This is a repository copy of Robust Distributed Diffusion Recursive Least Squares Algorithms with Side Information for Adaptive Networks.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/141046/

Version: Accepted Version

Article:
Yu, Yi, Zhao, Haiquan, Caiado De Lamare, Rodrigo et al. (2 more authors) (Accepted: 2019) Robust Distributed Diffusion Recursive Least Squares Algorithms with Side Information for Adaptive Networks. IEEE Transactions on Signal Processing. ISSN 1053-587X (In Press)

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.
Robust Distributed Diffusion Recursive Least Squares Algorithms with Side Information for Adaptive Networks

Yi Yu, Haiquan Zhao, Senior Member, IEEE, Rodrigo C. de Lamare, Senior Member, IEEE, Yury Zakharov, Senior Member, IEEE, and Lu Lu

Abstract—This work develops robust diffusion recursive least squares algorithms to mitigate the performance degradation often experienced in networks of agents in the presence of impulsive noise. The first algorithm minimizes an exponentially weighted least-squares cost function subject to a time-dependent constraint on the squared norm of the intermediate update at each node. A recursive strategy for computing the constraint is proposed using side information from the neighboring nodes to further improve the robustness. We also analyze the mean-square convergence behavior of the proposed algorithm. The second proposed algorithm is a modification of the first one based on the dichotomous coordinate-descent approach. It has a performance similar to that of the former, however its complexity is significantly lower especially when input regressors of agents have a shift structure and it is well suited to practical implementation. Simulations show the superiority of the proposed algorithms over previously reported techniques in various impulsive noise scenarios.

Index Terms—Distributed algorithms, diffusion cooperation, dichotomous coordinate-descent, impulsive noises, recursive least squares algorithms.

I. INTRODUCTION

OVER the past decade, distributed parameter estimation over wireless sensor networks with multiple nodes (agents) has attracted much attention. It only relies on the local data exchange between interconnected nodes, and therefore removes the requirement of a powerful central processor and, as such, reduces communications bandwidth of the traditional centralized estimation whilst retaining similar estimation performance [3]. [5]. Distributed estimation has been applied to target localization [4], clustering [5], frequency estimation [4] and spectrum estimation in Cognitive radio (CR) [7] [8].

This work is partially supported by National Nature Science Foundation of P.R. China (Nos: 61871461, 61571374, 61433011, 11472297). The work of Y. Zakharov is partly supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through Grant EP/R003297/1. The work of L. Lu is supported by China Postdoctoral Science Foundation Funded Project under Grant 2018M649016. An earlier version of this work was reported in the conference presentation IEEE International Conf. Acoustics, Speech and Signal Processing (ICASSP), Alberta, Canada, April 2018 [1].

Y. Yu is with the School of Information Engineering, Southwest University of Science and Technology, Mianyang, 621010, China (e-mail: yyyi_xuyuan@163.com).

H. Zhao is with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu, 610031, China. (e-mail: hqzhao_swjtu@126.com).

R. C. de Lamare is with the CETUC, PUC-Rio, Rio de Janeiro 22451-900, Brazil, and also with the Department of Electronic Engineering, University of York, York YO10 5DD, U.K. (e-mail: rcdl500@ohm.york.ac.uk).

Y. Zakharov is with the Department of Electronic Engineering, University of York, York YO10 5DD, U.K. (e-mail: yury.zakharov@york.ac.uk).

L. Lu is with the School of Electronics and Information Engineering, Sichuan University, Chengdu, China. (lulu19900303@126.com).

A. Prior and Related Work

According to the cooperation strategies between interconnected nodes, existing algorithms can be categorized as incremental [9], consensus [10], and diffusion [11]–[14] types. Among these, the diffusion strategy is popular, because it does not require a Hamiltonian cycle path as in the incremental type, thereby it is more robust to nodes/links failures; it is stable and shows a faster convergence rate and a lower mean-square error than that of the consensus approach. Several diffusion algorithms were proposed, e.g., diffusion least mean square (dLMS) algorithm [11] and its variable step size variants [15], [16].

In practice, the measurements can be corrupted by non-Gaussian noise with impulsiveness. Impulsive noise has small occurrence probability but much higher amplitude than the nominal measurements. It may occur due to atmospheric phenomena, or man-made due to either electric machinery in the operation environment [17]–[19]. Other examples are keyboard clicking or pen dropping in teleconference [20], double-talk in echo cancellation [21], biological noise [22] or ice cracking [23] in various underwater signals, out-of-band spectral leakage in CR [24], etc. In such scenarios, the conventional algorithms like the dLMS designed for Gaussian noise would undergo a significant performance deterioration. To this end, many robust distributed algorithms have been proposed. Some algorithms are based on the instantaneous gradient-descent method to minimize different robust criteria, for instance, the diffusion error nonlinearity (dEN) [25], diffusion least mean p-th power (dLMP) [26], diffusion sign error LMS (dSE-LMS) [27], and diffusion maximum correntropy (dMC) algorithms. Moreover, because of the insensitivity of correntropy to impulsive noise, the maximum total correntropy diffusion algorithm was proposed in [29] for the case of large outliers in communication links. Nevertheless, their main limitation is slow convergence especially when the nodes’ input signals are colored (highly correlated). As shown in [25], the dEN algorithm converges slower than the dSE-LMS algorithm. In [30], by resorting to the adaptive projected subgradient method, a robust diffusion algorithm was developed which projects the output errors onto halfspaces defined by Huber’s error function at each node, thereby speeding up the convergence. However, the setting of the parameters controlling the algorithm’s robustness requires prior knowledge of the noise distribution which is often unavailable.
It is well-known that due to the exponentially weighted least squares (EWLS) criterion, the diffusion recursive LS (dRLS) algorithm provides fast convergence even for colored signals [31], [32]. By means of the alternating direction method of multipliers to solve the EWLS problem, Mateos et al. proposed another type of distributed RLS algorithm [33]. Following this algorithm, to reduce computation and communication costs, its variants were presented by censoring observations with small innovations [34]. Likewise, these algorithms might experience convergence issues in impulsive noise environments, because impulsive noise samples are directly involved in the adaptation through output errors of nodes. For the single-agent case, many works have proposed RLS algorithms robust against impulsive noise, e.g., [35], [36]. However, distributed RLS-based techniques that are robust to impulsive noise have not been well investigated. The study in [37] develops the diffusion recursive least $p$-th power (dRLP) algorithm, while its robustness relies on the value of $p$ as the dLMP does.

Analogous to the RLS, distributed RLS requires high computational complexity. Apart from this, it may also suffer from numerical instability due to accumulation of round-off errors in finite-precision implementations [38]. Aiming to address these problems, an efficient alternative method is the dichotomous coordinate-descent (DCD) that solves a system of normal equations associated with the RLS-type algorithms [38]–[41]. In particular, the DCD method only involves shift and addition operations, thus the DCD-based RLS algorithms reduce the computational cost and improve the numerical stability in contrast with the original RLS counterparts, whilst preserving comparable estimation performance. For this reason, reference [42] also explored the use of the DCD in distributed networks, and developed the DCD-dRLS algorithm. It is worth mentioning that, however, the development of the DCD-based algorithms in impulsive noise environments has not been studied in single nor multi-agent scenarios.

### B. Contributions

The focus of this paper is on developing robust distributed RLS algorithms for scenarios with impulsive noise. Specifically, our contributions are listed as follows:

1) A robust dRLS (R-dRLS) algorithm is developed by extending the framework of [33] to multi-agent scenarios with a diffusion distributed strategy. To ensure that the proposed R-dRLS algorithm has good convergence performance after an abrupt change in the set of parameters to be estimated, we also propose a diffusion-based non-stationary control (NC) method.

2) Theoretical insights into the mean square steady-state and evolution behaviors of the R-dRLS algorithm in impulsive noise environment are presented.

3) We employ the DCD method for developing recursions used in the adaptation step of the R-dRLS algorithm, resulting in the DCD-R-dRLS algorithm with similar learning performance. Remarkably, the DCD-R-dRLS algorithm brings a reduction in computational complexity; especially for shift structured input regressors, it reduces the order of complexity from $O(M^2)$ to $O(M)$, where $M$ is the length of the estimated vector.

4) Simulation examples are presented to demonstrate the performance of the proposed algorithms in impulsive noise scenarios described by either Bernoulli-Gaussian (BG) or $\alpha$-stable processes.

In comparison to the preliminary results [1] related to this work, the current version is further developed due to the main contributions 2) and 3). We slightly improve the NC method by a smoothing operation as shown in [15]. Moreover, the effectiveness of the proposed algorithms are also verified in an application to distributed spectrum estimation.

This paper is organized as follows. In Section II, the estimation problem is described. The R-dRLS algorithm is derived in Section III. Analyses of its mean square behavior are presented in Section IV. In Section V, we review the DCD algorithm and propose the DCD-R-dRLS algorithm. In Section VI, extensive simulations are presented to verify the proposed algorithms. Finally, conclusions are given in Section VII.

