathbf{Q}_a)(\overline{\boldsymbol{\mathcal{B}}}^T)^j\boldsymbol{\Gamma}], \quad (44)$$

$$\mathrm{MSD}_k(\infty) \approx \sum_{j=0}^{\infty} \mathrm{Tr}[\overline{\boldsymbol{\mathcal{B}}}^j(\boldsymbol{\mathcal{Z}}+\mathbf{Q}_a)(\overline{\boldsymbol{\mathcal{B}}}^T)^j\boldsymbol{\Upsilon}_k]. \quad (45)$$

Since $\overline{\boldsymbol{\mathcal{B}}}$ is lower triangular, matrix $\overline{\boldsymbol{\mathcal{B}}}^j$ is given by

$$\overline{\boldsymbol{\mathcal{B}}}^j = \begin{bmatrix} \overline{\boldsymbol{\mathcal{B}}}^j_{11} & \mathbf{0}_{(MN)} \\ \bar{\boldsymbol{\mathcal{X}}}(j) & \overline{\boldsymbol{\mathcal{B}}}^j_{22} \end{bmatrix}, \quad (46)$$

being

$$\bar{\boldsymbol{\mathcal{X}}}(j) = \sum_{k=0}^{j-1} \overline{\boldsymbol{\mathcal{B}}}^k_{22}\overline{\boldsymbol{\mathcal{B}}}_{21}\overline{\boldsymbol{\mathcal{B}}}^{j-k-1}_{11} = \sum_{k=0}^{j-1} [\boldsymbol{\mathcal{C}}_2^T]^k \boldsymbol{\mathcal{C}}_1^T \left[\mathbf{I}_{(MN)} - \boldsymbol{\mathcal{L}}\right]^{j-k}, \quad (47)$$

where we have defined

$$\boldsymbol{\mathcal{L}} \triangleq \lim_{n\to\infty} \mathbb{E}\{\boldsymbol{\mathcal{M}}(n)\boldsymbol{\mathcal{R}}(n)\} = \mathrm{diag}\{\overline{\mathbf{R}}_1, \overline{\mathbf{R}}_2, \cdots, \overline{\mathbf{R}}_N\}. \quad (48)$$

Replacing (46) and (37) respectively in (44) and (45), we arrive at

$$\begin{aligned} \mathrm{NMSD}(\infty) \approx \frac{1}{N}\sum_{j=0}^{\infty} \mathrm{Tr}\Big[& \bar{\boldsymbol{\mathcal{X}}}(j)(\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j) \\ & + 2(\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j) \\ & + (\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1+\boldsymbol{\mathcal{Q}})\boldsymbol{\mathcal{C}}_2^j \Big], \end{aligned} \quad (49)$$

$$\begin{aligned} \mathrm{MSD}_k(\infty) \approx \sum_{j=0}^{\infty} \mathrm{Tr}\Big[& \Big(\bar{\boldsymbol{\mathcal{X}}}(j)(\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j) \\ & + 2(\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}+\boldsymbol{\mathcal{Q}})\bar{\boldsymbol{\mathcal{X}}}^T(j) \\ & + (\boldsymbol{\mathcal{C}}_2^T)^j(\boldsymbol{\mathcal{C}}_1^T\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}_1+\boldsymbol{\mathcal{Q}})\boldsymbol{\mathcal{C}}_2^j\Big)\mathbf{E}_k \otimes \mathbf{I}_M \Big], \end{aligned} \quad (50)$$

where $\boldsymbol{\mathcal{Q}} = \mathbf{J}_N \otimes \mathbf{Q}$. Note that the $MN \times MN$ matrix $\boldsymbol{\mathcal{Q}}$ is similar to matrix $\mathbf{Q}_a$, but has half its size.

If all the nodes of the network update their local estimates with the LMS algorithm, the theoretical steady-state MSD can be estimated by (49) and (50), recalling that matrices $\overline{\mathbf{R}}_k$, which appear in $\bar{\boldsymbol{\mathcal{X}}}(j)$, are given by (28) and that matrix $\boldsymbol{\mathcal{S}}$ reduces to

$$\boldsymbol{\mathcal{S}} = \mathrm{diag}\left\{\sigma^2_{v_1}\mu_1^2\mathbf{R}_1,\ \sigma^2_{v_2}\mu_2^2\mathbf{R}_2,\ \cdots,\ \sigma^2_{v_N}\mu_N^2\mathbf{R}_N\right\}. \quad (51)$$

On the other hand, assuming NLMS adaptation, we still have to obtain approximations for matrices $\overline{\mathbf{R}}_k$ and $\mathbb{E}\{\mu_k^2(n)\mathbf{u}_k(n)\mathbf{u}_k^T(n)\}$. For this purpose, we assume:

A4) The number of coefficients M is large enough for each element of the matrix $\mathbf{u}_k(n)\mathbf{u}_k^T(n)$ to be approximately independent from $\sum_{l=0}^{M-1}|u(n-l)|^2$. This is equivalent to applying the averaging principle of [39], since for large M, $\|\mathbf{u}_k(n)\|^2$ tends to vary slowly compared to the individual entries of $\mathbf{u}_k(n)\mathbf{u}_k^T(n)$.

