Diffusion Bias-Compensated RLS Estimation over Adaptive Networks

Alexander Bertrand*, Marc Moonen*, Ali H. Sayed†

* Dept. of Electrical Engineering (ESAT-SCD)
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium

† Electrical Engineering Dept.
University of California
Los Angeles, CA 90095, USA

E-mail: alexander.bertrand@esat.kuleuven.be
marc.moonen@esat.kuleuven.be
sayed@ee.ucla.edu
Phone: +32 16 321899, Fax: +32 16 321970

Abstract

We study the problem of distributed least-squares estimation over ad-hoc adaptive networks, where the nodes have a common objective to estimate and track a parameter vector. We consider the case where there is stationary additive colored noise on both the regressors and the output response, which results in biased local least-squares estimators. Assuming that the noise covariance can be estimated (or is known a-priori), we first propose a bias-compensated recursive least-squares algorithm (BC-RLS). However, this bias compensation increases the variance or the mean-square deviation (MSD) of the local estimators, and errors in the noise covariance estimates may still result in residual bias. We demonstrate that the MSD

Alexander Bertrand is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO). This research work was conducted during A. Bertrand's visit to the UCLA Adaptive Systems Laboratory as part of a collaboration with the ESAT Laboratory of Katholieke Universiteit Leuven, which was funded by a travel grant from the Research Foundation - Flanders (FWO). The research was supported by K.U.Leuven Research Council CoE EF/05/006 'Optimization in Engineering' (OPTEC) and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, 'Dynamical systems, control and optimization', 2007-2011), and Research Project FWO nr. G.0600.08 ('Signal processing and network design for wireless acoustic sensor networks'). The scientific responsibility is assumed by its authors. The work of A. H. Sayed was supported in part by NSF grants CCF-1011918, CCF-0942936, and ECS-0725441. A conference precursor of this manuscript has been published in [1].

and residual bias can then be significantly reduced by applying diffusion adaptation, i.e. by letting nodes combine their local estimates with those of their neighbors. We derive a necessary and sufficient condition for mean-square stability of the algorithm, under some mild assumptions. Furthermore, we derive closed-form expressions for its steady-state mean and mean-square performance. Simulation results are provided, which agree well with the theoretical results. We also consider some special cases where the mean-square stability improvement of diffusion BC-RLS over BC-RLS can be mathematically verified.

EDICS: ASP-ANAL, SEN-COLB, SEN-DIST

Index Terms

Adaptive networks, wireless sensor networks, distributed estimation, distributed processing, cooperation, diffusion adaptation

I. Introduction

We study the problem of distributed least-squares estimation over ad-hoc adaptive networks, where the nodes collaborate to pursue a common objective, namely, to estimate and track a common deterministic parameter vector. We consider the case where there is stationary additive colored noise on both the regressors and the output response, which results in biased local least-squares estimators. This is for example a common problem in the analysis of auto-regressive (AR) processes[1]. If this bias is significant and undesired, traditional adaptive methods, such as least mean squares (LMS) or recursive least squares (RLS), are not effective.

For the white noise case, many methods have been developed that yield unbiased estimators, some of which require prior knowledge of the noise variance. A popular method is total least-squares (TLS) estimation. Several adaptive TLS algorithms have been proposed, e.g., recursive TLS [2], total least-mean squares (TLMS) [3], and a distributed TLS method for ad-hoc networks [4]. Under the additional assumption that the noise-free regressors are also white, the modified least-mean squares (MLMS) [5] and modified recursive least-squares (MRLS) [6] algorithm have been proposed. Another important class of algorithms are based on the bias compensation principle [7], where the idea is to subtract an estimate of the asymptotic bias from the least-squares estimators [8]–[10].

In this paper, we consider the case where the regressor noise may be colored and correlated with the noise on the output response. We also rely on the bias compensation principle, and we assume that we

[1] A possible task may be to estimate and track the AR coefficients of a speech signal recorded by a network of microphone nodes that are spatially distributed over an environment. The noise in the microphone signals may then introduce a strong bias on the estimated coefficients.

have a good estimate of the noise covariance[2]. A bias-compensated recursive least-squares (BC-RLS) algorithm is then proposed, to solve an exponentially-weighted least-squares estimation problem. The latter allows to track the parameter vector when it changes over time, by putting less weight on older samples in the estimation.

It is a common observation in estimation theory that any attempt to reduce bias usually results in an increased variance[3] or mean-square deviation (MSD) of the estimator. This is also the case with the proposed BC-RLS algorithm. However, recent developments in adaptive filtering have demonstrated that it is possible to significantly reduce the MSD by letting multiple nodes cooperate [11]–[17]. In this paper, we rely on the idea of diffusion adaptation [11]–[14], [17], where nodes combine their local estimates with the estimates of the nodes in their neighborhoods. It is known that diffusion adaptation usually results in a smaller MSD at each node, without increasing bias. Diffusion adaptation has been successfully applied to the LMS algorithm [11]–[13], and to the RLS algorithm [14]. In this paper, we apply diffusion to the BC-RLS algorithm, which we refer to as diffusion BC-RLS (diffBC-RLS). Simulations demonstrate that diffusion indeed reduces the MSD of the algorithm, and furthermore, that it reduces the residual bias resulting from possible errors in the noise covariance estimates.

The main contribution of this paper is the derivation of the diffusion BC-RLS algorithm, as well as the study of the steady-state performance, both for the diffusion BC-RLS and for the undiffused BC-RLS algorithms. Under some assumptions that are common in the adaptive filtering literature, we will derive a necessary and sufficient condition for the mean-square stability of (diff)BC-RLS. For some special cases, it can be mathematically verified that diffusion improves the mean-square stability of the algorithm. This has also been observed in [12] for the case of diffusion LMS, i.e., cooperation has a stabilizing effect. We also derive a closed-form expression for the residual bias and the MSD in (diff)BC-RLS.

The outline of the paper is as follows. In Section II, we formally define the estimation problem, and we introduce the BC-RLS algorithm. We then define the diffusion BC-RLS algorithm in Section III. We analyze the diffBC-RLS algorithm (the undiffused BC-RLS algorithm is a special case) in terms of its mean and mean-square performance in Section IV. In Section V, we consider some special cases where some extra theoretical results can be obtained. Simulation results are presented in Section VI, and conclusions are drawn in Section VII.

[2] For example, in speech analysis, this can be estimated during silent periods in between words and sentences.

[3] In the sequel, we will focus on the mean-square deviation of the estimator instead of its variance, since the former is usually used to assess the mean-square performance of an adaptive filtering algorithm.

Notation

In this paper, we use boldface letters for random quantities and normal font for non-random (deterministic) quantities or samples of random quantities. We use capital letters for matrices and small letters for vectors. The superscript H denotes complex-conjugate transposition. The index i is used to denote time instants, and the index k is used to denote different nodes in a network with N nodes, defining the set of nodes $\mathcal{K}$. We use $E\{\mathbf{x}\}$ to denote the expected value of a random quantity $\mathbf{x}$.

II. Least-squares estimation with bias compensation

A. Problem Statement

Consider an ad-hoc sensor network with N nodes (the set of nodes is denoted by $\mathcal{K}$). The objective for each node is to estimate a common deterministic $M \times 1$ parameter vector w^o. At every time instant i, node k collects a measurement $d_k(i)$ (referred to as the 'output response') that is assumed to be related to the unknown vector w^o by

$$d_k(i) = \overline{u}_{k,i} w^o + v_k(i) \tag{1}$$

where the regressor $\overline{u}_{k,i}$ is a sample of an $1 \times M$ stochastic row vector[4] $\overline{\mathbf{u}}_{k,i}$, and $v_k(i)$ is a sample of a zero-mean stationary noise process $\mathbf{v}_k$ with variance $\sigma^2_{v_k}$. In [12]–[16], it was assumed that node k also has access to the regressors $\{\overline{u}_{k,i}\}$. Here, we assume that node k observes noisy regressors $\{u_{k,i}\}$, given by

$$u_{k,i} = \overline{u}_{k,i} + n_{k,i} \tag{2}$$

with the $1 \times M$ vector $n_{k,i}$ denoting a sample of a zero-mean stationary noise process $\mathbf{n}_k$ with covariance matrix $R_{n_k} = E\{\mathbf{n}_k^H \mathbf{n}_k\}$. We assume that $\mathbf{n}_k$ is uncorrelated with the regressors $\overline{\mathbf{u}}_{k,i}$, and that $\mathbf{n}_k$ and $\mathbf{v}_k$ are correlated[5] , yielding a non-zero covariance vector $r_{n_k v_k} = E\{\mathbf{n}_k^H \mathbf{v}_k\}$.

The local least-squares (LS) estimator of w^o at node k at time instant i, based on the noisy regressors, is the solution of the optimization problem

$$\hat{w}_{k,i} = \underset{w}{\arg\min} \sum_{j=1}^{i} \left(d_k(j) - u_{k,j} w\right)^2 + \delta \|w\|_2^2 \tag{3}$$

[4] We adopt the notation of [18], i.e., the regressors are defined as row vectors, rather than column vectors.

[5] For example, this may be the case for the estimation of the prediction coefficients of an auto-regressive (AR) process where the data is corrupted by additive colored noise, e.g., in the analysis of speech signals.

where δ is a small positive number that serves as a regularization parameter. The solution of (3) is given by

$$\hat{w}_{k,i} = \hat{R}_{u_k,i}^{-1} \hat{r}_{u_k d_k,i} \tag{4}$$

where

$$\hat{R}_{u_k,i} = \frac{1}{i+1} \left(\sum_{j=1}^{i} u_{k,j}^H u_{k,j} + \delta I_M \right) \tag{5}$$

$$\hat{r}_{u_k d_k,i} = \frac{1}{i+1} \sum_{j=1}^{i} u_{k,j}^H d_k(j) \tag{6}$$

and where I_M denotes the $M \times M$ identity matrix. The normalization with $1/(i+1)$ does not have an influence on $\hat{w}_{k,i}$, but it is introduced such that $\hat{R}_{u_k,i}$ can be used as an estimate of R_{u_k}, assuming stationarity and ergodicity. Since we use noisy regressors, the LS estimator is biased. In the case of stationary and ergodic data, it can be verified that

$$w_k^{\mathrm{LS}} = w^o + R_{u_k}^{-1} r_{n_k v_k} - R_{u_k}^{-1} R_{n_k} w^o \tag{7}$$

where $w_k^{\mathrm{LS}} = \lim_{i \to \infty} \hat{w}_{k,i} = R_{u_k}^{-1} r_{u_k d_k}$ with $R_{u_k} = E\{\mathbf{u}_{k,i}^H \mathbf{u}_{k,i}\}$ and $r_{u_k d_k} = E\{\mathbf{u}_{k,i}^H \mathbf{d}_k(i)\}$, for all $i \in \mathbb{N}$, where $\mathbf{u}_{k,i}$ and $\mathbf{d}_k(i)$ are defined as the stochastic processes that generate the samples $u_{k,i}$ and $d_k(i)$ defined in (2) and (1), respectively. It is noted that w_k^{LS} is in fact a minimum mean-square error estimator (MMSE), but we keep the superscript LS to emphasize that it is a limit case of the LS estimate (4). Let $w_k^b = w_k^{\mathrm{LS}} - w^o$, then the bias w_k^b of the MMSE estimator w_k^{LS} is equal to

$$w_k^b = R_{u_k}^{-1} \left(r_{n_k v_k} - R_{n_k} w^o \right) \ . \tag{8}$$

B. Bias-Compensated Least Squares (BC-LS)

Several BC-LS algorithms have been proposed for the white noise case ($R_{n_k} = \sigma_{n_k}^2 I_M$), which are asymptotically unbiased when the number of observations goes to infinity [8]–[10]. All these BC-LS algorithms are based on the bias compensation principle [7], i.e., an estimate $\hat{w}_{k,i}^b$ of the asymptotic bias w_k^b can be subtracted from the LS estimator $\hat{w}_{k,i}$ to obtain the unbiased estimator (generalized here to incorporate colored noise and mutually correlated noise):

$$\theta_{k,i} \triangleq \hat{w}_{k,i} - \hat{w}_{k,i}^b = \hat{w}_{k,i} + \hat{R}_{u_k,i}^{-1} \left(\hat{R}_{n_k} w^o - \hat{r}_{n_k v_k} \right) \tag{9}$$

where $\hat{w}^b_{k,i} = \hat{R}^{-1}_{u_k,i}\left(\hat{r}_{n_k v_k} - \hat{R}_{n_k} w^o\right)$, and where $\hat{r}_{n_k v_k}$ and $\hat{R}_{n_k}$ are estimates of $r_{n_k v_k}$ and R_{n_k}, respectively. It is assumed that good estimates $\hat{r}_{n_k v_k}$ and $\hat{R}_{n_k}$ are available. In the case of white noise, these estimates can be computed blindly during operation of the algorithm [8]–[10].