**Notation:** Throughout the paper, all vectors are column vectors. We use the parenthesis on $i$ to denote matrices and vectors, and the subscript on $i$ to denote scalars. The superscript $(\cdot)^T$ denotes the transpose, $\|\cdot\|_2$ denotes the $l_2$-norm of a vector, and $E\{\cdot\}$ denotes the expectation of random variables. We use $\text{col}\{\cdot\}$ to denote an enlarged column vector structured by stacking its columns on top of each other, $\text{diag}\{\cdots\}$ to yield a diagonal matrix with its arguments, and $\text{Tr}\{\cdot\}$ to denote the trace of a matrix. $I_M$ is the identity matrix of size $M \times M$, $\otimes$ is the Kronecker product, and $\mathbb{I}$ is the column vector of length $M$ with all entries being one. For symmetric matrices $X$ and $Y$, the notation $X \succeq Y$ stands for $X - Y \succeq 0$, meaning that the matrix difference $X - Y$ is positive semi-definite.

### II. Problem Formulation

Let us consider a diffusion network with $N$ nodes located at different positions in space, as shown in Fig. [1] where each node communicates only with its neighboring nodes by a link (single-hop communication). All nodes connected directly to node $k$ (including itself) are referred to as its neighborhood, denoted as $\mathcal{N}_k$. At every time instant $i \geq 0$, every node has access to an $M \times 1$ input regressor vector $u_{k,i}$ and an output measurement $d_k(i)$, which are related as:

$$d_k(i) = u_{k,i}^T w^o + v_k(i),$$

where $w^o$ is a parameter vector of size $M \times 1$ to be estimated, and $v_k(i)$ is the additive noise at node $k$. The additive noises $v_k(i)$ and $v_l(j)$ are spatially and temporally independent for $k \neq l$ and $i \neq j$. Moreover, any $u_{k,i}$ is independent of any $v_l(j)$. The model (1) is used in many applications [3], [43]. The objective of the in-network processing is to estimate $w^o$, using the available data $\{u_{k,i}, d_k(i)\}$ collected at nodes. For this purpose, the global EWLS estimation problem is described as [31]:

$$w^* = \arg \min_{w} \left\{ \lambda^{i+1} \|w\|_2^2 + \sum_{j=0}^{i} \sum_{k=1}^{N} (d_k(j) - u_{k,j}^T w)^2 \right\},$$

**Notation:** Throughout the paper, all vectors are column vectors. We use the parenthesis on $i$ to denote matrices and vectors, and the subscript on $i$ to denote scalars. The superscript $(\cdot)^T$ denotes the transpose, $\|\cdot\|_2$ denotes the $l_2$-norm of a vector, and $E\{\cdot\}$ denotes the expectation of random variables. We use $\text{col}\{\cdot\}$ to denote an enlarged column vector structured by stacking its columns on top of each other, $\text{diag}\{\cdots\}$ to yield a diagonal matrix with its arguments, and $\text{Tr}\{\cdot\}$ to denote the trace of a matrix. $I_M$ is the identity matrix of size $M \times M$, $\otimes$ is the Kronecker product, and $\mathbb{I}$ is the column vector of length $M$ with all entries being one. For symmetric matrices $X$ and $Y$, the notation $X \succeq Y$ stands for $X - Y \succeq 0$, meaning that the matrix difference $X - Y$ is positive semi-definite.
where $\delta > 0$ is a regularization constant, and $\lambda$ $(0 < \lambda \leq 1)$ is the forgetting factor. The dRLS algorithm solves (2) in a diffusion-based distributed manner [31]. As already mentioned in the introduction, the noise $v_k(i)$ may be non-Gaussian with impulsiveness so that the algorithms derived from (2), e.g., the dRLS algorithm, would exhibit poor convergence and even diverge. In general, when studying robust adaptive algorithms, both contaminated-Gaussian (CG) [27], [44] and $\alpha$-stable [45], [46] random processes are often used for modeling impulsive noise.

III. PROPOSED R-dRLS ALGORITHM

In this section, we derive the R-dRLS algorithm and propose a control method for endowing it with tracking capability. The diffusion strategy has two alternatives: the adapt-then-combine (ATC) and the combine-then-adapt (CTA). However, we focus only on the ATC policy, which performs first the adaptation step and then the combination step. This is based on the fact that the extension to CTA is straightforward by reversing the order of the adaptation and combination steps [2], [3]. In what follows, we neglect the notation ATC for brevity.

A. dRLS Algorithm

To conveniently develop the R-dRLS algorithm, we rederive here the dRLS algorithm from the following method instead of directly solving (2).

In the adaptation step, every node $k$, at time instant $i$, finds an intermediate estimate $\psi_k(i)$ of $w^o$ by minimizing the individual local cost function:

$$J_k(\psi_k,i) = \|\psi_k,i - w_{k,i-1}\|^2_{B_{k,i}} + \|d_k(i) - u^T_{k,i}\psi_k,i\|^2,$$

with $B_{k,i} = \Phi_{k,i} - u_{k,i}u^T_{k,i}$, where

$$\Phi_{k,i} \triangleq \lambda^{-i+1}\delta I_M + \sum_{j=0}^{i} \lambda^{i-j} u_{k,j}u^T_{k,j} \quad (4)$$

is the time-averaged correlation matrix for the input vector at node $k$, and $w_{k,i-1}$ is an estimate of $w^o$ at node $k$ at time instant $i-1$. Notice that the quadratic form $\|x\|^2_{B_{k,i}} \triangleq x^TB_{k,i}x$ in (3) defines the Riemannian distance between vectors $\psi_k,i$ and $w_{k,i-1}$, where $B_{k,i}$ is a Riemannian metric tensor characterizing that the distance properties are not uniform along the $M$-dimensional space [47], [48].

Setting the derivative of $J_k(\psi_k,i)$ with respect to $\psi_k,i$ to zero, we obtain

$$\psi_k,i = w_{k,i-1} + P_{k,i}u_{k,i}c_k(i), \quad (5)$$

where

$$c_k(i) = d_k(i) - u^T_{k,i}w_{k,i-1} \quad (6)$$

stands for the output error at node $k$, and

$$P_{k,i} \triangleq \Phi_{k,i}^{-1} = \frac{1}{\lambda}(P_{k,i-1} - \frac{P_{k,i-1}u_{k,i}u^T_{k,i}P_{k,i-1}}{\lambda + u^T_{k,i}P_{k,i-1}u_{k,i}}), \quad (7)$$

with $P_{k,0}$ initialized as $P_{k,0} = \delta^{-1}I_M$. The recursion (7) is the result of applying the matrix inversion lemma [43].

At the combination step, the intermediate estimates $\psi_{m,i}$, $m \in N_k$ from the neighborhood of node $k$ are linearly weighted, yielding a combined estimate $w_{k,i}$:

$$w_{k,i} = \sum_{m \in N_k} c_{m,k}\psi_{m,i}, \quad (8)$$

where the combination coefficients $\{c_{m,k}\}$ are non-negative, and satisfy:

$$\sum_{m \in N_k} c_{m,k} = 1, \quad \text{and} \quad c_{m,k} = 0 \text{ if } m \notin N_k. \quad (9)$$

Note that $c_{m,k}$ is a weight that node $k$ assigns to the intermediate estimate $\psi_{m,i}$ received from its neighbor node $m$. If one assumes $w_{k,i-1} = \Phi_{k,i-1}^{-1}z_{k,i-1}$ in (5), where $z_{k,i} = \lambda z_{k,i-1} + u_{k,i}d_k(i)$, (5) is a standard RLS update for node $k$. In summary, (3)–(8) formulate the dRLS algorithm. It is noteworthy that the term $P_{k,i}u_{k,i}$ in (5) provides the decorrelating ability for colored inputs, thus speeding up the convergence.

**Remark 1**: In general, $\{c_{m,k}\}$ in (9) are determined by one of many static rules (e.g., the Metropolis rule [49] that we adopt in this paper) which keeps them constant during the estimation. Considering that nodes may be working under different signal-to-noise ratios (SNRs), several adaptive rules have been proposed to optimize the algorithm behavior [49]–[51]. However, these adaptive rules are severely polluted when impulsive noise samples appear, since the output errors at nodes directly participate in the adaptation of $c_{m,k}$. Designing a robust adaptive rule is an alternative, but it is not the focus of this paper. In another approach, based on the detection of impulsive noise, Ahn et al. proposed a robust variable weighting coefficients dLMS (RVWC-dLMS) algorithm which sets the weighting coefficients to zero at nodes disturbed by impulsive noise [52]. Likewise, the RVWC scheme can be extended to dRLS in a straightforward way, resulting in the RVWC-dRLS algorithm with robustness in impulsive noises as can be seen in the simulations later on.