A5) The regressors $\mathbf{u}_k(n)$, $k = 1, 2, \ldots, N$ are formed by a tapped-delay line with Gaussian entries and the regularization factor is equal to zero ($\delta = 0$). This is a common assumption in the analysis of adaptive filters and leads to reasonable analytical results [40]. Under **A4** and **A5**, we obtain the following approximations from [41]:

$$\overline{\mathbf{R}}_k \approx \tilde{\mu}_k \frac{\mathbf{R}_k}{\sigma^2_{u_k}(M-2)}, \quad (52)$$

$$\mathbb{E}\{\mu_k^2(n)\mathbf{u}_k(n)\mathbf{u}_k^T(n)\} \approx \tilde{\mu}_k^2 \frac{\mathbf{R}_k}{\sigma^4_{u_k}(M-2)(M-4)}. \quad (53)$$

The model to compute the steady-state MSD of the network and of its individual nodes can be summarized as follows: (i) compute the matrices of the combination weights using (19)-(21) and the matrix $\boldsymbol{\mathcal{Q}}$, according to the environment variation; (ii) for LMS (resp., NLMS) adaptation, use (28) [resp., (52) and (53)] in the computation of matrices $\boldsymbol{\mathcal{S}}$ and $\bar{\boldsymbol{\mathcal{X}}}(j)$, defined respectively by (36) and (47); and finally, (iii) use these matrices in (49) and (50).

V. NMSE-Based Adaptive Combiners

As shown in Section I, the implementation of adaptive combiners is crucial for heterogeneous networks. For instance, when the nodes have different step sizes in the adaptation step, the combiners should favor the diffusion of the estimates of the fastest nodes during network convergence. However, the network should favor the nodes with better SNR and smaller adaptation step size in steady state, as they produce lower steady-state misadjustment.

In this section we present two strategies for learning the combiners suitable for our Decoupled ATC scheme. These two strategies are based on an approximate minimization of the network Mean-Square Error at each step n as defined in (5). Since every node only optimizes its own combination coefficients, this is equivalent to minimizing $\mathrm{MSE}_k(n)$ node-wise. As stated in Section II-B, there are different well-known algorithms that can be used to optimize $\mathrm{MSE}_k(n)$ including gradient-based or LS strategies. However, it should be remarked that due to the nature of the problem, in particular because of the expected large correlation among the solution estimates shared by the nodes, not all adaptive algorithms to update $c_{\ell k}(n)$ would obtain a competitive performance. In this work, we include two approaches to

adapt the combination coefficients that have demonstrated their benefits with respect to other schemes.

Finally, let us recall that we would like to satisfy the convexity constraint of Section IV-B to guarantee stability (and also to follow the criterion of other works in this field, e.g., [4], [25], [26], [29], [30]). Since a direct application of the algorithms below may give rise to values of $c_{\ell k}(n)$ outside range $[0,1]$, we will enforce the combination parameters $c_{\ell k}(n)$ to remain in the desired interval $[0,1]$ at each iteration. For simplicity, if any $c_{\ell k}(n)$ results negative after its update, we simply set it to zero and then rescale the remaining combination weights so that they sum up to one. We would like to remark that more complex projection rules could have been used to implement this constraint but, as the proposed method shows a good performance, we leave as future work the analysis and evaluation of alternative solutions.

A. Affine Projection Algorithm

In this section we present an Affine Projection Algorithm (APA) for the stochastic minimization of the MSE in (5).

First, it is useful to define some notation. We stack the combination coefficients $c_{\ell k}(n)$ of node k, with $\ell \in \bar{\mathcal{N}}_k$, in a length-$\bar{N}_k$ vector $\bar{\mathbf{c}}_k(n)$. Doing so, we can write

$$c_{kk}(n) = 1 - \sum_{\ell \in \bar{\mathcal{N}}_k} c_{\ell k}(n) = 1 - \mathbb{1}_{\bar{N}_k}^T \bar{\mathbf{c}}_k(n). \tag{54}$$

Then, defining $y_{\ell k}(n) = \mathbf{u}_k^T(n)\mathbf{w}_\ell(n-1)$ and $\tilde{y}_{\ell k}(n) = y_{\ell k}(n) - y_k(n)$ with $\ell \in \bar{\mathcal{N}}_k$, collecting all these differences into a column vector $\tilde{\mathbf{y}}_k(n)$, and using (54), $\mathrm{MSE}_k(n)$ can be rewritten as

$$\mathrm{MSE}_k(n) = \mathbb{E}\left\{ \left[e_k(n) - \bar{\mathbf{c}}_k^T(n)\tilde{\mathbf{y}}_k(n) \right]^2 \right\}. \tag{55}$$

Applying the standard APA algorithm [35] to minimize this cost function, we obtain a regularized affine projection algorithm for the adaptation of $\bar{\mathbf{c}}_k(n)$:

$$\bar{\mathbf{c}}_k(n) = \bar{\mathbf{c}}_k(n-1) + \mu_c [\epsilon \mathbf{I}_{\bar{N}_k} + \tilde{\mathbf{Y}}_k^T(n)\tilde{\mathbf{Y}}_k(n)]^{-1} \tilde{\mathbf{Y}}_k^T(n) \times [\mathbf{e}_k(n) - \tilde{\mathbf{Y}}_k(n)\bar{\mathbf{c}}_k(n-1)], \tag{56}$$

where μ_c is a step size to control the adaptation of $\bar{\mathbf{c}}_k(n)$, ϵ is a small regularization parameter to prevent division by zero, $\tilde{\mathbf{Y}}_k(n)$ is an $L \times \bar{N}_k$ matrix whose L rows corresponds with the last L values of vector $\tilde{\mathbf{y}}_k(n)$, $\mathbf{e}_k(n) = [e_k(n), e_k(n-1), \cdots, e_k(n-L+1)]^T$, and $\mathbf{I}_{\bar{N}_k}$ represents the $\bar{N}_k \times \bar{N}_k$ identity matrix, with $\bar{N}_k$ the cardinal of $\bar{\mathcal{N}}_k$.