Since w^o is unknown in (9), it also has to be replaced with an estimate. The common approach is then to use the previous bias-compensated estimate of w^o instead of the exact w^o in (9). We then obtain the recursive bias-compensated algorithm

$$\psi_{k,i} = \hat{w}_{k,i} + \hat{R}^{-1}_{u_k,i}\left(\hat{R}_{n_k}\psi_{k,i-1} - \hat{r}_{n_k v_k}\right) \tag{10}$$

where $\psi_{k,i}$ replaces $\theta_{k,i}$.

C. Bias-Compensated Recursive Least Squares (BC-RLS)

The BC-LS algorithm can be modified so that it fits into an adaptive filtering context, where also exponential weighting can be incorporated (for tracking purposes). The exponentially-weighted LS estimator (at node k) solves the optimization problem

$$\hat{w}_{k,i} = \arg\min_{w} \sum_{j=1}^{i} \lambda^{i-j}\left(d_k(j) - u_{k,j}w\right)^2 + \lambda^i \delta \|w\|_2^2 \tag{11}$$

where $0 \ll \lambda \leq 1$ is a forgetting factor, putting more weight on more recent observations. The solution of this problem is again given by (4), but the estimates $\hat{R}_{u_k,i}$ and $\hat{r}_{u_k d_k,i}$ are now redefined as

$$\hat{R}_{u_k,i} = \sum_{j=1}^{i} \lambda^{i-j} u^H_{k,j} u_{k,j} + \lambda^i \delta I_M \tag{12}$$

$$\hat{r}_{u_k d_k,i} = \sum_{j=1}^{i} \lambda^{i-j} u^H_{k,j} d_k(j)\ . \tag{13}$$

It is noted that the effective window length is equal to $\frac{1}{1-\lambda} = \sum_{j=0}^{\infty} \lambda^j$, and since there is no normalization for the window length, $\hat{R}_{u_k,i}$ and $\hat{r}_{u_k d_k,i}$ can be considered to be estimates of $\frac{1}{1-\lambda}R_{u_k}$ and $\frac{1}{1-\lambda}r_{u_k d_k}$, respectively [18]. From now on, $\hat{w}_{k,i}$ refers to the solution of the exponentially weighted LS problem (11) and not to the solution of the unweighted LS problem (3), and the same holds for $\hat{R}_{u_k,i}$ and $\hat{r}_{u_k d_k,i}$, now defined by (12)-(13). In Section IV, we will show that, under certain assumptions, (11) is an unbiased estimator of the local MMSE solution at node k, i.e., w_k^{LS}.

The solution of (11) is recursively computed by means of the recursive least-squares (RLS) algorithm

[18]:

$$P_{k,i} = \lambda^{-1}\left(P_{k,i-1} - \frac{\lambda^{-1}P_{k,i-1}u_{k,i}^{H}u_{k,i}P_{k,i-1}}{1+\lambda^{-1}u_{k,i}P_{k,i-1}u_{k,i}^{H}}\right) \tag{14}$$

$$\hat{w}_{k,i} = \hat{w}_{k,i-1} + P_{k,i}u_{k,i}^{H}\left(d_k(i) - u_{k,i}\hat{w}_{k,i-1}\right) \tag{15}$$

with $\hat{w}_{k,0} = 0$ and $P_{k,0} = \delta^{-1}I_M$. At every time instant i, the matrix $P_{k,i}$ is equal to $\hat{R}_{u_k,i}^{-1}$ as defined in (12). With this fact, (10) is transformed into the recursion

$$\psi_{k,i} = \hat{w}_{k,i} + \frac{1}{1-\lambda}P_{k,i}\left(\hat{R}_{n_k}\psi_{k,i-1} - \hat{r}_{n_k v_k}\right) \tag{16}$$

where the factor $\frac{1}{1-\lambda}$ scales $\hat{R}_{n_k}$ and $\hat{r}_{n_k v_k}$ to match with the new definition of $\hat{R}_{u_k,i}$ in (12). The bias correction term in (16) can also be motivated by observing the bias of the exponentially-weighted estimator (11), which will be calculated explicitly later on (see expression (40)). We will refer to the above algorithm as bias-compensated RLS (BC-RLS). It is noted that (16) reduces to the BC-LS recursion (10) if $\lambda = 1$ and if the scaling factor $\frac{1}{1-\lambda}$ in (16) is omitted. We do not provide a convergence analysis of BC-RLS here, since it is a special case of the diffusion BC-RLS algorithm described in the sequel, in particular when cooperation is turned off.

III. Diffusion BC-RLS

In a sensor network, each node k has its own node-specific BC-RLS estimator of w^o, denoted by $\psi_{k,i}$. It is often observed in estimation theory that bias removal introduces a larger MSD and vice versa (see, e.g., [19]). This also often holds in the case of BC-RLS, since the bias compensation usually increases the MSD of the estimators in each node due to the addition of the extra term. It is to be expected that the spatial average of all the $\psi_{k,i}$'s provides a better estimate for w^o, with a smaller MSD. This average could in principle be computed in a distributed fashion by iterative consensus averaging algorithms [20]. The main idea of these algorithms is to collect the estimates $\{\psi_{l,i}\}$ from the neighbors of node k at time i and to iterate over them repeatedly by computing a weighted average, i.e.,

1) Initialize $j \leftarrow 0$ and $\psi_{l,i}^{0} = \psi_{l,i}$.
2) Compute a weighted average

$$\psi_{k,i}^{j+1} = \sum_{l\in\mathcal{N}_k} a_{kl}\psi_{l,i}^{j} \tag{17}$$

3) $j \leftarrow j+1$
4) Return to step 2.

Here, $\mathcal{N}_k$ denotes the set of neighboring nodes of node k (node k included), and a_{kl} is the entry in row k and column l of an $N \times N$ combiner matrix[6] A, where A satisfies

$$A\mathbb{1} = \mathbb{1} \tag{18}$$

with $\mathbb{1} = [1 \ldots 1]^H$ and where $a_{kl} = 0$ if $l \notin \mathcal{N}_k$. The matrix A can be any right-stochastic matrix, but with some constraints due to the network topology. After convergence, the result of (17) becomes the actual estimate $\psi_{k,i}$ for node k at time i. Thus, observe that at every time instant i, multiple consensus iterations need to be applied to the data $\{\psi_{l,i}\}$ to approximate their mean and obtain an improved $\psi_{k,i}$.

Applying consensus averaging in the case of BC-RLS would therefore require a 2-step approach involving two time-scales: one over i and another over j, in between successive i's. First, the nodes estimate a local $\psi_{k,i}$ based on (16), after which an average consensus algorithm is started to iteratively compute

$$\psi_{k,i} = \frac{1}{N}\sum_{l=1}^{N} \psi_{l,i} \tag{19}$$

at each node $k \in \mathcal{K}$. This two-step approach is impractical in real-time systems with high sampling rates since the consensus averaging requires multiple iterations over j for every single iteration i, resulting in a large processing delay and a large amount of communication bandwidth and processing power. By applying diffusion strategies instead (see, e.g., [12], [13]), the iterations of the consensus averaging are merged with those of the BC-RLS algorithm, i.e., the consensus averaging is cut off after a single iteration over j. As a result, only one iteration index remains, and the computational complexity and communication bandwidth are significantly reduced while the network is endowed with improved learning and tracking abilities. The following table summarizes the diffusion BC-RLS (diffBC-RLS) algorithm that would result from a diffusion strategy. Observe how the left-hand side of (23) is a new variable $w_{k,i}$, which then enters into the update (22). In contrast, in a consensus implementation (apart from the second time-scale), the variables that appear on both sides of (17) are the same ψ variables. In (22)-(23), a filtering operation is embedded into (22) to map $w_{k,i-1}$ to $\psi_{k,i}$ at each node and all $\psi_{l,i}$ are then combined into $w_{k,i}$ in (23).

[6] This combiner matrix has to satisfy some constraints to let the consensus averaging algorithm converge [20]. However, since the diffusion BC-RLS algorithm, as derived in the sequel, does not require these constraints, we omit them here.

Diffusion BC-RLS algorithm

Start with $w_{k,0} = 0$, $\hat{w}_{k,0} = 0$ and $P_{k,0} = \delta^{-1} I_M$ for each node $k \in \mathcal{K}$. For every time instant $i > 0$, repeat

1) **RLS update**: for every node $k \in \mathcal{K}$, repeat

$$P_{k,i} = \lambda^{-1} \left(P_{k,i-1} - \frac{\lambda^{-1} P_{k,i-1} u_{k,i}^H u_{k,i} P_{k,i-1}}{1 + \lambda^{-1} u_{k,i} P_{k,i-1} u_{k,i}^H} \right) \tag{20}$$

$$\hat{w}_{k,i} = \hat{w}_{k,i-1} + P_{k,i} u_{k,i}^H \left(d_k(i) - u_{k,i} \hat{w}_{k,i-1} \right) \ . \tag{21}$$

2) **Bias correction update**: for every node $k \in \mathcal{K}$, repeat

$$\psi_{k,i} = \hat{w}_{k,i} + \frac{1}{1-\lambda} P_{k,i} \left(\hat{R}_{n_k} w_{k,i-1} - \hat{r}_{n_k v_k} \right) \ . \tag{22}$$

3) **Spatial update**: for every node $k \in \mathcal{K}$, repeat

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{kl} \psi_{l,i} \tag{23}$$

Remark: It is noted that the RLS update (20)-(21) in node k is spatially isolated, i.e., it does not involve cooperation between the nodes. One may be tempted to also apply diffusion to the RLS estimates, based on the diffusion RLS algorithm in [14]. However, applying diffusion on (20)-(21) will change the local bias in each node, i.e., the local bias at node k will not satisfy (8) anymore. Since the bias compensation (22) is based on (8), and only relies on local statistics, it will not match with the actual bias. Therefore, diffusion of the RLS estimates in combination with the bias compensation (22) is only possible when an invariant spatial profile can be assumed, such that the bias (8) is the same in each node.