1In the literature, the RVWC scheme was presented for more general diffusion strategies (namely, also exchanging information among nodes in adaptation step). However, here we do not consider this case for a fair comparison. Besides, such general diffusion strategies require higher computational complexity and communication load [5].
B. R-dRLS Algorithm

An impulsive noise sample at time instant $i$ might lead the dRLS algorithm to diverge via $e_k(i)$ in (5) due to its large amplitude and the propagation of its effect. This degradation effect can last for many iterations. To endow the algorithm with robustness in impulsive noise scenarios, we propose to minimize (3) under the following constraint:

$$\|\psi_{k,i} - w_{k,i-1}\|_2^2 \leq \xi_k(i-1),$$

(10)

where $\xi_k(i-1)$ is a positive bound. A similar constraint appeared in an adaptive filter for a single agent scenario [35], but when generalizing to the distributed version with multiple agents, the constraint could be imposed on the adaptation at all the nodes. This constraint represents that the energy (squared norm) of the update at every node $k$ from $w_{k,i-1}$ to $\psi_{k,i}$ always does not exceed the amount $\xi_k(i-1)$ regardless of the type of noise (possibly, impulsive noise), thereby guaranteeing the robustness of the algorithm. In doing so, if (5) satisfies (10), i.e.,

$$\|g_{k,i}\|_2 |e_k(i)| \leq \sqrt{\xi_k(i-1)},$$

(11)

where $g_{k,i} \triangleq P_{k,i}u_k,i$ represents the Kalman gain vector, then (5) is a solution of the above constrained minimization problem. Conversely, if (10) is not satisfied (usually in the case of appearance of impulsive noise), i.e.,

$$\|g_{k,i}\|_2 |e_k(i)| > \sqrt{\xi_k(i-1)},$$

we propose to replace the update (5) by its normalized form to satisfy the equality in (10), which is described by

$$\psi_{k,i} = w_{k,i-1} + \sqrt{\xi_k(i-1)} \frac{g_{k,i}}{|g_{k,i}|} \text{sign}(e_k(i)),$$

(12)

where sign($\cdot$) is the sign function. Thus, combining (5), (11) and (12), we obtain the adaptation step for each node $k$ as:

$$\psi_{k,i} = w_{k,i-1} + \min \left[ \frac{\sqrt{\xi_k(i-1)}}{|g_{k,i}|} |e_k(i)|, 1 \right] g_{k,i} e_k(i).$$

(13)

Evidently, the crucial problem is how to properly choose the bound $\xi_k(i)$ as it controls the robustness of the algorithm against impulsive noise and influences its dynamic behavior. To be more specific, we wish $\xi_k(i)$ to have larger values at the earlier adaptation stage to provide a fast initial convergence, while for enforcing good robustness against impulsive noise, its values cannot be too large. In addition, we also wish to obtain a small estimation error at steady-state, so $\xi_k(i)$ should be reduced to a small value. Based on these requirements, we consider the equality in (10) to propose a useful recursive method for adjusting $\xi_k(i)$, as described by

$$\zeta_k(i) = \beta \xi_k(i-1) + (1 - \beta) \|\psi_{k,i} - w_{k,i-1}\|_2^2$$

$$= \beta \xi_k(i-1) + (1 - \beta) \min[\|g_{k,i}\|_2^2 |e_k(i)|, \xi_k(i-1)],$$

$$\xi_k(i) = \sum_{m \in N_k} c_{m,k} \zeta_m(i),$$

(14)

where $\beta$ is a memory factor with $0 < \beta < 1$. At every node $k$, $\xi_k(i)$ can be initialized by $\xi_k(0) = E_c \sigma_{u_k}^2 / (M \sigma_{e_k}^2)$, where $E_c$ is a positive integer, and $\sigma_{u_k}^2$ and $\sigma_{e_k}^2$ are powers of the output measurement $d_k(i)$ and the input regressor $u_k,i$, respectively. As one can see in (14), every node $k$ not only uses its own adaptive rule to update $\xi_k(i)$, but also exploits the side information $\zeta_m(i)$ transmitted from its neighboring nodes by the diffusion cooperation. In doing so, the proposed R-dRLS algorithm is more effective at computing consistent estimates at all nodes, which will be observed in Section VI-A.

Table I details the proposed R-dRLS algorithm together with the NC method.

**Remark 2:** As can be seen from (13), the operation mode of the proposed R-dRLS algorithm in the adaptation step can be as follows. At time instant $i$, if $\|g_{k,i}\|_2^2 |e_k(i)| \leq \xi_k(i-1)$, the classical RLS update is performed. If not, the squared norm of the RLS increment is limited to the amount $\xi_k(i-1)$ as in (12) for guaranteeing the robustness in impulsive noise. At the early iterations, the values of $\xi_k(i-1)$ can be high compared to $\|g_{k,i}\|_2^2 |e_k(i)|$ so that the algorithm will behave as the dRLS algorithm, providing a fast initial convergence. On the other hand, whenever an impulsive noise sample appears, due to its significant magnitude, the R-dRLS algorithm will work as an dRLS update multiplied by a very small scaling factor $\frac{\sqrt{\xi_k(i-1)}}{|g_{k,i}|} |e_k(i)|$. It has been shown in [53], [54] that in the adaptation update term, the multiplication of a small scaling factor can reduce the negative influence of impulsive noise on the estimation. Thus, this also indirectly implies that the R-dRLS algorithm has robustness against impulsive noise.
noise. Moreover, the robustness is further maintained over the iterations, due to the decreasing property of \( \xi_k(i) \) given by (14). In addition to this, the diminishing \( \xi_k(i) \) also leads to a reduction in the steady-state error of the algorithm. To sum up, the R-dRLS algorithm can be considered as an improved dRLS algorithm with a variable ‘step-size’ scheme, which has an automatic switch between 1 and \( \frac{\sqrt{\xi_k(i-1)}}{\|g_{k,i}\|_2\|x_k(i)\|} \), as can be observed in (13).

C. NC Method

As a consequence of the diminishing sequence \( \{\xi_k(i)\} \), the R-dRLS algorithm has poor ability of tracking (i.e., re-convergence of the algorithm) after \( w^o \) undergoes an abrupt change. In order to overcome this problem, inspired by the idea in [21] for the single-agent scenario, we propose here a change. In order to overcome this problem, inspired by the dRLS algorithm in impulsive noise. Assuming that the vector \( \mathbf{P} \) can track this change rapidly. Meanwhile, \( V \) and \( \tau \) both undergo an abrupt change at the last time instant.

From Step 1, one can see that using a larger \( V \), the algorithm has lower steady-state error but a higher delay in tracking. Moreover, for a large occurrence probability of impulsive noise, the value of \( V \) should be increased to better discard the impulsive noise samples in the computation of \( \Delta_k(i) \). From our extensive simulations, we found that for both \( V_1 \) and \( V_2 \), good choices are \( V_1 = 0.75V \) and \( V_2 = 0.75V \) (21).

Step 2: If \( \Delta_k(i) > \tau_h \), where \( \tau_h \) is a predefined threshold, it is decided that a change of \( w^o \) has occurred. Then, we reset \( \xi_k(i) \) to its initial value \( \xi_k(0) \) so that the R-dRLS algorithm can track this change rapidly. Meanwhile, \( P_{k,i} \) should also be re-initialized with \( P_{k,0} \).

It is worth noting that in this scheme the parameters \( \tau \), \( \theta \), and \( \tau_h \) are not affected by each other so that their choices are simplified.

IV. MEAN SQUARE PERFORMANCE ANALYSES

A. Steady-state Behavior

In this section, we discuss the steady-state behavior of the R-dRLS algorithm in impulsive noise. Assuming that the vector \( w^o \) is invariant, then we define the estimate deviation and intermediate estimate deviation vectors respectively as:

\[
\mathbf{\tilde{w}_{k,i}} = w^o - w_{k,i}, \quad \mathbf{\tilde{\psi}_{k,i}} = w^o - \psi_{k,i}.
\]

Using these definitions and (14), it is easy to rearrange (13) and (18), respectively, as:

\[
\mathbf{\tilde{\psi}_{k,i}} = \mathbf{w}_{k,i-1} - \frac{\xi_k(i) - \beta \xi_k(i-1)}{1 - \beta} \mathbf{g}_{k,i} \mathbf{sign}(e_{k,i}),
\]

and

\[
\mathbf{w}_{k,i} = \sum_{m \in N_k} c_{m,k} \mathbf{\tilde{\psi}_{m,i}}.
\]

Equating the squared \( l_2 \)-norm of both sides of (17) and then taking the expectation, we obtain

\[
E \left\{ \|\mathbf{\tilde{\psi}_{k,i}}\|_2^2 \right\} = E \left\{ \|\mathbf{w}_{k,i-1}\|_2^2 \right\} - 2E \left\{ \frac{\xi_k(i) - \beta \xi_k(i-1) \mathbf{w}_{k,i-1}^T \mathbf{g}_{k,i} \mathbf{sign}(e_{k,i})}{1 - \beta} \right\} + E \left\{ \frac{\xi_k(i) - \beta E \{\xi_k(i-1)\}}{1 - \beta} \right\}.
\]