This recursion requires the inversion of an $\bar{N}_k \times \bar{N}_k$ matrix at each iteration, resulting in an attractive implementation if the projection order L is larger than the number of neighbors of node k, $\bar{N}_k$. Otherwise, if for any node $\bar{N}_k > L$, we can invoke the matrix inversion lemma [35] to rewrite (56) as

$$\bar{\mathbf{c}}_k(n) = \bar{\mathbf{c}}_k(n-1) + \mu_c \tilde{\mathbf{Y}}_k^T(n) [\epsilon \mathbf{I}_L + \tilde{\mathbf{Y}}_k(n)\tilde{\mathbf{Y}}_k^T(n)]^{-1} \times [\mathbf{e}_k(n) - \tilde{\mathbf{Y}}_k(n)\bar{\mathbf{c}}_k(n-1)], \tag{57}$$

which requires the inversion of an $L \times L$ matrix.

Equations (56) –or (57)– and (54), constitute the ϵ-APA algorithm for adapting the combiners at each node. More details about the derivation are provided in Appendix A.

B. Least-Squares Algorithm

In this section, we follow a Least-Squares approach similar to the one in [33] and [34]. Instead of minimizing (5) using a stochastic minimization algorithm, we replace $\mathrm{MSE}_k(n)$ by the following related cost function [35],

$$J_k(n) = \sum_{i=1}^{n} \beta(n,i) \check{e}_k^2(n,i), \tag{58}$$

where $\beta(n,i)$ is a temporal weighting window, and

$$\check{e}_k(n,i) = e_k(i) - \bar{\mathbf{c}}_k^T(n)\tilde{\mathbf{y}}_k(i) \tag{59}$$

represents the error incurred by node k at time i when the outputs of all nodes belonging to $\mathcal{N}_k$ are combined using the combiners at time n.

Following a standard LS method to minimize this cost function, we obtain

$$\bar{\mathbf{c}}_k(n) = \left(\mathbf{P}_k(n) + \epsilon \mathbf{I}_{\bar{N}_k} \right)^{-1} \mathbf{z}_k(n), \tag{60}$$

where a small regularization constant ϵ is again introduced since $\mathbf{P}_k(n)$ could be ill-conditioned [34]. Similarly to the case of combination of multiple filters [42], $\mathbf{P}_k(n)$ can be interpreted as the autocorrelation matrix of vector $\tilde{\mathbf{y}}_k(n)$ while $\mathbf{z}_k(n)$ would be seen as the cross-correlation vector between $\tilde{\mathbf{y}}_k(n)$ and $e_k(n)$.

For further details about the derivation of the LS algorithm, please refer to Appendix B.

Temporal Weighting Window: The temporal weighting window $\beta(n,i)$, in the cost function (58) and the computation of $\mathbf{P}_k(n)$ and $\mathbf{z}_k(n)$, deserves some discussion. In this paper, we propose the use of an exponential weighting window,

$$\beta(n,i) = \gamma^{n-i}, \tag{61}$$

where γ is a forgetting factor $0 < \gamma \leq 1$.

This contrasts with our choice in previous works [33], [34], where we leaned towards a rectangular window, which provided a good convergence but a worse steady-state performance than standard ATC with adaptive combiners [30]. The reason for that choice was the instability problems of affine combiners when long windows were used. In this paper, as we use the more stable convex combiners (see Section IV-B) an exponential window can be safely employed. This window has two remarkable advantages with respect to a rectangular window: 1) It is more efficient in terms of memory and computation; and 2) it allows a recursive implementation. In addition, as we show in the experiments in the next section, the LS algorithm with exponential window outperforms other state-of-the art approaches.

VI. Simulation Results

In this section we present a number of simulation results to illustrate the behavior of D-ATC and the proposed adaptive combiners rules in stationary estimation and tracking scenarios. In the simulations, we consider only the NLMS algorithm to update the nodes due to its inherent advantages with respect to LMS. Nevertheless, it should be remarked that we have carried out experiments where nodes are updated with the LMS algorithm obtaining similar conclusions.

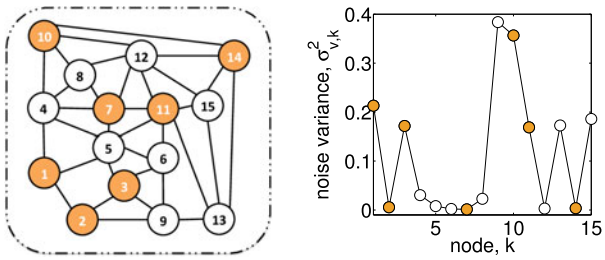


Fig. 4. (a) Network topology for the simulation experiments: orange shaded nodes are adapted with $\tilde{\mu}_k = 0.1$ and the rest with $\tilde{\mu}_k = 1$. (b) noise power $\sigma^2_{v,k}$ at each node in the network.

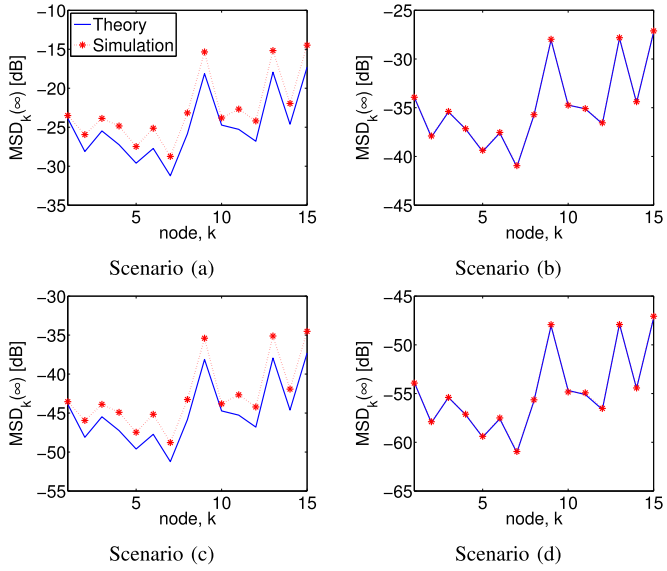


Fig. 5. Comparison between empirical performance and the theoretical model for the node steady-state MSD.