IV. ANALYSIS

In this section, we analyze the steady-state performance of the diffBC-RLS algorithm described in Section III. First, we provide a closed-form expression for the bias if there are estimation errors in $\hat{R}_{n_k}$ and $\hat{r}_{n_k v_k}$ (under some standard ergodicity assumptions). Second, if $\hat{R}_{n_k} = R_{n_k}$ and $\hat{r}_{n_k v_k} = r_{n_k v_k}$, we show that the diffBC-RLS algorithm is asymptotically unbiased and we provide a closed-form expression for the MSD, i.e.

$$\text{MSD}_k = \lim_{i \to \infty} E\{ \| \tilde{\boldsymbol{w}}_{k,i} \|^2 \} \tag{24}$$

where

$$\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i} \ . \tag{25}$$

Observe that we are now denoting $\boldsymbol{w}_{k,i}$ by a boldface letter to highlight the fact that it is a random quantity whose variance we are interested in evaluating. It is noted that all results of the analysis of diffBC-RLS also apply to the undiffused BC-RLS algorithm (16), by choosing the combiner matrix A equal to the identity matrix.

A. Data Model

The performance analysis of adaptive filters is rather challenging [18], [21], [22], and it is common to adopt some simplifying assumptions to gain insight in the properties of these algorithms. For the analysis of the diffBC-RLS algorithm, we will introduce some assumptions that are similar to what is traditionally used in the adaptive filtering literature. Simulations show that the theoretical results that are obtained under these assumptions match well with the true performance of the algorithm, for forgetting factors λ that are close to unity and for stationary data.

Assumption 1: The regressors $\overline{u}_{k,i}$ and the additive noise components $n_{k,i}$ are both zero-mean and temporally independent. Furthermore, the covariance matrix $R_{\overline{u}_{k,i}} = E\{\overline{\mathbf{u}}_{k,i}^H \overline{\mathbf{u}}_{k,i}\}$ is time-invariant, i.e., $R_{\overline{u}_{k,i}} = R_{\overline{u}_k}$, $\forall\, i \in \mathbb{N}$. We will therefore often omit the index i in the sequel, when referring to random processes.

It is noted that this assumption also implies that the same conditions hold for the noisy regressors $u_{k,i}$, i.e. $R_{u_{k,i}} = R_{u_k}, \ \forall\, i \in \mathbb{N}$. Furthermore, since the stochastic processes $\overline{\mathbf{u}}_k$ and $\mathbf{n}_k$ are assumed to be uncorrelated, we find that

$$R_{u_k} = R_{\overline{u}_k} + R_{n_k} \ . \tag{26}$$

Assumption 2: All data is spatially uncorrelated, i.e., for $k \neq l$: $E\{\mathbf{u}_k^H \mathbf{u}_l\} = 0$, $E\{\overline{\mathbf{u}}_k^H \overline{\mathbf{u}}_l\} = 0$, $E\{\mathbf{n}_k^* \mathbf{n}_l\} = 0$, $E\{\mathbf{v}_k^* \mathbf{v}_l\} = 0$, $E\{\mathbf{v}_k^* \mathbf{n}_l\} = 0$ and $E\{\mathbf{u}_k^H \mathbf{d}_l\} = 0$.

Since we only perform a steady-state analysis of the algorithm, we will consider the steady-state behavior of the matrix $P_{k,i}$. As $i \to \infty$, we find from (12), and the fact that $P_{k,i}^{-1} = \hat{R}_{u_k,i}$, that

$$\lim_{i\to\infty} E\{\boldsymbol{P}_{k,i}^{-1}\} = \frac{1}{1-\lambda} R_{u_k} \triangleq P_k^{-1} \ . \tag{27}$$

The following two assumptions are made to make the analysis of diffBC-RLS tractable, and both of them are common in the analysis of RLS-type algorithms (see for example [18]).

Assumption 3: $\exists\, i_0$ such that for all $i > i_0$, $P_{k,i}$ and $P_{k,i}^{-1}$ can be replaced with their expected values, i.e. $\exists\, i_0$, such that for all $i > i_0$:

$$P_{k,i} \approx E\{\boldsymbol{P}_{k,i}\} \tag{28}$$

$$P_{k,i}^{-1} \approx E\{\boldsymbol{P}_{k,i}^{-1}\}\ . \tag{29}$$

Assumption 3 implies that the time average of the observed regressors (denoted by $P_{k,i}^{-1}$) can be replaced with the expected value of the stochastic variable that generates these observations. This is a common ergodicity assumption in the analysis of the performance of RLS-type algorithms (see, e.g., [14], [18]), and often yields good results in practice.

Assumption 4: $\exists\, i_0$ such that for all $i > i_0$:

$$E\{\boldsymbol{P}_{k,i}\} \approx E\{\boldsymbol{P}_{k,i}^{-1}\}^{-1} = P_k = (1-\lambda)R_{u_k}^{-1}\ . \tag{30}$$

The last assumption is a coarse approximation, since the expected values $E\{\boldsymbol{P}_{k,i}^{-1}\}$ and $E\{\boldsymbol{P}_{k,i}\}$ do not necessarily share the same inverse relation as their arguments. However, for λ close to unity and a not too large condition number for R_{u_k}, this is a good approximation [14], [18]. However, even in cases where this approximation is not very good, the formulas that are derived in the analysis are still useful to analyze the influence of different parameters, i.e., they usually reflect the correct trends when parameters are varied.

Remark I: Assumption 3 removes some temporal variations in the algorithm, which usually results in an underestimate of the MSD. Assumption 4 increases this effect even more. This can be intuitively explained as follows. Assume that we can approximate the stochastic matrix $\boldsymbol{P}_{k,i}$ with the model $\boldsymbol{P}_{k,i} = Q_{k,i}\boldsymbol{\Lambda}_{k,i}Q_{k,i}^H$, where $\boldsymbol{\Lambda}_{k,i}$ is a stochastic diagonal matrix, and $Q_{k,i}$ a deterministic unitary matrix. In this case $E\{\boldsymbol{P}_{k,i}\} = Q_{k,i}E\{\boldsymbol{\Lambda}_{k,i}\}Q_{k,i}^H$. By using Jensen's inequality, we know that for any random positive diagonal matrix $\boldsymbol{\Sigma}$

$$E\{\boldsymbol{\Sigma}^{-1}\} \geq E\{\boldsymbol{\Sigma}\}^{-1} \tag{31}$$

(this is an elementwise inequality). By substituting $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}_{k,i}^{-1}$, we find that

$$E\{\boldsymbol{\Lambda}_{k,i}\} \geq E\{\boldsymbol{\Lambda}_{k,i}^{-1}\}^{-1}\ . \tag{32}$$

As a consequence, the norm of $E\{\boldsymbol{P}_{k,i}\}$ will be larger than the norm of $E\{\boldsymbol{P}_{k,i}^{-1}\}^{-1}$. Hence, when using approximation (30), we replace $E\{\boldsymbol{P}_{k,i}\}$ with a matrix that has a smaller norm. This will usually results in an underestimate of the MSD, as will be further explained at the end of Subsection IV-C.

Remark II: For notational convenience, we will replace the approximate equality signs '$\approx$' in (28)-(30) with strict equality signs '$=$' in the sequel.

B. Mean Performance

In this subsection, we analyze the steady-state mean performance of the diffBC-RLS algorithm, i.e., we derive a closed-form expression for $E\{\tilde{w}_{k,i}\}$ when i goes to infinity. In this analysis, we incorporate possible estimation errors on the noise covariances, i.e.,

$$\hat{R}_{n_k} = R_{n_k} + \Delta R_{n_k} \tag{33}$$

$$\hat{r}_{n_k v_k} = r_{n_k v_k} + \Delta r_{n_k v_k} \ . \tag{34}$$

We will first derive an expression for the asymptotic bias of the RLS estimator $\hat{w}_{k,i}$. Similar to (25), we define $\check{w}_{k,i} = w^o - \hat{w}_{k,i}$. With (21), we readily find that

$$\check{w}_{k,i} = \check{w}_{k,i-1} - P_{k,i} u_{k,i}^H \left(d_k(i) - u_{k,i} \hat{w}_{k,i-1} \right) \ . \tag{35}$$

Substituting (1) and (2) into (35), we obtain

$$\check{w}_{k,i} = \check{w}_{k,i-1} - P_{k,i} \overline{u}_{k,i}^H \overline{u}_{k,i} w^o - P_{k,i} n_{k,i}^H \overline{u}_{k,i} w^o - P_{k,i} u_{k,i}^H v_k(i) + P_{k,i} u_{k,i}^H u_{k,i} \hat{w}_{k,i-1} \ . \tag{36}$$

Taking the expectation of both sides, and using (26), (28)-(30), we find that for sufficiently large i

$$E\{\boldsymbol{\check{w}}_{k,i}\} = E\{\boldsymbol{\check{w}}_{k,i-1}\} - P_k (R_{u_k} - R_{n_k}) w^o - P_k r_{n_k v_k} + (1-\lambda) E\{\boldsymbol{\hat{w}}_{k,i-1}\} \ . \tag{37}$$

Again using (30), we obtain

$$E\{\boldsymbol{\check{w}}_{k,i}\} = \lambda E\{\boldsymbol{\check{w}}_{k,i-1}\} + P_k (R_{n_k} w^o - r_{n_k v_k}) \ . \tag{38}$$

Expanding the recursion in (38), we find that

$$E\{\boldsymbol{\check{w}}_{k,i}\} = \lambda^{i-i_0} E\{\boldsymbol{\check{w}}_{k,i_0}\} + \sum_{j=i_0}^{i-1} \lambda^{j-i_0} P_k (R_{n_k} w^o - r_{n_k v_k}) \tag{39}$$

where i_0 is chosen such that Assumptions 3 and 4 remain valid. Letting i go to infinity, we obtain

$$\lim_{i \to \infty} E\{\boldsymbol{\check{w}}_{k,i}\} = \frac{1}{1-\lambda} P_k (R_{n_k} w^o - r_{n_k v_k}) \ . \tag{40}$$

Not surprisingly, we find that the asymptotic bias of the exponentially weighted RLS algorithm is equal to the asymptotic bias (8) of the unweighted least-squares estimator.