Likewise treating (18) and applying Jensen’s inequality [55 p.77], we obtain

\[
E \left\{ \|\mathbf{w}_{k,i}\|_2^2 \right\} \leq \sum_{m \in N_k} c_{m,k} E \left\{ \|\mathbf{\tilde{\psi}_{m,i}}\|_2^2 \right\}.
\]

Typically, \( \beta \) is close to 1 so that the variances of \( \xi_k(i) \) and \( \xi_k(i) \) given in (14) would be small enough. Accordingly, it can be assumed that

\[
E \left\{ \frac{\xi_k(i) - \beta \xi_k(i-1) \mathbf{w}_{k,i-1}^T \mathbf{g}_{k,i} \mathbf{sign}(e_{k,i})}{1 - \beta} \right\} \approx \frac{\mathbf{w}_{k,i-1}^T \mathbf{g}_{k,i} \mathbf{sign}(e_{k,i})}{1 - \beta} E \{\xi_k(i)\} - \beta E \{\xi_k(i-1)\}
\]

Then, with this approximation, (19) is changed to

\[
E \left\{ \|\mathbf{\tilde{\psi}_{k,i}}\|_2^2 \right\} = E \left\{ \|\mathbf{w}_{k,i-1}\|_2^2 \right\} - 2 \times \frac{E \{\xi_k(i)\} - \beta E \{\xi_k(i-1)\}}{1 - \beta} \times \frac{\mathbf{w}_{k,i-1}^T \mathbf{g}_{k,i} \mathbf{sign}(e_{k,i})}{\|\mathbf{g}_{k,i}\|_2^2} + \frac{E \{\xi_k(i)\} - \beta E \{\xi_k(i-1)\}}{1 - \beta}.
\]

To deal with the (a) term in (22), some assumptions are helpful.

Assumption 1: The input regressors \( u_{k,i} \) are zero-mean with covariance matrices \( \mathbf{R}_k = E \{u_{k,i}u_{k,i}^T\} \) and spatially independent.

Assumption 2: The regressors \( \{u_{k,i}\} \) are independent of the estimates \( \{w_{m,j}\} \) for \( j \leq i \) and all \( k, m \), referred to as
the *independence assumption*, which is known as useful in the analysis of adaptive algorithms [43] and distributed estimation algorithms [3, 56].

**Assumption 3:** There is an iteration number \( i_0 \) such that for all \( i > i_0 \), the time-averaged matrix \( \Phi_{k,i} \) at every node \( k \) can be replaced by its expected value \( E \{ \Phi_{k,i} \} \). This is an ergodicity assumption since for all \( \Phi \)

\[
\lim_{i \to \infty} E \{ \Phi_{k,i} \} = \frac{R_k}{1-\lambda} = \Phi_k.
\]

(23)

Correspondingly, we can also replace the random matrix \( \Phi_{k,i}^{-1} \) by \( \Phi_k^{-1} = E \{ \Phi_k^{-1} \} \) for a sufficiently large number of iterations \( i \). Note that such replacements are commonly used in the performance analysis of RLS-type algorithms, see [31], [32], [43], [57] and the references therein.

Applying assumption 3, we are able to represent the term (a) in (22) as:

\[
E \left\{ \frac{\bar{w}_{k,i}^T g_{k,i} \| e_k(i) \|}{g_{k,i}^T g_{k,i}} \right\} \\
\approx E \left\{ \frac{\bar{w}_{k,i}^T \Phi_k^{-1} u_{k,i} \| e_k(i) \|}{\sqrt{u_{k,i}^T \Phi_k^{-2} u_{k,i}}} \right\} \approx E \left\{ \frac{\bar{w}_{k,i}^T R_k^{-1} u_{k,i} \| e_k(i) \|}{\sqrt{u_{k,i}^T R_k^{-2} u_{k,i}}} \right\}.
\]

(24)

In the light of assumption 1, if the dimension of \( w^o \) is large, i.e., \( M \gg 1 \), the fluctuation of the denominator term in (24) from one iteration to the next can be assumed to be small. So, we could make the following approximation (which is also verified in Appendix A):

\[
E \left\{ \frac{\bar{w}_{k,i}^T R_k^{-1} u_{k,i}}{\sqrt{u_{k,i}^T R_k^{-2} u_{k,i}}} \right\} \approx \chi_k E \left\{ e_{a,k}^{-1}(i) \| e_k(i) \| \right\},
\]

where

\[
e_{a,k}^{-1}(i) \triangleq \bar{w}_{k,i}^T R_k^{-1} u_{k,i},
\]

\[
e_{a,k}^{-1}(i) \triangleq \bar{w}_{k,i}^T R_k^{-1} u_{k,i},
\]

\[
\chi_k = E \left\{ \frac{1}{\sqrt{u_{k,i}^T R_k^{-2} u_{k,i}}} \right\}.
\]

(25)

(26)

Considering the presence of impulsive noise, we need the following assumptions to continue the analysis.

**Assumption 4:** At every node \( k \), the additive noise \( v_k(i) \) is drawn from a Gaussian random process, \( v_k(i) = \theta_k(i) + \eta_k(i) \), where \( \theta_k(i) \) is the background noise assumed to be zero-mean white Gaussian with variance \( \sigma_{\theta,k}^2 \). The impulsive part \( \eta_k(i) \) is described as \( \eta_k(i) = b_k(i) g_k(i) \), where \( b_k(i) \) is drawn from a Bernoulli random process with the probability \( P[b_k(i) = 1] = p_{r,k} \), and \( g_k(i) \) is drawn from a white Gaussian process with zero-mean and variance \( \sigma_{g,k}^2 = h \sigma_{\theta,k}^2 \), \( h \gg 1 \). Usually, \( p_{r,k} \) is also called the appearance probability of an impulsive noise sample.

Then, the mean and variance of \( v_k(i) \) are zero and \( \sigma_{v,k}^2 = p_{r,k} (h + 1) \sigma_{\theta,k}^2 + (1 - p_{r,k}) \sigma_{g,k}^2 \), respectively. Note that, only when \( p_{r,k} = 0 \) or \( 1 \), \( v_k(i) \) is Gaussian; otherwise, \( v_k(i) \) is non-Gaussian. Also, \( v_k(i) \) conditioned on \( b_k(i) \) is Gaussian [44]. Although the \( \alpha \)-stable process is more appropriate for modeling impulsive noise in practice [20], [45], [46], one would not consider it in the algorithm’s analysis because its probability density function has no explicit form. Accordingly, the above assumption was used frequently for performance analysis of adaptive algorithms in impulsive noise environments, providing mathematical tractability [21], [27], [43], [58].

Furthermore, as pointed out in [59], when \( M \gg 1 \), then by using the central limit theorem, it can be assumed that \( e_{a,k}(i) \) and \( \sigma_{a,k}^{-2}(i) \) are zero mean Gaussian variables for any constant matrix \( \Sigma \). Then, we can employ the following Lemma.

**Lemma:** Let \( e_a \) and \( u \) be jointly Gaussian zero-mean random variables. Let \( e = e_a + v \), where \( v \) is a zero-mean CG random variable with variance \( \sigma_v^2 = p_{r,k} (h + 1) \sigma_{\theta,k}^2 + (1 - p_{r,k}) \sigma_{g,k}^2 \), and \( v \) is independent of \( e_a \) and \( u \). If \( e_1 = e_a + \omega_1 \) and \( e_2 = e_a + \omega_2 \), where \( \omega_1 \) and \( \omega_2 \) are zero-mean Gaussian random variables with variances \( \sigma_{\omega_1}^2 = (h + 1) \sigma_{\theta,k}^2 + \sigma_{\omega_2}^2 = \sigma_{g,k}^2 \), and are independent of \( u \) and \( e_a \), then

\[
E \{ \text{sign}(e) u \} = p_r E \{ \text{sign}(e_1) u \} + (1 - p_r) E \{ \text{sign}(e_2) u \}.
\]

Such a Lemma has been commonly used in the past for analyzing the sign-based algorithms [27], [58]. Based on Price’s theorem in [60], Lemma and assumption 2, we can establish the following equation

\[
E \left\{ e_{a,k}^{-1}(i) \| e_k(i) + v_k(i) \| \right\} \approx \varpi_{k,i} E \left\{ \frac{R_k^{-1}}{e_{a,k}(i) e_{a,k}(i) \| \bar{w}_{k,i-1} \|} \right\},
\]