We simulate the 15-node network of Fig. 4(a), where all the nodes employ NLMS adaptation. The nodes step sizes are taken as $\mu_k \in \{0.1, 1\}$ as illustrated in Fig. 4. The input signals $\mathbf{u}_k(n)$ follow a multidimensional Gaussian with zero mean and the same scalar covariance matrix, $\sigma_u^2 \mathbf{I}_M$, with $\sigma_u^2 = 1$. Unless otherwise stated, the observation noise $v_k(n)$ at each node is also Gaussian distributed with zero mean and variance $\sigma^2_{v,k}$ randomly chosen between $[0, 0.4]$ as shown in Fig. 4(b). For the stationary estimation problem, the parameter vector $\mathbf{w}_o$ is a length-50 vector with values uniformly taken from range $[-1, 1]$. As a tracking model, we use the one presented in (11).

First, we present a set of experiments with the aim to validate the theoretical analysis of Section IV. Then, we compare the behavior of our rules to state-of-the-art adaptive combination algorithms for standard ATC [30], both in stationary and tracking scenarios.

A. Validation of the Theoretical Analysis for D-ATC

In the first place, we carry out some numerical simulations to validate the analysis of Section IV. To do so, we compare in Fig. 5 the theoretical and empirical steady-state MSD_k for the nodes of a D-ATC scheme with Metropolis combiners [4] in the stationary estimation scenario. Our objective in this section is just to show that the analysis correctly predicts the steady-state performance of each individual node, as well as the NMSD. Although we consider just the case of Metropolis combination rule, we have checked that other rules, e.g., uniform combiners, would lead to similar conclusions about the accuracy of the analysis. In Fig. 5, we plot the steady-state MSD for four different scenarios where the step sizes μ_k and the noise variances $\sigma^2_{v,k}$ have been varied from those in Fig. 4, according to Table II. From Fig. 5 we can conclude that the matching between the analysis and the simulation is quite good, even for not so small step sizes [scenarios (a) and (c)].

TABLE II
SCENARIOS SIMULATED IN FIG. 5

Scenario (a)	Scenario (b)	Scenario (c)	Scenario (d)
μ_k	$\mu_k/10$	μ_k	$\mu_k/10$
$\sigma^2_{v,k}$	$\sigma^2_{v,k}$	$\sigma^2_{v,k}/10$	$\sigma^2_{v,k}/10$

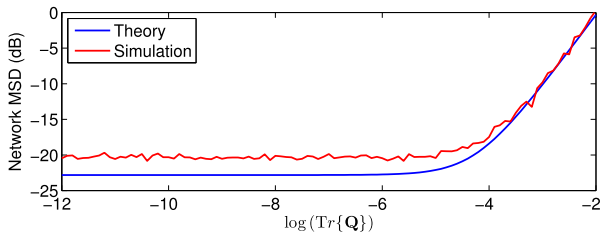


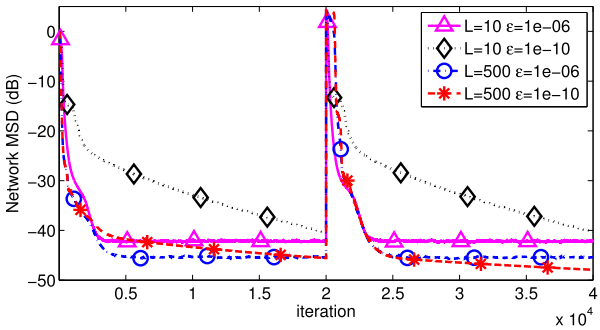
Fig. 6. Comparison between the analysis and the simulation in a tracking scenario with respect to the logarithm of $\text{Tr}\{\mathbf{Q}\}$.

We have also studied the accuracy of the model in tracking situations. In Fig. 6 we plot the steady-state NMSD for different speeds of change, i.e., values of $\text{Tr}\{\mathbf{Q}\}$. We can see that the matching is also quite good, especially for fast changes. For smaller $\text{Tr}\{\mathbf{Q}\}$ we observe a mismatch up to 2 dB, similarly to the stationary scenario depicted in Fig. 5(a).

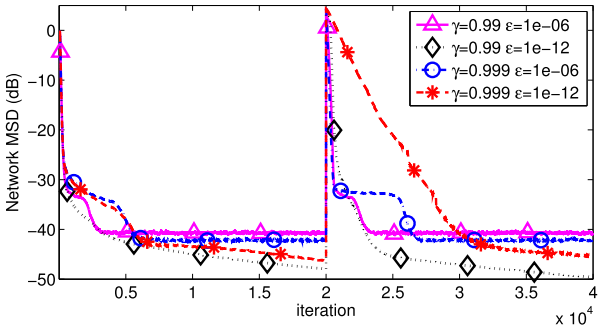
B. Stationary Performance of D-ATC With Adaptive Combiners

Before comparing the performance of D-ATC and ATC with adaptive combiners, we study in Fig. 7 the sensitivity of the proposed combiner learning rules, APA and LS, with respect to their settings. We observe that there is a trade-off between convergence/reconvergence speed and steady-state performance in the selection of these parameters. In fact, we can conclude that the influences of different parameters are coupled among them.