Let us now introduce some notation that is required to describe the diffusion process of diffBC-RLS, based on stacked variables from all nodes. Let

$$\begin{array}{lcll} w_i & = & \text{col}\{w_{1,i}, \ldots, w_{N,i}\} & (MN \times 1) \\ \hat{w}_i & = & \text{col}\{\hat{w}_{1,i}, \ldots, \hat{w}_{N,i}\} & (MN \times 1) \\ r_{nv} & = & \text{col}\{r_{n_1 v_1}, \ldots, r_{n_N v_N}\} & (MN \times 1) \\ w^o & = & \mathbb{1} \otimes w^o & (MN \times 1) \\ \mathcal{A} & = & A \otimes I_M & (MN \times MN) \\ \mathcal{P}_i & = & \text{blockdiag}\{P_{1,i}, \ldots, P_{N,i}\} & (MN \times MN) \\ \mathcal{R}_n & = & \text{blockdiag}\{R_{n_1}, \ldots, R_{n_N}\} & (MN \times MN) \\ \mathcal{R}_u & = & \text{blockdiag}\{R_{u_1}, \ldots, R_{u_N}\} & (MN \times MN) \end{array}$$

where col$\{.\}$ denotes a stacked column vector, $\otimes$ denotes a Kronecker product and blockdiag$\{.\}$ denotes a block-diagonal matrix. All the derived quantities (such as $\tilde{w}_i$, $\hat{\mathcal{R}}_n$, etc.) have a similar notation for the stacked case, but are omitted for conciseness. Using this notation, and by combining (22) and (23), the recursion of the diffusion RLS algorithm can now be written as

$$w_i = \mathcal{A}\left(\hat{w}_i + \frac{1}{1-\lambda}\mathcal{P}_i(\hat{\mathcal{R}}_n w_{i-1} - \hat{r}_{nv})\right) \ . \tag{41}$$

Subtracting (41) from w^o, and using the fact that $w^o = \mathcal{A}w^o$, yields

$$\tilde{w}_i = \mathcal{A}\left(\check{w}_i - \frac{1}{1-\lambda}\mathcal{P}_i(\hat{\mathcal{R}}_n w_{i-1} - \hat{r}_{nv})\right) \ . \tag{42}$$

Taking the expectation of both sides, and using (33), (34) and (40), we obtain (for $i > i_0$):

$$\begin{aligned} E\{\tilde{\boldsymbol{w}}_i\} &= \frac{1}{1-\lambda}\mathcal{A}\mathcal{P}\mathcal{R}_n E\{\tilde{\boldsymbol{w}}_{i-1}\} - \frac{1}{1-\lambda}\mathcal{A}\mathcal{P}\left(\Delta\mathcal{R}_n E\{\boldsymbol{w}_{i-1}\} - \Delta r_{nv}\right), \\ &= \frac{1}{1-\lambda}\mathcal{A}\mathcal{P}\mathcal{R}_n E\{\tilde{\boldsymbol{w}}_{i-1}\} - \frac{1}{1-\lambda}\mathcal{A}\mathcal{P}\left(\Delta\mathcal{R}_n E\{\boldsymbol{w}_{i-1}\} - \Delta r_{nv} + \Delta\mathcal{R}_n w^o - \Delta\mathcal{R}_n w^o\right), \\ &= \frac{1}{1-\lambda}\mathcal{A}\mathcal{P}\hat{\mathcal{R}}_n E\{\tilde{\boldsymbol{w}}_{i-1}\} - \frac{1}{1-\lambda}\mathcal{A}\mathcal{P}\left(\Delta\mathcal{R}_n w^o - \Delta r_{nv}\right) \ . \end{aligned} \tag{43}$$

Notice that, in the last step, we incorporate the term with $\Delta\mathcal{R}_n$ into the first term, such that $\mathcal{R}_n$ is transformed into $\hat{\mathcal{R}}_n$. Expanding the recursion (43), and using $\mathcal{P} = (1-\lambda)\mathcal{R}_u^{-1}$ (Assumption 4), we find that

$$E\{\tilde{\boldsymbol{w}}_i\} = \left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right)^{i-i_0} E\{\tilde{\boldsymbol{w}}_{i_0}\} - \left(\sum_{j=i_0}^{i-1}\left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right)^{j-i_0}\right)\mathcal{A}\mathcal{R}_u^{-1}\left(\Delta\mathcal{R}_n w^o - \Delta r_{nv}\right) \tag{44}$$

where i_0 is chosen such that Assumptions 3 and 4 remain valid. From this equation, it is observed that stability in the mean[7] of the diffBC-RLS algorithm is obtained if and only if

$$\boxed{\rho\left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) < 1} \tag{45}$$

where $\rho(X)$ denotes the spectral radius of the matrix X, i.e., the magnitude of the eigenvalue of X with largest absolute value. Indeed, if this spectral radius is strictly smaller than 1, the first term vanishes when $i \to \infty$ and the summation in the second term converges. The latter follows from the Taylor expansion of a matrix $(I_M - X)^{-1}$ for any $M \times M$ matrix X satisfying $\rho(X) < 1$, which is given by

$$(I_M - X)^{-1} = \sum_{j=0}^{\infty} X^j \ . \tag{46}$$

Therefore, if (45) holds, it follows that the asymptotic bias of the diffBC-RLS estimators is equal to

$$\boxed{\lim_{i\to\infty} E\{\tilde{\boldsymbol{w}}_i\} = \left(I_{MN} - \mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right)^{-1} \mathcal{A}\mathcal{R}_u^{-1}\left(\Delta r_{nv} - \Delta\mathcal{R}_n w^o\right).} \tag{47}$$

A first important observation is that the estimator is asymptotically unbiased if $\Delta\mathcal{R}_n = 0$ and $\Delta r_{nv} = 0$, i.e., if there is perfect knowledge of the noise covariance. The smaller the error in $\hat{\mathcal{R}}_n$ and $\hat{r}_{nv}$, the smaller the resulting bias.

Note that setting $\mathcal{A} = I_{MN}$ yields the bias of the undiffused BC-RLS estimators (16). It is not possible to make general statements whether diffusion ($\mathcal{A} \neq I_{MN}$) will decrease the bias of the estimators, since this depends on the space-time data statistics (represented by $\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n$) and the network topology (represented by $\mathcal{A}$). This also holds for the stability condition (45). However, since $\mathcal{A}$ has a unity spectral radius, it often has a 'non-expanding' effect, and therefore does not worsen the stability (i.e. the spectral radius (45) does not increase). For some particular cases, it can be mathematically verified that the stability indeed increases, and we refer to Section V for some examples. If the stability increases, this often yields a smaller bias. To see this, observe that $\rho\left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) \leq \rho\left(\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right)$ implies that

$$\rho\left(\left(I_{MN} - \mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right)^{-1}\right) \leq \rho\left(\left(I_{MN} - \mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right)^{-1}\right) \ . \tag{48}$$

This implies that a mapping based on the lefthand side of (48) is 'more contractive' or 'less expanding' than the mapping on the righthand side (corresponding to the undiffused case). Therefore, the bias given in (47) with $\mathcal{A} \neq I_{MN}$ is often (but not necessarily) smaller than with $\mathcal{A} = I_{MN}$. Note that, if diffusion

[7] In Subsection IV-C, we will show that condition (45) for stability in the mean also implies mean-square stability of the diffBC-RLS algorithm.

is applied, there is an additional effect, namely an averaging operator $\mathcal{A}$ applied to the error vector $\mathcal{R}_u^{-1}(\Delta r_{nv} - \Delta\mathcal{R}_n w^o)$. If the combiner matrix A is symmetric, this is a non-expanding mapping, i.e. $\|\mathcal{A}x\| \leq \|x\|$ for all x.

Remark: It is noted that one has to be careful when using the stability condition (45), as it is derived based on Assumption 4. For small values of λ, this assumption is not satisfied, and the algorithm may become unstable, even if (45) holds. Decreasing λ is observed to make the algorithm less stable, since the true matrix $E\{\boldsymbol{P}_{k,i}\}$ has a larger norm than its replacement $R_{u_k}^{-1}$ due to Jensens inequality (see Remark I in subsection IV-A).

C. Mean-Square Performance

In this subsection, we analyze the steady-state mean-square performance[8] of the diffBC-RLS algorithm, i.e., we derive a closed-form expression for $\text{MSD}_k = \lim_{i\to\infty} E\{\|\tilde{\boldsymbol{w}}_{k,i}\|^2\}$. To make the analysis tractable, we assume that $\Delta\mathcal{R}_n = 0$ and $\Delta r_{nv} = 0$.

Let

$$w^{\text{LS}} \;=\; \text{col}\{w_1^{\text{LS}}, \ldots, w_N^{\text{LS}}\} \quad (MN \times 1) \tag{49}$$

where w_k^{LS} is the MMSE estimator in node k, defined in (7). From (7), we find that

$$w^o = \mathcal{A}w^o = \mathcal{A}\left(w^{\text{LS}} - \mathcal{R}_u^{-1} r_{nv} + \mathcal{R}_u^{-1}\mathcal{R}_n w^o\right) \, . \tag{50}$$

Subtracting w_i from both sides in (50), and substituting the diffBC-RLS recursion (22)-(23), we obtain for $i > i_0$ (with $\mathcal{P}_i = \mathcal{P} = (1-\lambda)\mathcal{R}_u^{-1}$ (Assumption 4)):

$$\tilde{w}_i = \mathcal{A} m_i + \mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n \tilde{w}_{i-1} \tag{51}$$

where

$$m_i \triangleq w^{\text{LS}} - \hat{w}_i \, . \tag{52}$$

By expanding the recursion (51), we find that

$$\tilde{w}_i = \sum_{j=i_0}^{i} \left(\mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n\right)^{i-j} \mathcal{A} m_j + \left(\mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n\right)^{i-i_0} \tilde{w}_{i_0} \tag{53}$$

[8] In the mean-square analysis of adaptive filters, one is usually also interested in the so-called excess mean-square error (EMSE) defined by $E\{\|\boldsymbol{u}_{k,i}\tilde{\boldsymbol{w}}_{k,i-1}\|^2\}$. However, since the goal of BC-RLS is to obtain an unbiased estimator for w^o, and not to minimize the EMSE, we do not consider the latter.

where i_0 is chosen such that Assumptions 3 and 4 remain valid. If the stability condition (45) is satisfied, the second term in (53) vanishes when $i \to \infty$, so we will omit it in the sequel. For the sake of an easy exposition, we will set $i_0 = 0$, which does not affect the righthand side of (53) for $i \to \infty$, and if (45) holds. Using the notation $\|x\|_\Sigma^2 = x^H \Sigma x$, we find the following expression for the MSD of node k (in steady-state):

$$\text{MSD}_k = \lim_{i\to\infty} E\{\|\tilde{\boldsymbol{w}}_i\|^2_{\mathcal{E}_k}\} = \lim_{i\to\infty} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} E\{\boldsymbol{m}_{i-m}^H \mathcal{B}_k^{mn} \boldsymbol{m}_{i-n}\} \tag{54}$$

where

$$\mathcal{B}_k^{mn} = \mathcal{A}^H \left(\mathcal{R}_n \mathcal{R}_u^{-1} \mathcal{A}^H\right)^m \mathcal{E}_k \left(\mathcal{A} \mathcal{R}_u^{-1} \mathcal{R}_n\right)^n \mathcal{A} \tag{55}$$

and where $\mathcal{E}_k = E_k \otimes I_M$ with E_k denoting an $N \times N$ matrix with zero-valued entries, except for a one on the k-th diagonal entry. The matrix $\mathcal{E}_k$ serves as a selector matrix to select the part of $\tilde{w}_i$ corresponding to the k-th node.