(27)

where

\[
\varpi_{k,i} = \sqrt{\frac{2}{\pi}} \left\{ \frac{p_{r,k}}{\sqrt{E \{ e_{a,k}^{-2}(i) \}} + (h + 1) \sigma_{\theta,k}^2}}
\]

\[+ \frac{1 - p_{r,k}}{\sqrt{E \{ e_{a,k}^{-2}(i) \}} + \sigma_{g,k}^2} \right\} 
\]

(28)

and the notation \( E \{ s | q \} \) accounts for the expectation of \( s \) conditioned on \( q \). Subsequently, the right-hand term in equality (25) becomes

\[
E \left\{ e_{a,k}^{-1}(i) \| e_k(i) + v_k(i) \| \right\} \\
= E \left\{ E \left\{ \frac{R_k^{-1}}{e_{a,k}^{-1}(i) e_{a,k}(i) \| \bar{w}_{k,i-1} \|} \right\} \right\} \\
= \varpi_{k,i} E \left\{ \| \bar{w}_{k,i-1} \| \right\},
\]

(a)

(29)

where the equality (a) is the result of using (26) under assumption 2.
Substituting (25) and (29) into (22), it is rearranged as
\[
E \left\{ \| \tilde{w}_{k,i} \|_2^2 \right\} = E \left\{ \| \tilde{w}_{k,i-1} \|_2^2 \right\} - 2\chi_k \omega_k A \sqrt{E \left\{ \zeta_k(i) \right\} - \beta E \left\{ \xi_k(i-1) \right\}} E \left\{ \| \tilde{w}_{k,i-1} \|_2^2 \right\} + E \left\{ \zeta_k(i) \right\} - \beta E \left\{ \xi_k(i-1) \right\}.
\]
(30)

Next, we introduce the following network global vectors:
\[
\chi_i \equiv \text{col} \left\{ E \left\{ \| \tilde{w}_{1,i} \|_2^2 \right\}, ..., E \left\{ \| \tilde{w}_{N,i} \|_2^2 \right\} \right\},
\]
\[
\omega_i \equiv \text{col} \left\{ E \left\{ \| \tilde{w}_{1,i} \|_2^2 \right\}, ..., E \left\{ \| \tilde{w}_{N,i} \|_2^2 \right\} \right\},
\]
and the network global matrices
\[
\Lambda_i \equiv \text{diag} \left\{ \chi_1 \omega_1, ..., \chi_N \omega_N \right\},
\]
\[
\Omega_i \equiv \text{diag} \left\{ E \left\{ \left( \zeta_1(i) \right) \right\} - \beta E \left\{ \xi_1(i-1) \right\}, ..., E \left\{ \left( \zeta_N(i) \right) \right\} - \beta E \left\{ \xi_N(i-1) \right\} \right\}.
\]
(32)

Also, we define the matrix \( C \) to collect the combination coefficients, i.e., \( C_{m,k} = c_{m,k} \). Following (31) and (32), we can formulate (20) and (30) for all nodes as follows:
\[
\omega_i \leq C^T \chi_i
= C^T \left[ \omega_{i-1} - 2\Lambda_i \Omega_i \omega_{i-1} + \Omega_i^2 1_M \right].
\]
(33)

Taking the \( \infty \)-norm for both sides of (33) leads to
\[
\| \omega_i \|_\infty \leq \| C^T \left[ \omega_{i-1} - 2\Lambda_i \Omega_i \omega_{i-1} + \Omega_i^2 1_M \right] \|_\infty
\leq \| C^T \| \| \omega_{i-1} - 2\Lambda_i \Omega_i \omega_{i-1} + \Omega_i^2 1_M \|_\infty
\equiv \| \omega_{i-1} - 2\Lambda_i \Omega_i \omega_{i-1} + \Omega_i^2 1_M \|_\infty.
\]
(34)

where the equality (a) uses the fact that \( \| C^T \| = 1 \) in that the summation of each column of \( C \) is 1. Since \( \Lambda_i \) and \( \Omega_i \) are diagonal matrices with positive entries, (34) can be equivalently expressed as [12]:
\[
E \left\{ \| \tilde{w}_{k,i} \|_2^2 \right\} \leq E \left\{ \| \tilde{w}_{k,i-1} \|_2^2 \right\}
- 2\chi_k \omega_k A \sqrt{E \left\{ \zeta_k(i) \right\} - \beta E \left\{ \xi_k(i-1) \right\}} E \left\{ \| \tilde{w}_{k,i-1} \|_2^2 \right\}
+ E \left\{ \zeta_k(i) \right\} - \beta E \left\{ \xi_k(i-1) \right\}
\]
for \( k = 1, ..., N \). When the algorithm reaches the steady-state, i.e., \( E \left\{ \| \tilde{w}_{k,i} \|_2^2 \right\} = E \left\{ \| \tilde{w}_{k,i-1} \|_2^2 \right\} \) as \( i \to \infty \), from (35) we will get:
\[
2\chi_k \omega_k \lim_{i \to \infty} E \left\{ \| \tilde{w}_{k,i-1} \|_2^2 \right\} \leq \lim_{i \to \infty} E \left\{ \zeta_k(i) \right\} - \beta E \left\{ \xi_k(i-1) \right\}.
\]
(36)

In view of the result that \( E \{ \zeta_k(i) \} \) and \( E \{ \xi_k(i) \} \) converge approximately to 0 as \( i \to \infty \) (see Appendix B) as well as \( \chi_k \neq 0 \) and \( \omega_{k,i} \neq 0 \), thus, from (36) we are able to deduce that
\[
E \left\{ \| \tilde{w}_k(i) \|_2^2 \right\} \approx 0, \quad \text{for} \quad k = 1, ..., N.
\]
(37)

As a result, (37) illustrates that based on given assumptions, the R-dRLS algorithm can converge to the true parameter vector in the mean-square sense after enough iterations even in impulsive noise environments.

### B. Analysis of Evolution Behavior

The result (37) is qualitative so that it does not predict the steady-state performance of the algorithm, due mainly to the use of the upper bound relation (20). In this subsection, we will establish a recursive model to describe the evolution behavior of the algorithm in impulsive noise. We start by defining the following network vectors collected from all nodes:
\[
\tilde{\psi}_i \equiv \text{col} \{ \tilde{\psi}_{1,i}, ..., \tilde{\psi}_{N,i} \},
\]
\[
\tilde{w}_i \equiv \text{col} \{ \tilde{w}_{1,i}, ..., \tilde{w}_{N,i} \},
\]
\[
\Xi_i \equiv \text{col} \{ \Xi_{1,i}, ..., \Xi_{N,i} \},
\]
(38)

where
\[
\Xi_{k,i} = \sqrt{\zeta_k(i) - \beta \xi_k(i-1)} \frac{g_{k,i}}{|g_{k,i}|} \text{sign}(e_k(i))
\]
(39)

for nodes \( k = 1, ..., N \). By these defined vectors, we can associate (17) with (18) at all the nodes:
\[
\bar{w}_i = C^T [\tilde{w}_{i-1} - \Xi_i],
\]
(40)

where \( C = C \otimes I_M \). Post-multiplying (40) by its transpose and taking the expectation, we have
\[
W_i = C^T \left[ W_{i-1} - E \left\{ \bar{w}_{i-1} \Xi_i^T \right\} - E \left\{ \Xi_i \Xi_i^T \right\} \right] C,
\]
(41)

where \( W_i = E \{ \tilde{w}_i \tilde{w}_i^T \} \) denotes the covariance matrix of the deviation vector \( \tilde{w}_i \), and its \( k \)-th diagonal block of size \( M \times M \), i.e., \( W_{k,i} = E \{ \tilde{w}_{k,i} \tilde{w}_{k,i}^T \} \), represents the covariance matrix of the deviation vector \( \tilde{w}_{k,i} \) at node \( k \).

To evaluate terms I-III in (41), in addition to the spatially independence in assumption 1, we also require the input regressors \( u_{k,i} \) to be statistically independent in time, which is also often used in analysis of distributed estimation algorithms [2], [3]. Therefore, performing similar manipulations as in Section IV-A on the expectations under assumptions 2-4, Lemma and Price’s theorem, we can compute these three terms. Specifically, the term I in (41) becomes
\[
E \{ \bar{w}_{i-1} \Xi_i^T \} = E \{ \tilde{w}_{i-1} \Xi_i^T \bar{w}_{i-1} \}
= W_{i-1} \{ (\Lambda_i \Omega_i) \otimes I_M \},
\]
(42)
where we rewrite $\varpi_{k,i}$ contained in $\Lambda_i$ as

$$
\varpi_{k,i} = \sqrt{\frac{2}{\pi}} \left\{ \frac{\mu_{r,k}}{\sqrt{\text{Tr}(W_{k,i-1}R_k) + (h+1)\sigma^2_{\theta}}} + \frac{1 - \mu_{r,k}}{\sqrt{\text{Tr}(W_{k,i-1}R_k + \sigma^2_{\theta})}} \right\}.
$$

(43)

The term in (43) is the transpose of (43). For any $k$ and $m$ belonging to the set $\{1, \ldots, N\}$, we define the $(m,k)$-th $M \times M$ matrix $E\{\Xi, \Xi^T\}$ as follows:

$$
E\{\Xi, \Xi^T\}_{m,k} = E\{\Xi_m, \Xi_{k,i}^T\}.
$$

When $k = m$, (44) represents the k-th diagonal block of $E\{\Xi, \Xi^T\}$, which is described as

$$
E\{\Xi, \Xi^T\}_{k,k} = \Omega_{k,i}^2 E\left\{ \frac{R^{-1}_{k} u_{k,i} u_{k,i}^T R^{-1}_{k}}{u_{k,i}^T R^{-2}_{k} u_{k,i}} \right\},
$$

(45)

where $\Omega_{k,i}$ is the $k$-th element of $\Omega_i$.