Regarding the forgetting factor γ in the LS rule, note that, when it is correctly chosen [see Fig. 7(b)], we can obtain a large steady-state enhancement hardly affecting the convergence. This was not the case with the rectangular window [34], where instability issues prevented us from using a very small

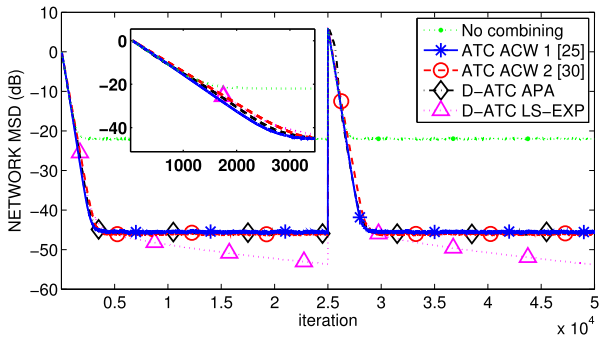


(a) D-ATC with APA adaptive combiners

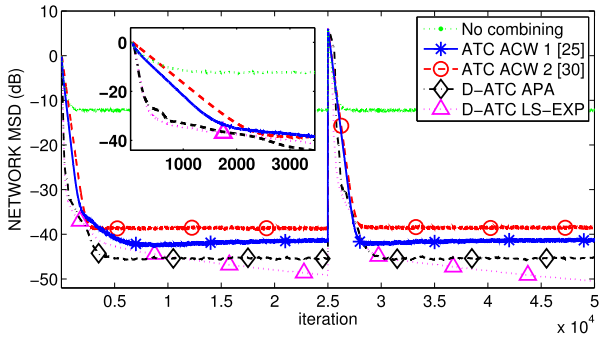


(b) D-ATC with LS adaptive combiners

Fig. 7. Parameter selection for D-ATC with (a) APA and (b) LS adaptive combiners.



(a) Homogeneous network



(b) Heterogeneous network

Fig. 8. Network MSD performance for a stationary estimation problem. A zoom of the first 3000 iterations is provided for a more clear illustration of the convergence properties of the algorithms.

TABLE III
SIMULATION PARAMETERS

	D-ATC APA	D-ATC LS	ATC ACW 1 [25]	ATC ACW 2 [30]
Stationary. Fig 8	$L = 500$ $\epsilon = 10^{-6}$ $\mu_c = 1$	$\gamma = 0.99$ $\epsilon = 10^{-12}$	$\alpha = 0.2$ $\epsilon = 10^{-6}$	$\nu = 0.1$
Tracking. Fig 9.(a)	$L = 10$ $\epsilon = 10^{-6}$ $\mu_c = 1$	$\gamma = 0.9999$ $\epsilon = 10^{-10}$	$\alpha = 0.05$ $\epsilon = 10^{-6}$	$\nu = 0.2$
Tracking. Fig 9.(b)	$L = 500$ $\epsilon = 10^{-6}$ $\mu_c = 1$	$\gamma = 0.99$ $\epsilon = 10^{-10}$	$\alpha = 0.2$ $\epsilon = 10^{-6}$	$\nu = 0.1$

regularization constant, and limited the number of useful window sizes, causing degradation in the steady-state performance.

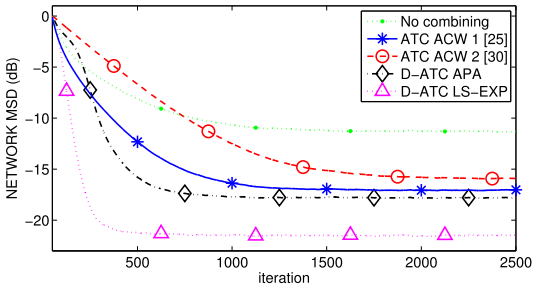
Next, we compare our D-ATC scheme with adaptive combiners, with other state-of-the-art ATC algorithms with adaptive combiners: 1) ATC with adaptive combiners proposed by Takahashi *et al.* [25], and 2) a more recent approach by Tu *et al.* [4], [30]. We also include a baseline network where the nodes do not combine their estimates. The free parameters of all algorithms are chosen to maximize the steady-state performance while keeping a similar convergence rate and, for reproducibility, are shown in Table III. Fig. 8 shows the results for all the mentioned schemes. Although we are more interested in the heterogeneous case, Fig. 8(a) shows also a comparison for a similar homogeneous network where all the step sizes $\tilde{\mu}_k$ are 0.1 to show the suitability of adaptive combiners in general.

If we compare the results in Fig. 8(a) and (b), a first conclusion we can extract is that heterogeneous networks achieve a faster initial convergence and after the change in the optimal solution in the middle of the experiment (iteration $2.5 \cdot 10^4$). For the homogeneous networks, it is interesting to notice that D-ATC with LS adaptive combiners can obtain an additional gain in steady state with respect to all other schemes, illustrating the suitability of the network MSE as the optimization criterion to update the combiners.

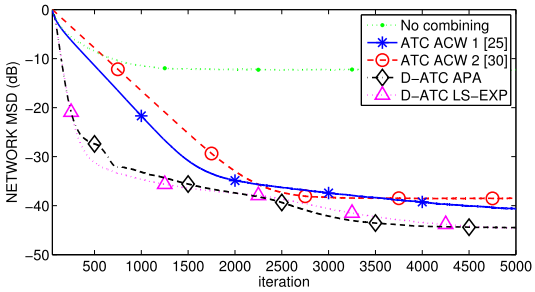
Focusing now on the heterogeneous case, in Fig. 8(b), we can see that D-ATC with both adaptive rules (APA and LS) significantly outperforms standard ATC both in steady state and during the convergence (see the zoom of the first 3000 iterations for a more clear comparison among algorithms). The combination of our adaptive rules and the decoupled scheme seems to be more effective in this heterogeneous setup. Note that adaptive rules for learning the combination weights for standard ATC [25], [30], are derived for homogeneous networks, when only the noise variance changes among the nodes. That explains most of the gap between both approaches.