Expression (54) can be rewritten with a trace operator Tr(.):

$$E\{\|\tilde{\boldsymbol{w}}_i\|^2_{\mathcal{E}_k}\} = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \text{Tr}\left(\mathcal{B}_k^{mn} E\{\boldsymbol{m}_{i-m} \boldsymbol{m}_{i-n}^H\}\right) . \tag{56}$$

In Appendix A, the following expression is derived (for large enough i):

$$E\{\boldsymbol{m}_{i-m} \boldsymbol{m}_{i-n}^H\} = \lambda^{|m-n|} E\{\boldsymbol{m}_i \boldsymbol{m}_i^H\} . \tag{57}$$

With this result, we can rewrite (56) as

$$E\{\|\tilde{\boldsymbol{w}}_i\|^2_{\mathcal{E}_k}\} = \text{Tr}\left(\mathcal{M}_k E\{\boldsymbol{m}_i \boldsymbol{m}_i^H\}\right) . \tag{58}$$

where

$$\mathcal{M}_k = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \lambda^{|m-n|} \mathcal{B}_k^{mn} . \tag{59}$$

In Appendix B the following approximation for $\lim_{i\to\infty} E\{\boldsymbol{m}_i \boldsymbol{m}_i^H\}$ is derived, based on a result from [23]:

$$\lim_{i\to\infty} E\{\boldsymbol{m}_i \boldsymbol{m}_i^H\} \approx \frac{1-\lambda}{2} \mathcal{V} \mathcal{R}_u^{-1} \tag{60}$$

where

$$\mathcal{V} = \text{diag}\{\sigma_1^2, \ldots, \sigma_N^2\} \otimes I_M \tag{61}$$

and with

$$\sigma_k^2 = w^{o\,H} b_k - r_{n_k v_k}^H w^o + \sigma_{v_k}^2 - b_k^H R_{u_k}^{-1} b_k \tag{62}$$

where

$$b_k = R_{n_k} w^o - r_{n_k v_k} \ . \tag{63}$$

It is possible to derive a closed form expression for $\mathcal{M}_k$ defined in (59), based on the eigenvalue decomposition $\mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n = Q\Sigma Q^{-1}$ where Σ is a diagonal matrix with the eigenvalues as its diagonal elements, and where Q contains the corresponding normalized eigenvectors in its columns. We also define the MN-dimensional vector η containing the diagonal elements of Σ^H (the conjugated eigenvalues) in the same order as they appear on the diagonal. In Appendix C, the following closed form expression is derived:

$$\mathcal{M}_k = \mathcal{A}^H \left(\mathcal{M}_{k,2} + \mathcal{M}_{k,2}^H - \mathcal{M}_{k,1} \right) \mathcal{A} \tag{64}$$

with

$$\mathcal{M}_{k,1} = Q^{-H} \left(\frac{Q^H \mathcal{E}_k Q}{\mathbb{1}\mathbb{1}^H - \eta\eta^H} \right) Q^{-1} \tag{65}$$

$$\mathcal{M}_{k,2} = Q^{-H} \left(I_{MN} - \lambda \Sigma^H \right)^{-1} \left(\frac{Q^H \mathcal{E}_k Q}{\mathbb{1}\mathbb{1}^H - \eta\eta^H} \right) Q^{-1} \tag{66}$$

where the double-lined fraction denotes an elementwise division of the matrices in the numerator and denominator (i.e. a Hadamard quotient).

We thus find a closed-form expression for the MSD at node k:

$$\boxed{\mathrm{MSD}_k = \tfrac{1-\lambda}{2} \mathrm{Tr}\left(\mathcal{M}_k \mathcal{V} \mathcal{R}_u^{-1} \right) \ .} \tag{67}$$

It is noted that only the matrix $\mathcal{M}_k$ depends on the combiner matrix A, since it is incorporated in the eigenvalue decomposition of $\mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n$. Note that $\mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n$ is the same matrix that appears in the stability condition (45). Note also that, if the stability condition (45) holds, the denominators in (65) and (66) cannot become zero and $\left(I_{MN} - \lambda \Sigma^H \right)$ cannot become singular, i.e., the algorithm is stable in the mean-square sense.

Again, it is impossible to make general statements about the impact of diffusion on the MSD at a certain node. However, from the Hadamard quotient in (65)-(66), one can expect that the norm of $\mathcal{M}_k$ will be smaller if the norm of η is small. In many cases, setting the matrix $\mathcal{A} \neq I_{MN}$ will decrease the norm of η (although this is not true in general), and then diffusion indeed has a beneficial influence on the MSD. This means that, if diffusion increases the stability (i.e., the spectral radius of $\mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n$ decreases), it often also improves the mean-square performance. In section V, we will consider some special cases where it can indeed be mathematically verified that diffusion decreases the infinity norm of η.

Remark: From (67), it can be seen that the norm of $\mathcal{R}_u^{-1}$ has an influence on the MSD. Intuitively, if $\mathcal{R}_u^{-1}$ has a smaller norm, this will often result in a smaller MSD. This is not only because $\mathcal{R}_u^{-1}$ explicitely appears in the trace, but also because it has an influence through the matrix $\mathcal{M}_k$ in the same manner (a smaller norm of $\mathcal{R}_u^{-1}$ usually results in a smaller norm of η and therefore larger denominators in (65)-(66)). As explained in Remark I of Subsection IV-A, the norm of the matrix $(1-\lambda)\mathcal{R}_u^{-1} = \mathcal{P}$ is smaller than the norm of $E\{\boldsymbol{\mathcal{P}}\}$. However, Assumption 4 replaces $E\{\boldsymbol{\mathcal{P}}\}$ with $\mathcal{P}$. This in effect, together with the removal of some variability due to Assumption 3, usually results in an underestimate of the MSD.

V. Special Cases

In this section, we consider some special cases where the diffBC-RLS algorithm is guaranteed to be stable, or where it can be mathematically verified that diffusion improves stability of the BC-RLS algorithm, i.e., (compare with (45))

$$\rho\left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) \leq \rho\left(\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) \ . \tag{68}$$

As mentioned earlier, if (68) holds, diffusion often (but not necessarily) also decreases the bias and the MSD of the estimators. It is noted that diffusion in general provides better results (with respect to stability, bias and MSD) due to the non-expanding effect of the combiner matrix A. The beneficial influence of diffusion is therefore not limited to the special cases given below. These merely serve as “motivating” examples where the beneficial influence of diffusion can be theoretically verified.

Remark: Unless stated otherwise, we assume that the combiner matrix A is symmetric, which is required in most cases to make some conclusions. The Metropolis rule (see, e.g., [20]) offers a procedure to select the weights, based on the network topology, that yields a symmetric combiner matrix A.

A. Invariant Spatial Profile

If the regressor and noise covariance matrices are the same in each node, i.e., $R_{n_k} = R_n$, $R_{u_k} = R_u$, and if each node uses the same estimate $\hat{R}_{n_k} = \hat{R}_n$, for $k \in \{1, \ldots, N\}$, we find that

$$\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n = A \otimes R_u^{-1}\hat{R}_n \ . \tag{69}$$

Since the set of eigenvalues of $X \otimes Y$ is equal to the set of all pairwise products between the eigenvalues of X and the eigenvalues of Y, we find that

$$\rho\left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) = \rho\left(R_u^{-1}\hat{R}_n\right) \tag{70}$$

i.e., the algorithm is stable if and only if the undiffused BC-RLS at a single node is stable. In this case, diffusion has no effect on stability[9]. However, since the eigenvalues of $\mathcal{A}$ are inside the unit circle, many eigenvalues of $\mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n = A \otimes R_u^{-1}R_n$ will be strictly smaller than the corresponding eigenvalues of $R_u^{-1}R_n$ (and none of the eigenvalues can increase). This means that the norm of η in (65) will be smaller than in the undiffused case ($A = I_N$), which mostly results in a smaller MSD. The same holds for the asymptotic bias given in (47), since smaller eigenvalues of $\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n$ yield a more contractive or a less expanding mapping $\left(I_{MN} - \mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right)^{-1}$. It is noted that the combiner matrix A does not need to be symmetric to obtain the above results.

B. 2-norm Constraint ($\|\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 < 1$)

If $\|R_{u_k}^{-1}\hat{R}_{n_k}\|_2 < 1$, for $k \in \{1, \ldots, N\}$, where $\|.\|_2$ denotes the matrix 2-norm, the block-diagonal structure of $\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n$ implies that

$$\|\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 < 1 \ . \tag{71}$$

Although the condition (71) does not imply (68), it is an interesting case since stability of the diffBC-RLS algorithm is guaranteed when (71) holds. Indeed, we have that

$$\rho\left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) \leq \|\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 \leq \|\mathcal{A}\|_2\|\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 = \|\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 < 1 \ . \tag{72}$$

The first inequality follows from the fact that the spectral radius is the infimum of all induced norms of a matrix (including the two-norm), and the second inequality follows from the fact that the two-norm is sub-multiplicative. Since the two-norm and the spectral radius are the same for symmetric matrices (we assume a symmetric $\mathcal{A}$), we have that $\|\mathcal{A}\|_2 = \rho(\mathcal{A}) = 1$.

It is noted that $\|\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 < 1$ is satisfied if either the noise $n_{k,i}$ or the regressors $\overline{u}_{k,i}$ are white at each node, and if $\hat{\mathcal{R}}_n = \mathcal{R}_n$ (see subsections V-C and V-D).

[9] This is not surprising, since the invariant spatial profile assumption implies that either all nodes are stable, or none of them are. In the latter case, diffusion adaptation cannot help.

C. White Noise on Regressors

Assume that we have prior knowledge that $R_{n_k} = \sigma^2_{n_k} I_M$ and $\hat{R}_{n_k} = \hat{\sigma}^2_{n_k} I_M$, for $k \in \{1, \ldots, N\}$. Let $\overline{U}_k \overline{\Lambda}_k \overline{U}_k^H$ denote the eigenvalue decomposition of the clean regressor covariance matrix $R_{\overline{u}_k}$, then we obtain

$$R^{-1}_{u_k} \hat{R}_{n_k} = \overline{U}_k \text{diag} \left\{ \frac{\hat{\sigma}^2_{n_k}}{\overline{\lambda}_{k,1} + \sigma^2_{n_k}}, \ldots, \frac{\hat{\sigma}^2_{n_k}}{\overline{\lambda}_{k,M} + \sigma^2_{n_k}} \right\} \overline{U}_k^H \tag{73}$$

where $\overline{\lambda}_{k,1} \geq \overline{\lambda}_{k,2} \geq \ldots \geq \overline{\lambda}_{k,M}$ are the diagonal elements of $\overline{\Lambda}_k$.

Note that $\mathcal{R}_u^{-1} \hat{\mathcal{R}}_n$ is symmetric in this case, and therefore

$$\rho\left(\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) \leq \|\mathcal{A}\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 \leq \|\mathcal{A}\|_2 \|\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\|_2 = \rho\left(\mathcal{A}\right)\rho\left(\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) = \rho\left(\mathcal{R}_u^{-1}\hat{\mathcal{R}}_n\right) \tag{74}$$

i.e., (68) holds. Furthermore, $\|R^{-1}_{u_k}\hat{R}_{n_k}\|_2 = \frac{\hat{\sigma}^2_{n_k}}{\overline{\lambda}_{k,1} + \sigma^2_{n_k}}$. If $\hat{\sigma}^2_{n_k} < \overline{\lambda}_{k,1} + \sigma^2_{n_k}$ for each k, i.e., if the noise variances are not significantly overestimated, we know from subsection V-B that the diffBC-RLS algorithm is stable.

D. White Regressors

Assume that we have prior knowledge that $R_{\overline{u}_k} = \sigma^2_{\overline{u}_k} I_M$, for $k \in \{1, \ldots, N\}$. Furthermore, assume that $\hat{\mathcal{R}}_n = \mathcal{R}_n$, i.e., a good estimate of the noise covariance is available. We thus have that

$$R^{-1}_{u_k} \hat{R}_{n_k} = \left(\sigma^2_{\overline{u}_k} I_M + R_{n_k}\right)^{-1} R_{n_k} \ . \tag{75}$$

Since $(\alpha I + X)^{-1} X = X (\alpha I + X)^{-1}$ for every X and α, it follows that $R^{-1}_{u_k} \hat{R}_{n_k}$ is a symmetric matrix. Therefore, with a similar reasoning as in subsection V-C, we again obtain (68). From (75), it is also obvious that $\|R^{-1}_{u_k}\hat{R}_{n_k}\|_2 < 1$, and therefore we know from subsection V-B that the diffBC-RLS algorithm is stable.

VI. SIMULATION RESULTS

In this section, we provide simulation results to compare the performance of the BC-RLS and diffBC-RLS algorithm, and we compare the simulation results with the theoretical results of Section IV.