From (45) and (46), we obtain the term III in (41):

$$
E\{\Xi, \Xi^T\} = ((\Lambda_i, \Omega_i) \otimes I_M)[W_{i-1} - \hat{W}_{i-1}]\times ([\Lambda_i, \Omega_i] \otimes I_M) + \hat{R},
$$

(47)

where

\[
\hat{W}_{i-1} = \text{diag}\{W_{1,i-1}, \ldots, W_{N,i-1}\},
\]

\[
\hat{R} = \text{diag}\left\{ \Omega_{1,i}^2 E\left\{ \frac{R^{-1}_{1} u_{1,i} u_{1,i}^T R^{-1}_{1}}{u_{1,i}^T R^{-2}_{1} u_{1,i}} \right\}, \ldots, \right\},
\]

\[
\Omega_{N,i}^2 E\left\{ \frac{R^{-1}_{N} u_{N,i} u_{N,i}^T R^{-1}_{N}}{u_{N,i}^T R^{-2}_{N} u_{N,i}} \right\} \right\}.
\]

By substituting (42) and (47) into (41), we obtain the recursive expression for $W_i$:

$$
W_i = C^T [W_{i-1} - \hat{W}_{i-1}([\Lambda_i, \Omega_i] \otimes I_M) - ([\Lambda_i, \Omega_i] \otimes I_M)^T W_{i-1}^T + ([\Lambda_i, \Omega_i] \otimes I_M) \times \left[ W_{i-1} - \hat{W}_{i-1} \right]] ((\Lambda_i, \Omega_i) \otimes I_M) + \hat{R} \right] C.
$$

(49)

The mean square deviation (MSD) at node $k$ is defined as MSD$\xi_k(i) \triangleq \text{Tr}\{W_k(i)\}$, and the network MSD over all the nodes is defined as MSD$_{\text{net}}(i) = 1/N \sum_{k=1}^N$ MSD$\xi_k(i) = \text{Tr}\{W_i\}/N$ [2]. Equation (49) models the MSD evolution behavior of the algorithm. It needs to be mentioned that to implement the model (49), $E\{\xi_k\}$ and $E\{\xi_k^2\}$ defined in $\Omega_i$ still need to be evaluated further. However, as shown in (14), $\xi_k$ and $\xi_k^2$ between interconnected nodes are affected by each other and there is a comparison operation, so it is difficult to provide an evolution expression for them. In this paper, we suggest that $E\{\xi_k\}$ and $E\{\xi_k^2\}$ are obtained by the ensemble average using simulations. Consequently, although (49) is a semi-analytic result, it can also be used to evaluate the convergence of the proposed algorithm.

V. DCD-BASED ALGORITHMS

In this section, we review the DCD-dRLS algorithm from [42], and then develop a robust DCD-dRLS algorithm.

A. The Original DCD-dRLS Algorithm

Since the dRLS algorithm involves the matrix operation of size $M \times M$ in the computations of $g_{k,i}$ and $g_{k,i}$ at every node, it requires a computational complexity that scales as a quadratic function of $M$ in terms of additions and multiplications per iteration $i$. To reduce the complexity, the adaptation step of the DCD-dRLS algorithm is described as [42]:

$$
\psi_{k,i} = w_{k,i-1} + \Delta w_{k,i},
$$

(50)

where the increment $\Delta w_{k,i}$ is obtained by solving the normal equation:

$$
\Phi_{k,i} \Delta w_{k,i} = b_{k,i},
$$

(51)

$$
b_{k,i} = \lambda r_{k,i-1} + c_{k}(t) u_{k,i},
$$

(52)

$r_{k,i-1}$ defines the residual vector at node $k$ at time instant $i-1$:

$$
r_{k,i} = b_{k,i-1} - \Phi_{k,i-1} \Delta w_{k,i-1}.
$$

(53)

For reducing the complexity of computing $\Delta \hat{w}_{k,i}$ and $r_{k,i}$, the DCD method presented in Table II is used; see [38]–[40] for details. In Table II, $r_{k,i}$ is the $l$-th entry of a vector $r_{k,i}$, and $\Phi_{k,i}$ is the $l$-th entry and the $l$-th column of $\Phi_{k,i}$, respectively.

<table>
<thead>
<tr>
<th>Parameters: $H$, $N_u$, $M_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization: $\Delta w_{k,i} = 0$, $r_{k,i} = b_{k,i}$, $y = 1$, $\mu = H/2$</td>
</tr>
<tr>
<td>$t = \text{arg max}_{l=1}^{N_u}</td>
</tr>
<tr>
<td>while $</td>
</tr>
<tr>
<td>$y = y + 1$, $\mu = H/2$</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if $y &gt; M_b$</td>
</tr>
<tr>
<td>break</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>$[\Delta \hat{w}_{k,i}]<em>l = [\Delta \hat{w}</em>{k,i}]<em>l + \mu \text{sign}([r</em>{k,i}]_l)$</td>
</tr>
<tr>
<td>$r_{k,i} = r_{k,i} - \mu \text{sign}([r_{k,i}]<em>l)[\Phi</em>{k,i}]_l$</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

(51)

The accuracy and complexity of the DCD method are dependent on three parameters: $H$, $M_b$, and $N_u$. In general, $H$ is chosen as a power-of-two number; $M_b$ is the number of bits being enough for a fixed-point representation of $\Delta \hat{w}_{k,i}$ within an amplitude range $[-H, H]$; and $N_u$ defines a maximum number of elements in $\Delta \hat{w}_{k,i}$ that can be updated at a time instant. The DCD method only requires $2N_u + M_b$ additions at most at each time instant with no multiplication [38]. Also, a larger $N_u$ makes the solution $\Delta \hat{w}_{k,i}$ closer to the optimal solution $\Delta w_{k,i}$ in (51), but increases the number of additions. It follows that if $N_u < M$, the DCD-based algorithm implements a selective partial update (51).

Similar to the dRLS algorithm, however, the DCD-dRLS algorithm will also encounter the performance deterioration when impulsive noise happens.
TABLE III
PROPOSED DCD-R-dRLS ALGORITHM

<table>
<thead>
<tr>
<th>Parameters: $0 &lt; \beta &lt; 1$, $\lambda$, $\delta$ and $E_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization: $w_{0,k} = 0$, $\Phi_{k,0} = \delta I_M$ and $\xi_k(0) = E_c \frac{\sigma^2_w}{\rho}$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4: To achieve robustness against impulsive noise, we present a solution to the DCD-R-dRLS algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi_{k,i}$ denotes the cardinality of $\mathcal{N}<em>k$. For shift structured input regressor at node $k$, we need to recalculate $b</em>{k,i}$ in (51) as: $b_{k,i} = \lambda r_{k,i-1} + \frac{\sqrt{\xi_k(i-1)}}{|\Delta \hat{w}_{k,i}|<em>2^2} e_k(i) u</em>{k,i}$.</td>
</tr>
</tbody>
</table>

B. Proposed DCD-R-dRLS Algorithm

To achieve robustness against impulsive noise, we present here the DCD-R-dRLS algorithm.

**Step 1:** At every node $k$, we firstly use the DCD method to solve the normal equation (51) with (4) and (52), yielding a solution $\Delta \hat{w}_{k,i}^{(1)}$ and residual vector $r_{k,i}^{(1)}$. In the presence of impulsive noise, we also impose a constraint similar to that in (10):

$$\|\Delta \hat{w}_{k,i}\|_2^2 \leq \xi_k(i-1). \tag{54}$$

**Step 2:** If $\|\Delta \hat{w}_{k,i}\|_2^2 \leq \xi_k(i-1)$, we set $\Delta \hat{w}_{k,i} = \Delta \hat{w}_{k,i}^{(1)}$ and $r_{k,i} = r_{k,i}^{(1)}$ and then perform the update (50). Otherwise, we need to recalculate $b_{k,i}$ in (51) as:

$$b_{k,i} = \lambda r_{k,i-1} + \frac{\sqrt{\xi_k(i-1)}}{\|\Delta \hat{w}_{k,i}^{(1)}\|_2^2} e_k(i) u_{k,i}. \tag{55}$$

Subsequently, based on the DCD method, we obtain the solution $\Delta \hat{w}_{k,i}^{(2)}$ and the residual vector $r_{k,i}^{(2)}$ from the normal equation (51) under (4) and (55), thereby performing the update (50) with the increment

$$\Delta \hat{w}_{k,i} = \frac{\sqrt{\xi_k(i-1)}}{\|\Delta \hat{w}_{k,i}^{(2)}\|_2^2} \Delta \hat{w}_{k,i}^{(2)}. \tag{56}$$

and $r_{k,i} = r_{k,i}^{(2)}$.