C. Tracking Performance of D-ATC With Adaptive Combiners

We compare in Fig. 9 the performance of D-ATC and ATC, both with adaptive combiners, when tracking a time-varying solution for two different values of $\mathrm{Tr}\{\mathbf{Q}\}$. The parameters of these simulations are shown in Table III. Analyzing the results, we can conclude that D-ATC outperforms both ATC techniques



(a) Tracking a fast system, $\mathrm{Tr}\{\mathbf{Q}\} = 10^{-4}$.



(b) Tracking a slow system, $\mathrm{Tr}\{\mathbf{Q}\} = 10^{-8}$.

Fig. 9. Network MSD performance for a tracking problem: (a) fast variations, $\mathrm{Tr}\{\mathbf{Q}\} = 10^{-4}$, (b) slow variations $\mathrm{Tr}\{\mathbf{Q}\} = 10^{-8}$.

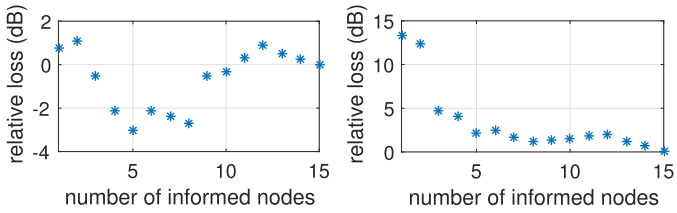
in terms of convergence and steady state, both for the fast and slow time-varying systems.

In conclusion, in all the presented experiments, the proposed D-ATC diffusion scheme outperforms standard ATC, when both schemes use adaptive rules to learn their combiners.

D. Networks With Uninformed Nodes

In this section, we consider a completely different kind of heterogeneous networks. We follow [24] that implements a network with both informed and uninformed nodes (i.e., with and without access to local measurements). In [24], it was shown that ATC networks with fixed combiners do not necessarily benefit from an increased number of informed nodes. In this section we show that our proposed scheme with adaptive combiners is able to use additional data more efficiently, so that an increment in the available information does not degrade network performance.

We consider again the network in Fig. 4 when we increase the number of informed nodes (nodes that receive data and perform the adaptation step) from 1 to 15 nodes. In Fig. 10(a) we show the steady-state NMSD for the standard ATC algorithm with uniform combiners. In such figure, we observe the counterintuitive result of [24]: An increment on the number of informed agents in a network can deteriorate its overall performance. However, when varying the number of informed nodes in a D-ATC network with adaptive LS combiners, an increment on the number of informed agents leads to improved performance, since the combiners are able to modify their values to exploit the new available information. This result further justifies the need of using adaptive combiners in heterogeneous networks.



(a) ATC with uniform combiners (b) D-ATC LS-EXP

Fig. 10. Relative loss in terms of Network MSD performance for a stationary estimation problem in a network with uninformed nodes for ATC with uniform combiners (left) and D-ATC with LS combiners (right). Relative loss is measured with respect to the NMSD of a fully informed network for either case (reference NMSD levels are -23.5 dB for the ATC network with static combiners, and -41.6 dB for the D-ATC LS scheme).

TABLE IV
TOTAL COMPUTATIONAL COST FOR THE ALGORITHMS AND CONFIGURATIONS EMPLOYED IN THE EXPERIMENTS

Operation	Sums	Mult.	Div.	Comp.
ATC ACW 1 [25]	22765	19878	30	81
ATC ACW 2 [30]	13011	9873	30	0
D-ATC LS	25478	25684	30	81
D-ATC APA ($L = 10$)	11717	11553	30	81
D-ATC APA ($L = 500$)	76397	76233	30	81

E. Computational Cost

Finally, we have calculated the computational cost incurred by all the algorithms and configurations evaluated experimentally in this section, as a function of the number of products, sums, divisions and comparisons per iteration.

As it can be seen in Table IV, the performance gains achieved by our proposal have an associated increment in the computational burden with respect to that of the ATC with adaptive combiners in [25] and [30]. This increment results more important for the case of the APA rule with large projection order L than for the case of both the LS algorithm and the APA rule with small L, where the computational cost is on the same order of magnitude than for [25] and [30]. This computational cost could be a limitation in densely connected networks.

VII. CONCLUSIONS AND FUTURE WORK

Heterogeneous diffusion networks offer some additional flexibility with respect to homogeneous networks in which all nodes implement the same update rule using common parameters. In this paper, we have presented a novel diffusion scheme that is especially fitted to heterogeneous networks. Each node of our decoupled ATC (D-ATC) scheme keeps a purely local estimate of the solution vector, and calculates an improved combined estimation using its local estimation and combined estimates received from other nodes in the network. We have shown that, if equipped with appropriate schemes for adapting the network combiners, the proposed diffusion scheme can outperform existing ATC networks (both with fixed and adaptive combiners), requiring only a slight increment in the computational cost.

This work opens a number of research lines worth exploring. From our point of view, one of the most important is the analysis

of asynchronous adaptation in networks. The decoupled nature of D-ATC strategy would make it a good option in such a case. Finally, it is also necessary to evaluate these schemes in the resolution of real tasks. We expect that this contribution helps to further develop the applicability of these networks.