The measurements $d_k(i)$ were generated according to (1), and the clean regressors $\overline{u}_{k,i}$ were chosen Gaussian i.i.d. with a covariance matrix $R_{\overline{u}_k} = Q_1 \text{diag}\{5, 4, 3, 2, 1\} Q_1^H$, where Q_1 is a random unitary matrix. The stacked vectors of the regressor noises and the measurement noises $\overline{n}_{k,i} = [n_{k,i}\, v_k(i)]$ were also chosen Gaussian i.i.d. with a random covariance matrix $E\{\overline{\mathbf{n}}^H_{k,i}\overline{\mathbf{n}}_{k,i}\} = s_k Q_2 \text{diag}\{2, 1.8, 1.6, 1.4, 1.2, 1\} Q_2^H$, where Q_2 is again a random unitary matrix, and s_k is a random scalar drawn from a uniform distribution

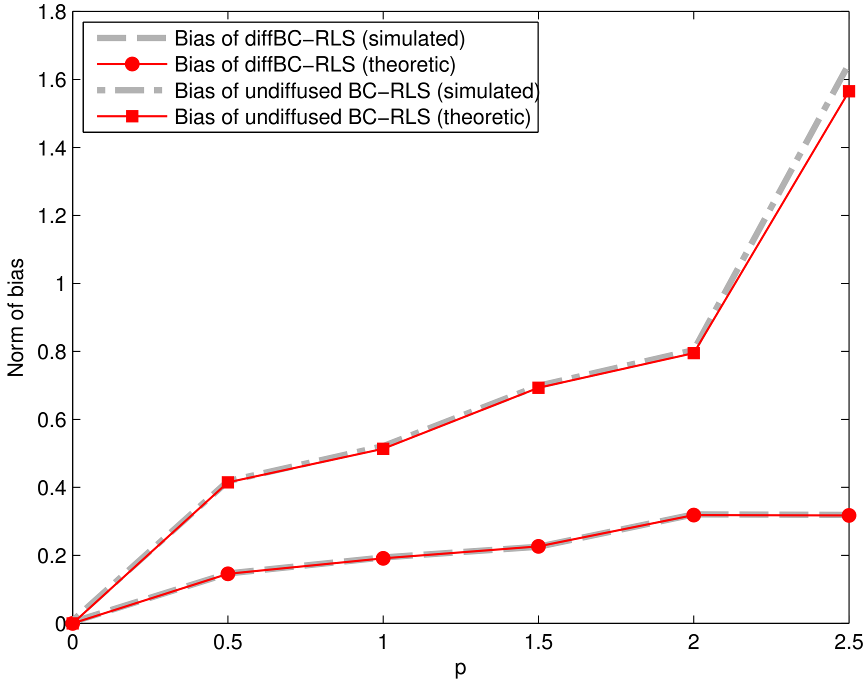

Fig. 1. The norm of the stacked asymptotic bias as a function of p, using $\lambda = 0.999$.

in the interval $[0.1, 1]$. Note that, due to the scaling with s_k, this is not an invariant spatial profile, since there is a different SNR in each node. The network had $N = 20$ nodes, and the topology was chosen randomly with a connectivity of 5 links per node on average. The size of the unknown vector w^o was $M = 5$, and the combiner matrix A was constructed using Metropolis weights. All results are averaged over 200 experiments.

A. *Bias*

In this subsection, we add some errors to the noise estimates $\hat{R}_{n_k} = R_{n_k} + \Delta R_{n_k}$ and $\hat{r}_{n_k v_k} = r_{n_k v_k} + \Delta r_{n_k v_k}$ to investigate the effect on the bias of the BC-RLS and diffBC-RLS estimators. The errors were modelled as

$$\Delta R_{n_k} = \sqrt{p|R_{n_k}|} \odot R_k \tag{76}$$

$$\Delta r_{n_k v_k} = \sqrt{p|r_{n_k v_k}|} \odot r_k \tag{77}$$

where $\odot$ denotes a Hadamard product (elementwise multiplication), the operator $|.|$ denotes an elementwise absolute value operator, and p is a positive scalar variable that is used to increase the error. The

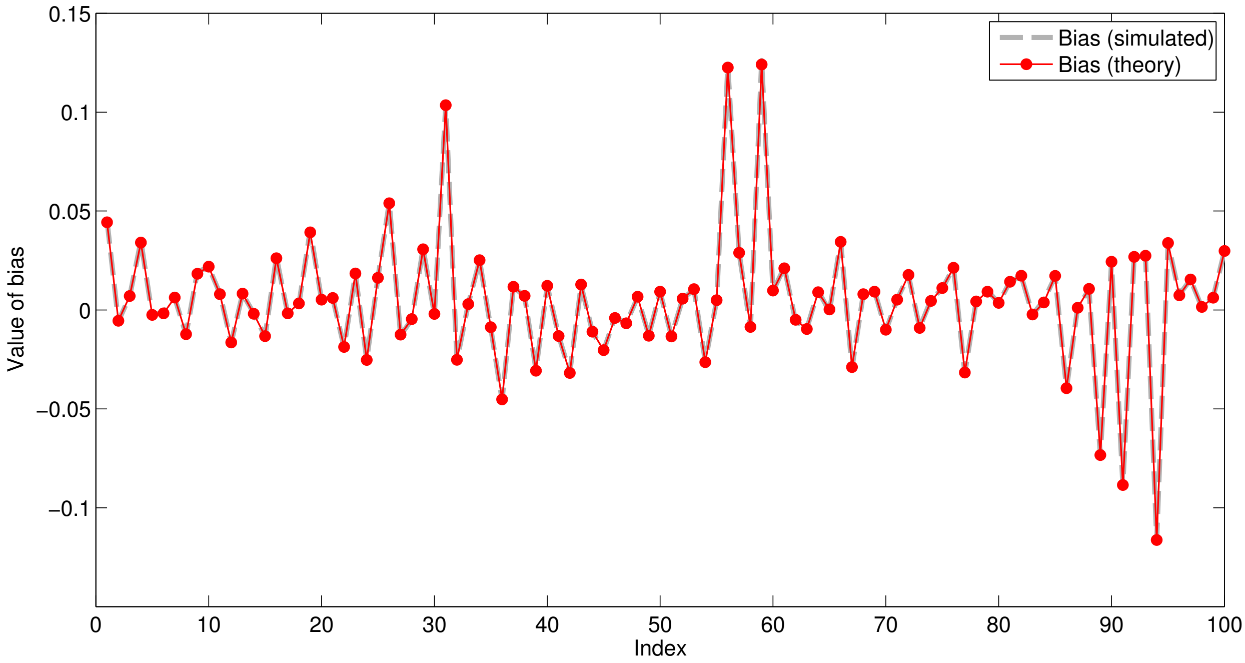

Fig. 2. The entries of $\lim_{i\to\infty} E\{\tilde{\boldsymbol{w}}_i\}$ in the diffBC-RLS algorithm (steady state) for $p = 0.2$, using $\lambda = 0.999$.

entries of the $M \times M$ matrix R_k and the M-dimensional vector were independently drawn from a normal distribution (i.e. with zero mean and unity variance).

Fig. 1 shows $\lim_{i\to\infty} \|E\{\tilde{\boldsymbol{w}}_i\}\|$, i.e. the norm of the stacked asymptotic bias, as a function of p, both for BC-RLS (without cooperation) and diffBC-RLS in steady state (with $\lambda = 0.999$). We see that the theoretical results (47) match very well with the simulated results. Furthermore, we observe that diffusion indeed significantly decreases the asymptotic bias of the BC-RLS estimators. Fig. 2 shows the entries of the stacked bias vector $\lim_{i\to\infty} E\{\tilde{\boldsymbol{w}}_i\}$ resulting from the diffBC-RLS algorithm (with $p = 0.2$), again demonstrating that the theoretical results are very accurate.

B. MSD

In Fig. 3, we show the MSD in the different nodes, for several values of λ. We see that the theoretical results (67) match very well with the simulated results, especially when λ is close to unity. However, when λ is too small (e.g. $\lambda = 0.9$), the algorithm becomes unstable in some iterations. The reason for this is the fact that the approximation (30) in Assumption 4 becomes invalid. As mentioned at the end of subsection IV-B, the algorithm may become unstable at some iterations due to Jensen's inequality, even though the

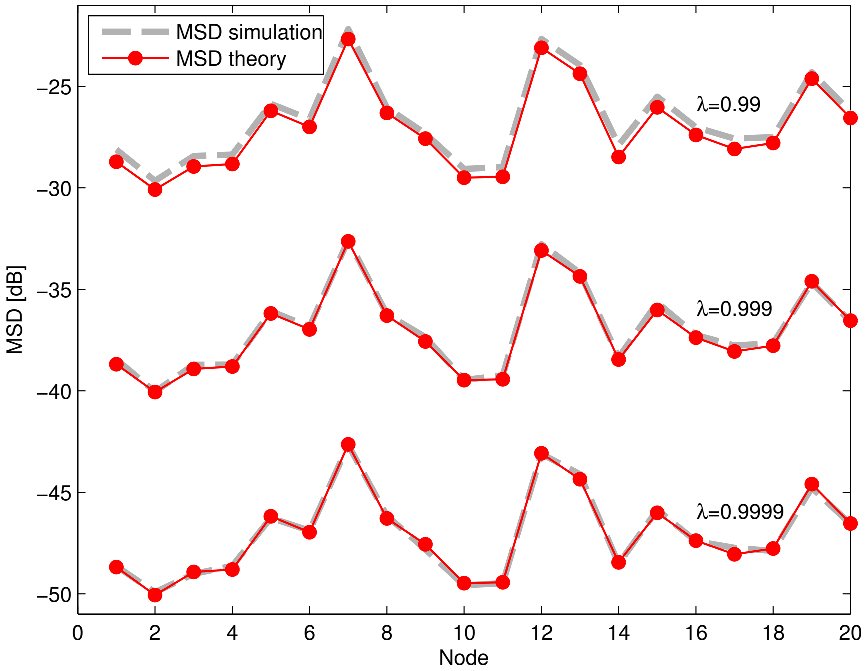

Fig. 3. The steady-state MSD values in each node, for different values of λ.

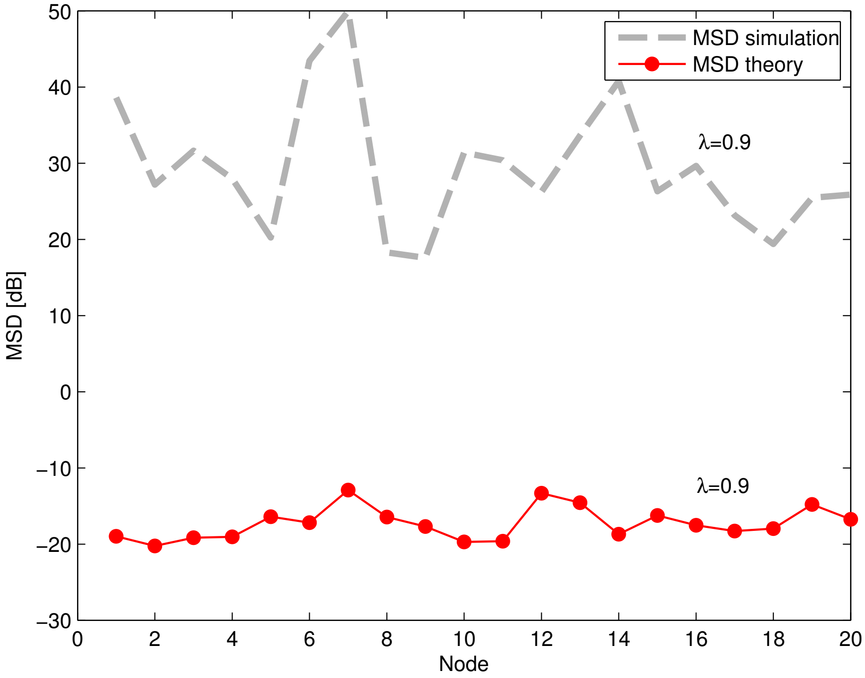

Fig. 4. The average MSD values in each node for $\lambda = 0.9$.