**Step 3:** The combination step (12) is performed.

**Step 4:** The bound parameter $\xi_k(i)$ in the DCD-R-dRLS algorithm is updated according to

$$\xi_k(i) = \beta \xi_k(i-1) + (1 - \beta) \|\Delta \hat{w}_{k,i}\|_2^2,$$

$$\xi_k(i) = \sum_{m \in \mathcal{N}_k} c_{m,k} \xi_m(i). \tag{57}$$

Table III summarizes the DCD-R-dRLS algorithm.

**Remark 3:** An impulsive noise sample appearing at time instant $i$ would yield a mismatch solution $\Delta \hat{w}_{k,i}^{(1)}$ so that $\|\Delta \hat{w}_{k,i}^{(1)}\|_2^2 > \xi_k(i-1)$. In this case, the scaling factor $\frac{\sqrt{\xi_k(i-1)}}{\|\Delta \hat{w}_{k,i}^{(1)}\|_2}$ in (55) is small enough to eliminate impulsive noise hidden in $e_k(i)$. A similar scaling factor $\frac{\sqrt{\xi_k(i-1)}}{\|\Delta \hat{w}_{k,i}^{(2)}\|_2}$ in (56) is to make the increment satisfy the constraint (54). Consequently, the DCD-R-dRLS algorithm improves the robustness to impulsive noise relative to the DCD-dRLS algorithm. Moreover, the decreasing sequence $\{\xi_k(i)\}$ shown in (57) further guarantees the robustness. It is worth noting that due to $\|g_{k,i}\|_2 e_k(i) \approx \|\Delta \hat{w}_{k,i}\|_2$, the DCD-R-dRLS algorithm is a DCD-based variant of the R-dRLS algorithm. Unlike the R-dRLS algorithm, based on the NC method we re-initialize $\Phi_{k,i}$ with $\Phi_{k,0}$ to endow the DCD-R-dRLS algorithm with the tracking capability when $w^o$ suddenly changes.

**Remark 4:** Let $C_{dcd}^+$ denote the only required number of additions for the DCD algorithm, with $C_{dcd}^+ \leq 2N_u M + M_b$. In Table IV, we provide the computational complexity of the existing dLMS, dRLS, DCD-dRLS, and both proposed R-dRLS and DCD-R-dRLS algorithms at node $k$ per time instant $i$, where $N_k$ denotes the cardinality of $\mathcal{N}_k$. For shift structured input regressor at node $k$, i.e., $u_{k,i} = [u_k(i), u_k(i-1), \ldots, u_k(i-M + 1)]^T$, where $u_k(i)$ is an input sample at time instant $i$, implementing $\Phi_{k,i}$, only calculating the first column is very simplified. In this situation, by copying the upper-left $(M - 1) \times (M - 1)$ block of $\Phi_{k,i-1}$ leads to the lower-right $(M - 1) \times (M - 1)$ block of $\Phi_{k,i}$. The remaining part of $\Phi_{k,i}$ that needs to be updated is the first row and first column. Owing to symmetry of $\Phi_{k,i}$, only calculating the first column is sufficient, which is formulated as:

$$[\Phi_{k,i}]_{i,1} = \lambda [\Phi_{k,i-1}]_{i,1} + u_k(i) u_{k,i}. \tag{58}$$

Note that, in the DCD-R-dRLS algorithm, $\kappa = 1$ represents the case $\|\Delta \hat{w}_{k,i}\|_2^2 > \xi_k(i-1)$ at time instant $i$ (which leads to the maximum complexity), otherwise $\kappa = 0$. The comparisons required in the R-dRLS and DCD-R-dRLS algorithms are counted as additions.

Consider an example with $n_k = 10$, $M_b = 32$ and $\kappa = 1$. Fig. 2 depicts the number of operations of some diffusion algorithms in terms of multiplications and additions at node $k$ at each time instant versus $M$. It is clear that the computational complexity of the dLMS algorithm, with the order of $O(M)$, is much lower than that of the dRLS algorithm. As expected, since $N_u < M$, compared with the standard dRLS and R-dRLS algorithms, their DCD versions obtain about 50% reduction in both multiplications and additions for the case of general input regressors. However, for shift structured input regressors, the computational cost is drastically reduced from the order $O(M^2)$ to $O(M)$, which is more pronounced in scenarios with large $M$. Moreover, the multiplications required in the DCD-based algorithms are not dependent of $N_u$. On the other hand, in contrast with the existing dRLS and DCD-dRLS algorithms, the additional complexities in the proposed R-dRLS and DCD-R-dRLS algorithms resulted from the computations of the scaling factor and the bound parameter are small. In addition to the complexity, both proposed algorithms, each node $k$ increases communication cost of $n_k - 1$ numbers for transmitting $\xi_k$ to its neighbors.

**Remark 5:** From the DCD-R-dRLS algorithm, we can directly obtain its special form for a single-agent scenario, referred to as the DCD-RLS algorithm. In other words,
TABLE IV

COMPUTATIONAL COMPLEXITY OF ALGORITHMS FOR NODE $k$ PER TIME INSTANT.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Multiplications</th>
<th>Additions</th>
<th>Divisions</th>
<th>Square-root</th>
</tr>
</thead>
<tbody>
<tr>
<td>dLS</td>
<td>$n_k M + 2M + 1$</td>
<td>$n_k M + M$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R-RLS</td>
<td>$n_k M + 4M^2 + 3M$</td>
<td>$n_k M + 2M + C_{dcd}^{+}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DCD-R-RLS</td>
<td>$n_k M + 2M^2 + 3M$</td>
<td>$n_k M + 2M + C_{dcd}^{+}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R-RLS</td>
<td>$n_k M + 5M$</td>
<td>$n_k M + 3M + C_{dcd}^{+}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DCD-R-RLS</td>
<td>$n_k M + (M + 1)$ + $4M^2 + 4M + 5$</td>
<td>$n_k M + (M + 1) + 3M^2 + M + 1$</td>
<td>$M + 1$</td>
<td>1</td>
</tr>
<tr>
<td>DCD-R-RLS</td>
<td>$n_k(M + 1) + 2M^2 + 4M + 3n M + 2$</td>
<td>$n_k(M + 1) + M^2 + 3M + n(2M - 1 + C_{dcd}^{+}) + C_{dcd}^{+}$</td>
<td>$2n$</td>
<td>$2n$</td>
</tr>
</tbody>
</table>

![Fig. 2](image-url). Complexity of the algorithms versus the length of the target vector at node $k$: (a) multiplications and (b) additions.

The Metropolis rule [49] used for computing the combination coefficients $\{c_{m,k}\}$ in combination step is expressed as:

$$c_{m,k} = \begin{cases} 
1/ \max(n_m, n_k), & \text{if } m \in N_k, m \neq k \\
1 - \sum_{m \neq k} c_{m,k}, & \text{if } m = k \\
0, & \text{otherwise}.
\end{cases}$$

A. Distributed Parameter Estimation

The vector $w^o$ to be estimated has a length of $M = 16$ and a unit norm; it is generated randomly from a zero-mean uniform distribution. The input regressor $u_{k,i}$ has a shift structure, where $u_{k,i}(i)$ is colored and generated by a second-order autoregressive system:

$$u_{k,i} = 1.6u_{k,i}(i - 1) - 0.81u_{k,i}(i - 2) + \epsilon_{k,i},$$

where $\epsilon_{k,i}$ is a zero-mean white Gaussian process with variance $\sigma_{\epsilon,k}^2$. The background noise $\theta_{k,i}$ is zero-mean white Gaussian noise with variance $\sigma_{\theta,k}^2$. Variances $\sigma_{\epsilon,k}^2$ and $\sigma_{\theta,k}^2$ are shown in Fig. 3(b) and (c), respectively, for all the nodes. We employ the network MSD to assess the performance of algorithms. All results are the average over 200 independent trials.

![Fig. 3](image-url). (a) topology of network with 20 nodes, and profiles of (b) $\sigma_{\epsilon,k}^2$ and (c) $\sigma_{\theta,k}^2$ per node $k$.