APPENDIX A
AFFINE PROJECTION ALGORITHM DERIVATION

Consider the cost function defined in (55) and repeated here for convenience

$$\text{MSE}_k(n) = \mathbb{E}\left\{[e_k(n) - \bar{\mathbf{c}}_k^T(n)\tilde{\mathbf{y}}_k(n)]^2\right\}. \tag{62}$$

Applying the regularized Newton's method [35] to minimize (62), we obtain

$$\bar{\mathbf{c}}_k(n) = \bar{\mathbf{c}}_k(n-1) + \mu_c[\epsilon\mathbf{I}_{\bar{N}_k} + \mathbf{R}_{\tilde{\mathbf{y}}_k}]^{-1} \times [\mathbf{R}_{e_k,\tilde{\mathbf{y}}_k} - \mathbf{R}_{\tilde{\mathbf{y}}_k}\bar{\mathbf{c}}_k(n-1)] \tag{63}$$

where $\mathbf{R}_{\tilde{\mathbf{y}}_k}$ is the autocorrelation matrix of vector $\tilde{\mathbf{y}}_k(n)$, and $\mathbf{R}_{e_k,\tilde{\mathbf{y}}_k}$ is the cross-correlation vector between $\tilde{\mathbf{y}}_k(n)$ and $e_k(n)$.

Replacing $\mathbf{R}_{\tilde{\mathbf{y}}_k}$ and $\mathbf{R}_{e_k,\tilde{\mathbf{y}}_k}$ by their approximations based on averages over the L most recent values of $\tilde{\mathbf{y}}_k(n)$ and $e_k(n)$ [35], we obtain the update equation for $\bar{\mathbf{c}}_k(n)$ described in (56):

$$\bar{\mathbf{c}}_k(n) = \bar{\mathbf{c}}_k(n-1) + \mu_c[\epsilon\mathbf{I}_{\bar{N}_k} + \tilde{\mathbf{Y}}_k^T(n)\tilde{\mathbf{Y}}_k(n)]^{-1}\tilde{\mathbf{Y}}_k^T(n) \times [\mathbf{e}_k(n) - \tilde{\mathbf{Y}}_k(n)\bar{\mathbf{c}}_k(n-1)]. \tag{64}$$

APPENDIX B
LEAST-SQUARES ALGORITHM DERIVATION

We start from the cost function (58), where we rewrite

$$\check{e}_k(n,i) = e_k(i) + \sum_{\ell=1}^{\bar{N}_k} c_{\ell k}(n)\left[y_k(i) - y_{\ell k}(i)\right]. \tag{65}$$

Taking now the derivatives of (58) with respect to each combination weight $c_{mk}(n)$, with $m = 1, 2, \ldots, \bar{N}_k$, we obtain

$$\frac{\partial J_k(n)}{\partial c_{mk}(n)} = 2\sum_{i=1}^{n}\beta(n,i)\check{e}_k(n,i)\left[y_k(i) - y_{mk}(i)\right]. \tag{66}$$

Replacing (65) in (66), setting the result to zero, and after some algebraic manipulations, we obtain

$$\sum_{i=1}^{n}\sum_{\ell=1}^{\bar{N}_k}\beta(n,i)c_{\ell k}(n)\tilde{y}_{\ell k}(i)\tilde{y}_{mk}(i) = \sum_{i=1}^{n}\beta(n,i)e_k(i)\tilde{y}_{mk}(i).$$

This defines for for each node k a system with $\bar{N}_k$ equations, introducing the usual matrix notation, reads

$$\mathbf{P}_k(n)\bar{\mathbf{c}}_k(n) = \mathbf{z}_k(n), \tag{67}$$

where $\mathbf{P}_k(n)$ is a square symmetric matrix of size $\bar{N}_k$ with components

$$[\mathbf{P}_k(n)]_{p,q} = \sum_{i=1}^{n}\beta(n,i)\tilde{y}_{(\bar{b}_k^{(p)},k)}(i)\tilde{y}_{(\bar{b}_k^{(q)},k)}(i), \tag{68}$$

with $p, q = 1, 2, \ldots, \bar{N}_k$. We introduce the index $\bar{b}_k^{(p)}$ which is the index of the p-th neighbor of k. In addition, $\mathbf{z}_k(n)$ is a column vector of length $\bar{N}_k$, whose pth element is given by

$$z_k^{(p)}(n) = \sum_{i=1}^{n}\beta(n,i)e_k(i)\tilde{y}_{(\bar{b}_k^{(p)},k)}(i), \tag{69}$$

for $p = 1, 2, \ldots, \bar{N}_k$. Thus, the solution of the problem is obtained from (67) using Tikhonov method [43], which leads to (60).

REFERENCES

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

[2] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.

[3] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[4] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing: Array and Statistical Signal Processing*, vol. 3, R. Chellapa and S. Theodoridis, Eds. New York, NY, USA: Academic, 2014, ch. 9, pp. 323–456 (see also arXiv:1205.4220, May 2012).

[5] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

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

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

[8] A. Khalili, M. Tinati, A. Rastegarnia, and J. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 974–979, Feb. 2012.

[9] B. H. Fadlallah and J. C. Principe, "Diffusion least-mean squares over adaptive networks with dynamic topologies," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2013, pp. 1–6.

[10] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3448–3460, Jul. 2015.

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

[12] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[13] P. Di Lorenzo, "Diffusion adaptation strategies for distributed estimation over gaussian markov random fields," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5748–5760, Nov. 2014.

[14] S. Xu, R. C. de Lamare, and H. V. Poor, "Distributed compressed estimation based on compressive sensing," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1311–1315, Sep. 2015.