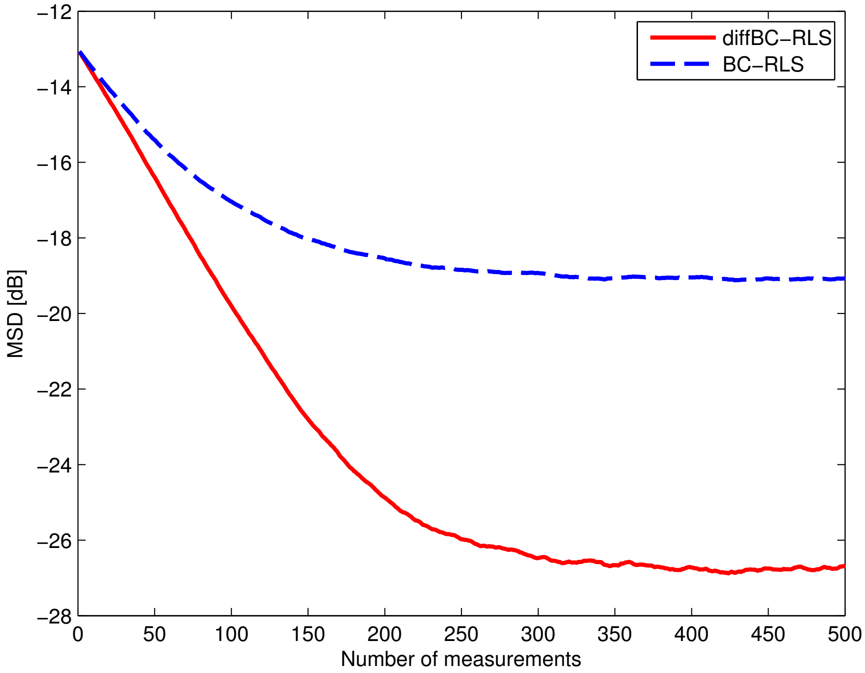

Fig. 5. MSD curves of the BC-RLS algorithm and the diffBC-RLS algorithm with $\lambda = 0.99$.

stability condition (45) is satisfied. This is demonstrated[10] in Fig. 4. Since the theoretical analysis does not incorporate this effect, there is no match between the theoretical results and the simulation results for this case.

In Fig. 5, the MSD is plotted as a function of the number of observations[11] both for BC-RLS (without cooperation) and diffBC-RLS. It is observed that the MSD is significantly smaller when the nodes diffuse their estimations.

VII. Conclusions

We have addressed the problem of distributed least-squares estimation over adaptive networks when there is stationary additive colored noise on both the regressors and the output response, which results in a bias on the least-squares estimators. Assuming that the noise covariance can be estimated (or is known a-priori), we have proposed a bias-compensated recursive least-squares (BC-RLS) algorithm. This bias compensation significantly increases the MSD of the local estimators, and errors in the noise covariance

[10] There is no steady-state in this case. The plotted MSD is the time average of the last 3000 iterations of the algorithm.

[11] Since the norm of $P_{k,i}$ can be very large in the beginning due to a small regularization parameter δ, the recursion usually starts to diverge until i becomes large enough. Therefore, to obtain an intelligible figure, we initialized the matrices $P_{k,0}$ with $(1-\lambda)R_{u_k}^{-1}$, i.e., the convergence of the RLS part is removed.

estimates may still result in a significant residual bias. By applying diffusion, i.e., letting neighboring nodes combine their local estimates, the MSD and residual bias can be significantly reduced. The latter is referred to as diffusion BC-RLS (diffBC-RLS). We have derived a necessary and sufficient condition for mean-square stability of the algorithm, under some mild assumptions. Furthermore, we have derived closed-form expressions for the residual bias and the MSD, which match well with the simulation results if the forgetting factor is close to unity. We have also considered some special cases where the stability improvement of diffBC-RLS over BC-RLS can be mathematically verified.

A possible application of the diffBC-RLS algorithm is the AR analysis of a speech signal in a wireless sensor network, with microphone nodes that are spatially distributed over an environment. RLS has been demonstrated to be able to track speech AR parameters [24] in environments with limited noise. For noisy recordings, bias compensation is crucial for AR analysis of speech signals. However, this bias compensation usually severely increases the MSD of the estimated speech AR coefficients (often resulting in unstable behavior), due to the ill-conditioned nature of the speech covariance matrix in certain speech phonemes. Since all the microphones observe the same speech signal (possibly at a different SNR), the stability of the algorithm and the MSD of the estimators can be greatly improved by applying diffusion adaptation.

VIII. Acknowledgements

The first author would like to thank all the co-workers in the Adaptive Systems Laboratory at UCLA for the fruitful discussions regarding several aspects of this manuscript, and the anonymous reviewers for their valuable suggestions to improve the manuscript.

Appendix

A. Derivation of expression (57)

We first consider the case where $m < n$. Because of the steady-state assumption, only the difference $t = n - m$ is important, i.e. $E\{\boldsymbol{m}_{i-m}\boldsymbol{m}_{i-n}^H\} = E\{\boldsymbol{m}_i\boldsymbol{m}_{i-t}^H\}$. Because of the spatial independence assumption and the fact that $E\{\boldsymbol{m}_i\} = 0$ (this follows from (40)), $E\{\boldsymbol{m}_i\boldsymbol{m}_{i-t}^H\}$ will be a block-diagonal matrix (note that there is no diffusion on the local RLS estimates). Therefore, we can focus on a single block corresponding to node k, i.e. the submatrix $E\{\boldsymbol{m}_{k,i}\boldsymbol{m}_{k,i-t}^H\}$ where $\boldsymbol{m}_{k,i} \triangleq w_k^{\mathrm{LS}} - \hat{\boldsymbol{w}}_{k,i}$.

From the RLS update (21), we find that (for large enough i and $t > 0$)

$$\begin{aligned} E\{\boldsymbol{m}_{k,i}\boldsymbol{m}_{k,i-t}^H\} &= E\{\boldsymbol{m}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} + P_{k,i}R_{u_k}E\{\hat{\boldsymbol{w}}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} \\ &= E\{\boldsymbol{m}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} + (1-\lambda)E\{\hat{\boldsymbol{w}}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} \\ &= E\{\boldsymbol{m}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} + (\lambda-1)E\{-\hat{\boldsymbol{w}}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} + (\lambda-1)E\{w_k^{\text{LS}}\boldsymbol{m}_{k,i-t}^H\} \\ &= E\{\boldsymbol{m}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} + (\lambda-1)E\{\boldsymbol{m}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} \\ &= \lambda E\{\boldsymbol{m}_{k,i-1}\boldsymbol{m}_{k,i-t}^H\} \end{aligned} \tag{78}$$

where we used Assumption 4 in the second step, and the fact that $E\{\boldsymbol{m}_{k,i}\} = 0$ (when $i \to \infty$) in the first and third step. By expanding the recursion (78), and because of the steady-state assumption, we find that

$$E\{\boldsymbol{m}_{k,i}\boldsymbol{m}_{k,i-t}^H\} = \lambda^t E\{\boldsymbol{m}_{k,i}\boldsymbol{m}_{k,i}^H\} \tag{79}$$

The case where $m > n$ or $t < 0$ immediately follows from (79) by using a substitution $j = i - t$:

$$\begin{aligned} E\{\boldsymbol{m}_{k,i}\boldsymbol{m}_{k,i-t}^H\} &= E\{\boldsymbol{m}_{k,j+t}\boldsymbol{m}_{k,j}^H\} \\ &= E\{\boldsymbol{m}_{k,j}\boldsymbol{m}_{k,j+t}^H\}^H \\ &= \lambda^{-t}E\{\boldsymbol{m}_{k,j}\boldsymbol{m}_{k,j}^H\} \\ &= \lambda^{-t}E\{\boldsymbol{m}_{k,i}\boldsymbol{m}_{k,i}^H\}\ . \end{aligned} \tag{80}$$

Both cases (together with the trivial case $m = n$) can be handled simultaneously by using an absolute value in the exponent of λ, which straightforwardly results in (57).

B. *Derivation of expression (60)-(63)*

Because of the spatial independence assumption, and the fact that $E\{\boldsymbol{m}_i\} = 0$ (this follows from (40)), $E\{\boldsymbol{m}_i\boldsymbol{m}_i^H\}$ will be a block-diagonal matrix (note that there is no diffusion on the local RLS estimates). Therefore, we can focus on a single block corresponding to node k, i.e. the submatrix $E\{\boldsymbol{m}_{k,i}\boldsymbol{m}_{k,i}^H\}$ where $\boldsymbol{m}_{k,i} \triangleq w_k^{\text{LS}} - \hat{\boldsymbol{w}}_{k,i}$.

For an RLS algorithm that observes regressors u_i and corresponding $d(i) = u_i\phi^o + e(i)$ where $e(i)$ is zero-mean noise with variance σ_e^2, the following approximation holds for $i \to \infty$ [23]:

$$E\{(\phi^o - \boldsymbol{w}_i)(\phi^o - \boldsymbol{w}_i)^H\} \approx \frac{1-\lambda}{2}\sigma_e^2 E\{\boldsymbol{u}_i^H\boldsymbol{u}_i\}\ . \tag{81}$$

It is noted that the variable $\hat{w}_{k,i}$ in the diffBC-RLS algorithm is an unbiased estimator of w_k^{LS}. Therefore, we can view $\hat{w}_{k,i}$ as the outcome of an RLS algorithm that observes regressor $u_{k,i} = \overline{u}_{k,i} + n_{k,i}$ and a corresponding

$$d_k(i) = u_{k,i} w_k^{\mathrm{LS}} + e_k(i) \; . \tag{82}$$

Since the observations $d_k(i)$ are actually equal to

$$d_k(i) = \overline{u}_{k,i} w^o + v_k(i) \tag{83}$$

we have to rewrite it to the form (82). With (7), and setting (82) equal to (83), some straightforward algebra yields

$$e_k(i) = v_k(i) - n_{k,i} w^o + u_{k,i} R_{u_k}^{-1} \left(R_{n_k} w^o - r_{n_k,v_k} \right) \; . \tag{84}$$

Using the approximation (81), we find that

$$E\{\boldsymbol{m}_{k,i} \boldsymbol{m}_{k,i}^H\} \approx \frac{1-\lambda}{2} \sigma_{e_k}^2 R_{u_k} \tag{85}$$

where $\sigma_{e_k}^2 = E\{|\boldsymbol{e}_k(i)|^2\}$. We will now derive an expression for $\sigma_{e_k}^2$. For the sake of an easy exposition, we omit the subscript k and the time index i in the sequel. We introduce the notation $b \triangleq R_n w^o - r_{nv}$. Observe that

$$E\{\left(\boldsymbol{v} - \boldsymbol{n} w^o\right) \boldsymbol{u}^H\} = r_{nv} - R_n w^o = -b \; . \tag{86}$$

With this, we find that

$$E\{|\boldsymbol{e}|^2\} = E\{|\boldsymbol{v} - \boldsymbol{n} w^o|^2\} + E\{\boldsymbol{u} R_u^{-1} b b^H R_u^{-1} \boldsymbol{u}^H\} - 2 b^H R_u^{-1} b \; . \tag{87}$$

The first term is equal to

$$E\{|\boldsymbol{v} - \boldsymbol{n} w^o|^2\} = w^{o\,*} R_{n_k} w^o - r_{n_k v_k}^H w^o - w^{o\,*} r_{n_k v_k} + \sigma_v^2 \; . \tag{88}$$

Using the trace operator, we find that the second term is equal to

$$E\{\boldsymbol{u} R_u^{-1} b b^H R_u^{-1} \boldsymbol{u}^H\} = \mathrm{Tr}\left(R_u^{-1} b b^H R_u^{-1} E\{\boldsymbol{u}^H \boldsymbol{u}\} \right) = b^H R_u^{-1} b \; . \tag{89}$$

Inserting (87)-(89) in (85), and repeating this for each node, we find expression (60)-(63).

C. Derivation of expression (65)-(66)

We introduce the notation

$$\mathcal{D} \triangleq \mathcal{A}\mathcal{R}_u^{-1}\mathcal{R}_n \tag{90}$$

where the eigenvalue decomposition of $\mathcal{D}$ is given by $\mathcal{D} = Q\Sigma Q^{-1}$. The matrix $\mathcal{M}_k$ defined in (59) can then be rewritten as

$$\mathcal{M}_k = \mathcal{A}^H \left(\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \lambda^{|m-n|} (\mathcal{D}^H)^m \mathcal{E}_k \mathcal{D}^n \right) \mathcal{A} \; . \tag{91}$$

First, observe that

$$\begin{aligned} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \lambda^{|m-n|} (\mathcal{D}^H)^m \mathcal{E}_k \mathcal{D}^n = & \sum_{m=0}^{\infty} (\mathcal{D}^H)^m \mathcal{E}_k \mathcal{D}^m \\ & + \sum_{n=1}^{\infty} \lambda^n \sum_{m=n}^{\infty} (\mathcal{D}^H)^m \mathcal{E}_k \mathcal{D}^{m-n} \\ & + \sum_{n=1}^{\infty} \lambda^n \sum_{m=n}^{\infty} (\mathcal{D}^H)^{m-n} \mathcal{E}_k \mathcal{D}^m \; . \end{aligned} \tag{92}$$

We will first focus on the second term of (92), which we denote as T_2. Note that the last term is the conjugate transpose of T_2. Using the eigenvalue decomposition of $\mathcal{D}$, we find that

$$\begin{aligned} T_2 &= Q^{-H} \left(\sum_{n=1}^{\infty} \lambda^n \sum_{m=n}^{\infty} (\Sigma^H)^m Q^H \mathcal{E}_k Q \Sigma^{m-n} \right) Q^{-1} \\ &= Q^{-H} \left(\sum_{n=1}^{\infty} \lambda^n (\Sigma^H)^n \sum_{m=n}^{\infty} (\Sigma^H)^{m-n} Q^H \mathcal{E}_k Q \Sigma^{m-n} \right) Q^{-1} \\ &= Q^{-H} \left(\sum_{n=1}^{\infty} \lambda^n (\Sigma^H)^n \left(\left(Q^H \mathcal{E}_k Q \right) \odot E \right) \right) Q^{-1} \end{aligned} \tag{93}$$

where $\odot$ denotes a Hadamard product (elementwise multiplication), and with E a matrix where the entry on the i-th row and the j-th column is given by

$$E_{ij} = \sum_{m=n}^{\infty} \eta_i^{m-n} (\eta_j^H)^{m-n} \tag{94}$$

where η_i is the i-th diagonal element of Σ^H (the conjugate of the i-th eigenvalue). If the stability condition (45) holds, then $|\eta_i \eta_j^H| < 1$, $\forall\, i, j \in \{1, \ldots, M\}$. Based on a Taylor expansion similar to (46), but for the complex scalar case, we find that

$$E_{ij} = \frac{1}{1 - \eta_i \eta_j^H} \; . \tag{95}$$

Using this, (93) can be rewritten as

$$T_2 = Q^{-H} \left(\sum_{n=1}^{\infty} \lambda^n (\Sigma^H)^n \right) \frac{Q^H \mathcal{E}_k Q}{\mathbb{1}\mathbb{1}^H - \eta\eta^H} Q^{-1} \tag{96}$$

where the MN-dimensional vector $\eta = \text{col}\{\eta_1, \ldots, \eta_{MN}\}$ contains the diagonal elements of Σ^H (the conjugated eigenvalues) in the same order as they appear on the diagonal.

Based on the expansion (46), we can rewrite (96) as

$$T_2 = Q^{-H} \left(\left(I_{MN} - \lambda \Sigma^H \right)^{-1} - I_{MN} \right) \frac{Q^H \mathcal{E}_k Q}{\mathbb{1}\mathbb{1}^H - \eta\eta^H} Q^{-1} \ . \tag{97}$$

A closed-form expression for the first term of (92) can be derived in a similar way, which results in

$$T_1 = Q^{-H} \frac{Q^H \mathcal{E}_k Q}{\mathbb{1}\mathbb{1}^H - \eta\eta^H} Q^{-1} \ . \tag{98}$$

Since $T_1 + T_2 + T_2^H = \mathcal{M}_{k,2} + \mathcal{M}_{k,2}^H - \mathcal{M}_{k,1}$, we eventually find (65)-(66).

References

[1] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion-based bias-compensated RLS for distributed estimation over adaptive sensor networks," in *Proc. of the European signal processing conference (EUSIPCO)*, Barcelona, Spain, Aug. 2011.

[2] C. E. Davila, "An efficient recursive total least squares algorithm for FIR adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 42, pp. 267–280, 1994.

[3] D.-Z. Feng, Z. Bao, and L.-C. Jiao, "Total least mean squares algorithm," *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2122 –2130, Aug. 1998.

[4] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 59, no. 5, pp. 2320–2330, May 2011.

[5] H. So, "Modified LMS algorithm for unbiased impulse response estimation in non-stationary noise," *Electronics Letters*, vol. 35, pp. 791–792, May 1999.

[6] D. Feng, Z. Bao, and X. Zhang, "Modified RLS algorithm for unbiased estimation of fir system with input and output noise," *Electronics Letters*, vol. 36, no. 3, pp. 273 –274, Feb. 2000.

[7] S. Sagara and K. Wada, "On-line modified least-squares parameter estimation of linear discrete dynamic systems," *International Journal of Control*, vol. 25, pp. 329–343, 1977.

[8] W. X. Zheng, "A least-squares based algorithm for fir filtering with noisy data," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2003, pp. IV–444 – IV–447 vol.4.

[9] L. Jia, C. Jin, and K. Wada, "On bias compensated recursive least-squares algorithm for FIR adaptive filtering," in *Proc. of IFAC Workshop on Adaptation and Learning in Control and Signal Processing*, 2001, pp. 347–352.

[10] L. Jia, R. Tao, Y. Wang, and K. Wada, "Forward/backward prediction solution for adaptive noisy FIR filtering," *Science in China: Information Sciences*, vol. 52, pp. 1007–1014, 2009.

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

[12] ——, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[13] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Transactions on Signal Processing*, vol. 58, pp. 1035–1048, March 2010.

[14] F. Cattivelli, C. G. Lopes, and A. H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[15] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Performance analysis of the consensus-based distributed LMS algorithm,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 981030, 19 pages, 2009. doi:10.1155/2009/981030.

[16] ——, “Distributed recursive least-squares for consensus-based in-network adaptive estimation,” *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, 2009.

[17] F. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069 –2084, 2010.

[18] A. H. Sayed, *Adaptive Filters*. NJ: John Wiley & Sons, 2008.

[19] D. W. Marquardt, “Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation,” *Technometrics*, vol. 12, no. 3, pp. pp. 591–612, 1970.

[20] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. International Symposium on Information Processing in Sensor Networks (IPSN)*. Piscataway, NJ, USA: IEEE Press, 2005, pp. 63–70.

[21] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[22] B. Widrow and S. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[23] L. Guo, L. Ljung, and P. Priouret, “Performance analysis of the forgetting factor RLS algorithm,” *International Journal of Adaptive Control and Signal Processing*, vol. 7, pp. 525–537, 1993.

[24] C. Papaodysseus, G. Roussopoulos, and A. Panagopoulos, “Using a fast RLS adaptive algorithm for efficient speech processing,” *Mathematics and Computers in Simulation*, vol. 68, no. 2, pp. 105 – 113, 2005.

Alexander Bertrand (M’08) received the M.Sc. degree (2007) and the Ph.D. degree (2011) in electrical engineering, both from Katholieke Universiteit Leuven, Belgium. He is currently a Postdoctoral Fellow of the Research Foundation - Flanders (FWO), affiliated with the Electrical Engineering Department (ESAT) of Katholieke Universiteit Leuven. In 2010, he was a visiting researcher at the Adaptive Systems Laboratory, University of California, Los Angeles (UCLA). His research interests are in multi-channel signal processing, ad hoc sensor arrays, wireless sensor networks, distributed signal enhancement, speech enhancement, and distributed estimation.

Dr. Bertrand received a Postdoctoral Fellowship from the Research Foundation - Flanders (FWO) (2011-2014), a Ph.D. scholarship of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) (2008-2011), and a travel grant from FWO for a Visiting Research Collaboration at UCLA (2010).

Marc Moonen (M'94, SM'06, F'07) received the electrical engineering degree and the PhD degree in applied sciences from Katholieke Universiteit Leuven, Belgium, in 1986 and 1990 respectively.

Since 2004 he is a Full Professor at the Electrical Engineering Department of Katholieke Universiteit Leuven, where he heads a research team working in the area of numerical algorithms and signal processing for digital communications, wireless communications, DSL and audio signal processing.

Dr. Moonen received the 1994 KU Leuven Research Council Award, the 1997 Alcatel Bell (Belgium) Award (with P. Vandaele), the 2004 Alcatel Bell (Belgium) Award (with R. Cendrillon), and was a 1997 "Laureate of the Belgium Royal Academy of Science". He received a journal best paper award from the *IEEE Transactions on Signal Processing* (with G. Leus) and from *Elsevier Signal Processing* (with S. Doclo).

He was chairman of the IEEE Benelux Signal Processing Chapter (1998-2002), and is currently President of EURASIP (European Association for Signal Processing).

He has served as Editor-in-Chief for the '*EURASIP Journal on Applied Signal Processing*' (2003-2005), and has been a member of the editorial board of '*IEEE Transactions on Circuits and Systems II' (2002-2003)*, '*IEEE Signal Processing Magazine*' (2003-2005) and '*Integration, the VLSI Journal*'. He is currently a member of the editorial board of '*EURASIP Journal on Advances in Signal Processing*', '*EURASIP Journal on Wireless Communications and Networking*', and '*Signal Processing*'.

Ali H. Sayed (F'01) is Professor of Electrical Engineering at the University of California, Los Angeles (UCLA), where he directs the Adaptive Systems Laboratory (*www.ee.ucla.edu/asl*). He has published widely in the areas of adaptation and learning, statistical signal processing, estimation theory, adaptive and bio-inspired networks, and signal processing for communications. He has authored or coauthored several books, including Adaptive Filters (Wiley 2008), Fundamentals of Adaptive Filtering (Wiley 2003), and Linear Estimation (Prentice-Hall, 2000). Dr. Sayed has served on the editorial boards of several journals, including as Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. His work has received several recognitions, including the 1996 IEEE Donald G. Fink Award, 2003 Kuwait Prize, 2005 Terman Award, and 2002 Best Paper Award, and 2005 Young Author Best Paper Award from the IEEE Signal Processing Society. He has served as a 2005 Distinguished Lecturer of the same society and as General Chairman of the 2008 International Conference on Acoustics, Speech and Signal Processing (ICASSP). He is serving as the Vice-President of Publications of the IEEE Signal Processing Society.