[15] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3487–3548, Jun. 2015.

[16] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[17] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks," *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 3183–3197, Jun. 2013.

[18] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[19] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 811–858, Feb. 2015.

[20] V. Mhatre and C. Rosenberg, "Homogeneous vs heterogeneous clustered sensor networks: A comparative study," in *Proc. IEEE Int. Conf. Commun.*, 2004, vol. 6, pp. 3646–3651.
[21] Y. Moreno, R. Pastor-Satorras, and A. Vespignani, "Epidemic outbreaks in complex heterogeneous networks," *Eur. Phys. J. B, Condens. Matter Complex Syst.*, vol. 26, no. 4, pp. 521–529, 2002.
[22] A. Damnjanovic *et al.*, "A survey on 3gpp heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.
[23] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. M. Zoubir, "Heterogeneous and multitask wireless sensor networksalgorithms, applications, and challenges," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 3, pp. 450–465, Apr. 2017.
[24] S.-Y. Tu and A. H. Sayed, "On the influence of informed agents on learning and adaptation over networks," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1339–1356, Mar. 2013.
[25] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010.
[26] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.
[27] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE Conf. Decision Control Eur. Control Conf.*, Seville, Spain, 2005, pp. 2996–3000.
[28] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65–78, Sep. 2004.
[29] 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, no. 5, pp. 1865–1877, May 2008.
[30] S.-Y. Tu and A. H. Sayed, "Optimal combination rules for adaptation and learning over networks," in *Proc. 4th IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process.*, 2011, pp. 317–320.
[31] C.-K. Yu and A. H. Sayed, "A strategy for adjusting combination weights over adaptive networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, 2013, pp. 4579–4583.
[32] J. Fernandez-Bes, J. Arenas-García, and A. H. Sayed, "Adjustment of combination weights over adaptive diffusion networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Florence, Italy, 2014, pp. 6409–6413.
[33] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, J. Arenas-García, and M. T. M. Silva, "Distributed estimation in diffusion networks using affine least-squares combiners," *Digit. Signal Process.*, vol. 36, pp. 1–14, 2015.
[34] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, M. T. M. Silva, and J. Arenas-García, "A novel scheme for diffusion networks with least-squares adaptive combiners," in *Proc. IEEE Workshop Mach. Learn. Signal Process.*, Santander, Spain, 2012, pp. 1–6.
[35] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.
[36] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.
[37] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, M. T. M. Silva, and J. Arenas-García, "Improved least-squares-based combiners for diffusion networks," in *Proc. Int. Symp. Wireless Commun. Syst.*, Ilmenau, Germany, 2013, pp. 1–5.
[38] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 2012.
[39] C. Samson and V. U. Reddy, "Fixed-point error analysis of the normalized ladder algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 5, pp. 1177–1191, Oct. 1983.
[40] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[41] M. C. Costa and J. C. M. Bermudez, "An improved model for the normalized LMS algorithm with Gaussian inputs and large number of coefficients," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Orlando, FL, USA, 2002, pp. 1385–1388.
[42] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, and J. Arenas-García, "Least-squares adaptation of affine combinations of multiple adaptive filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, Paris, France, 2010, pp. 2976–2979.
[43] G. Wahba, "Practical approximate solutions to linear operator equations when the data are noisy," *SIAM J. Numer. Anal.*, vol. 14, no. 4, pp. 651–667, May 1977.

Jesus Fernandez-Bes received the B.S. (Hons.) degree in telecommunication engineering and the Ph.D. (Hons.) degree from Universidad Carlos III de Madrid, Leganés, Spain, in 2010 and 2015, respectively. He is a Postdoctoral Researcher with the Aragón Institute for Engineering Research (I3A), Aragón, Spain, and also with the Biomedical Research Networking Center in Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), Málaga, Spain. His research has been focused in the development of new adaptive signal processing and machine learning techniques for different applications, including distributed estimation in wireless sensor networks. He is currently focused in the use of these techniques in Heart Electrophysiology Modeling.

Jerónimo Arenas-García received the Ph.D. (Hons.) degree in telecommunication technologies from Universidad Carlos III de Madrid, Leganés, Spain, in 2004. After a postdoctoral stay at the Technical University of Denmark, he returned to Universidad Carlos III, where he is currently a Lecturer of digital signal and information processing. His research interests include statistical learning theory, particularly in adaptive algorithms and advanced machine learning techniques. He has coauthored more than 80 papers on these topics. He has served as a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society (SPS), Associate Editor of IEEE SIGNAL PROCESSING LETTERS, and the President of the IEEE SPS Spain Chapter.

Magno T. M. Silva (M'05) received the B.S., M.S., and Ph.D. degrees, from Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil, in 1999, 2001, and 2015, resoectively, all in electrical engineering. Since August 2006, he has been with the Department of Electronic Systems Engineering, Escola Politécnica, Universidade de São Paulo, where he is currently an Associate Professor. From January to July 2012, he worked as a Postdoctoral Researcher at Universidad Carlos III de Madrid, Leganés, Spain. He currently serves as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS. His research interests include linear and nonlinear adaptive filtering, and machine learning for signal processing.

Luis A. Azpicueta-Ruiz (M'13) received the telecommunication engineering degree from Universidad Politcnica de Madrid, Madrid, Spain, in 2004, and the Ph.D. degree in telecommunication technologies from Universidad Carlos III de Madrid, Leganés, Spain, in 2011. He is a Lecturer of electroacoustics and acoustic engineering in the Department of Signal Theory and Communications, Universidad Carlos III de Madrid. His research interests include the fields of adaptive and distributed signal processing and their applications, mainly in audio and acoustic processing.