VI. SIMULATION RESULTS

Simulation examples are presented for a diffusion network with $N = 20$ nodes on distributed parameter estimation and distributed spectrum estimation. The network topology adopted for all simulations is shown in Fig. 3(a), unless otherwise specified. Herein, we do not consider the measurement sharing in the adaptation step for all diffusion algorithms.

the DCD-R-RLS algorithm is the DCD implementation of the algorithm presented in [35].
Example 1: Except for the background noise \( \theta_{k}(i) \), a cluster of impulses with length 200 is also added to corrupt \( \theta_{k}(i) \) at iteration \( i = 5001 \) \(^2\). The cluster is drawn from a zero-mean white Gaussian process, but with a large variance \( 10000 \sigma_{y,k}^2 \) to generate impulsive samples, where \( \sigma_{y,k} \) denotes the power of \( y_{k}(i) = u_{k}^T w \). Fig. 4 \(^3\) compares the performance of the proposed R-dRLS algorithm with that of the dRLS and both LTVFF-dRLS and LCTVFF-dRLS algorithms presented in \([57]\). The parameters of the algorithms are set to make a comparable convergence rate. The regularization constant for all RLS-type algorithms is chosen as \( \delta = 0.01 \). It is clear to see, for a small forgetting factor \( \lambda = 0.98 \), the conventional dRLS algorithm converges faster but has a higher estimation error; conversely, by increasing the forgetting factor, it has a lower estimation error but its convergence rate becomes slower. In particular, using a large forgetting factor \( \lambda = 0.998 \), the dRLS will need more time to converge again after a cluster of impulses enforces the algorithm to diverge. Due to the use of variable forgetting factor schemes, both LTVFF-dRLS and LCTVFF-dRLS algorithms solve this performance trade-off issue or not and perform corresponding updates. The R-dRLS algorithm can judge by \( (11) \) whether impulses occur or not and perform corresponding updates.

Example 2: The additive noise \( v_{k}(i) \) is a CG process given in assumption 4. At every node \( k \), we set \( p_{r,k} \) as a random number in the range of \( [0.001, 0.05] \) and \( \sigma_{r,k}^2 = 10000 \sigma_{g,k}^2 \). For a fair comparison of RLS-type algorithms, we choose the same forgetting factor \( \Lambda = 0.985 \) and regularization constant \( \delta = 0.01 \), except \( \delta = 0.5 \) in the dRLP and RVWC-dRLS algorithms.

Fig. 5 \(^4\) checks the validity of the semi-analytic result \( (49) \), where we plot \( E\{\xi_{1}(i)\} \) at node 1 (having similar results at other nodes). To take into account the assumption on input regressors \( u_{k,i} \) in analysis, here its entries are generated from a white Gaussian process \( \epsilon_{k}(i) \). To compute \( (49) \), we use the same impulsive noise parameters: \( p_{r,k} = 0.01 \) or 0.05, and \( \sigma_{g,k}^2 = 10000 \sigma_{g,k}^2 \) at all the nodes. As one can see, the theoretical results have good fit with the simulated results. Moreover, \( E\{\xi_{1}(i)\} \) obtained by the ensemble average of simulations is a decreasing function of the iteration \( i \), which further supports the theory in Appendix B.

Fig. 6 \(^5\) investigates the effect of the NC method on the R-dRLS algorithm. It can be seen that the R-dRLS algorithm will not re-converge after \( w^o \) changes to \( -w^o \) at iteration \( i = 2501 \). In this scenario, all algorithms have a large sharp phase transition of MSD due to the mismatch between \( -w^o \) and its estimate \( w_{k,1} \) at that moment. The NC method can endow the R-dRLS algorithm with good tracking capability for such a change of \( w^o \). Benefited from the smoothing operation \( (15) \), the NC \( (\tau = 0.96) \) only slightly degrades the steady-state performance of the R-dRLS algorithm compared with the non-smooth version in \([11]\) (i.e., \( \tau = 0 \)).

---

\(^2\) Such a scenario is similar to double-talk in echo cancellation.

---

\(^3\) Such a scenario is similar to double-talk in echo cancellation.

---

\(^4\) Such a scenario is similar to double-talk in echo cancellation.

---

\(^5\) Such a scenario is similar to double-talk in echo cancellation.
In Fig. 7, we compare the performance of the dRLS, dSE-LMS, dLMP, RVWC-dRLS, and dRLP algorithms with that of the proposed R-dRLS with NC algorithm. Note that, the R-dRLS (no cooperation) is that each node performs a standalone adaptive algorithm presented in [35]. As expected, the dRLS algorithm has a poor performance in the presence of impulsive noise, while other algorithms are robust. Among these robust algorithms, the convergence of dSE-LMS and dLMP algorithms is slow. Thanks to the decorrelation property of dRLS, the RVWC-dRLS, dRLP, and R-dRLS with NC algorithms obtain fast convergence. In particular, the proposed R-dRLS with NC algorithm has also a large reduction in the steady-state MSD. This is due mainly to the fact that its updated energy described by (10) and (14) diminishes with iterations.

Example 3: The additive noise $v_k(i)$ here is generated by the $\alpha$-stable process, also called the $\alpha$-stable noise. Its characteristic function is given by $\varphi(t) = \exp(-\gamma |t|^\alpha)$ [45], [48], where the characteristic exponent $\alpha \in (0, 2]$ describes the impulsiveness of the noise (smaller $\alpha$ leads to more impulsive noise samples) and $\gamma > 0$ represents the dispersion level of the noise. In particular, when $\alpha = 2$, it reduces to the Gaussian noise. It is rare to find $\alpha$-stable noise with $\alpha < 1$ in practice [45], [48]. In this example, thus we set $\alpha = 1.2$ and $\gamma = 2/15$. The learning performance of the algorithms is shown in Fig. 8. Fig. 9 shows the node-wise steady-state MSD of those algorithms in Fig. 8. By comparing these two figures, it is seen that the proposed R-dRLS algorithm is more likely to reach the same estimates at all nodes.

We also perform the simulations for the network in Fig. 10 with less connections among nodes. Fig. 11 shows the node-wise steady-state MSD of those algorithms in Fig. 9. By comparing these two figures, it is seen that the proposed R-dRLS algorithm is more likely to reach the same estimates at all nodes.
Example 4: Comparison of DCD-algorithms. Figs. 12 and 13 compare the DCD-R-dRLS algorithm using different $N_u$ values with its standard version in CG-noise and α-noise scenarios. The DCD parameters are $H = 4$ and $M_0 = 16$. It is seen that, the proposed DCD-R-dRLS algorithm is also robust to impulsive noises, and approaches the R-dRLS performance with increase in $N_u$. In this example, the DCD-R-dRLS algorithm with $N_u = 4$ ($< M$) has a good approximation to the R-dRLS algorithm, while the complexity of the former is significantly lower than that of the latter. Moreover, many simulations have been carried out in different impulsive noise scenarios by prolonging the iteration $i$ to a larger number than the one in Fig. 12, e.g., $5 \times 10^5$, using MATLAB R2013A on a Intel(R) Core(TM) i5-4590 CPU @ 3.30 GHz processor. We did not observe any numerical instability during the simulations for both proposed R-dRLS and DCD-dRLS algorithms.

B. Application: Distributed Spectrum Estimation

We have also tested the proposed algorithms’ performance in an application of distributed spectrum estimation in CR, in which the objective is to estimate the spectrum of a transmitted signal source $s$ in the network with $N$ nodes [7], [8], [57]. We use $\phi_s(f) = \sum_{m=1}^{M} q_m(f) w_{m} = q^T(f) w^o$ to denote the power spectral density (PSD) of the signal $s$ at frequency $f$, where $q(f) = [q_1(f), ..., q_M(f)]^T$ is a vector consisting of basis functions evaluated at normalized frequency $f$, and $w^o = [w_1^o, ..., w_M^o]^T$ stands for the power that transmits the signal $s$ over each of $M$ basis functions and needs to be estimated. Such basis expansion can accurately model the spectrum of the signal $s$ for large enough $M$. Considering $\mathcal{H}_{k}(f, i)$ is the transfer function of the channel between the station emitting the signal $s$ and receiver node $k$ at time instant $i$, the PSD of the received signal at node $k$ can be expressed as

$$\phi_{k, r}(f) = |\mathcal{H}_{k}(f, i)|^2 \phi_s(f) + \sigma_{r, k}^2 = q^T_{k,i}(f) w^o + \sigma_{r, k}^2,$$  \hspace{1cm} (58)$$

where $q_{k,i}(f) = |\mathcal{H}_{k}(f, i)| q(f)$, and $\sigma_{r, k}^2$ is the received noise power at node $k$.

At time instant $i$, each node $k$ observes the received PSD expressed in (58) over $N_c$ frequency samples $f_i = f_{\text{min}} : (f_{\text{max}} - f_{\text{min}})/N_c$ for $i = 1, ..., N_c$; accordingly, the output measurements of node $k$ obey the following relation:

$$d^k(i) = q^T_{k,i}(f_i) w^o + \sigma_{r,k}^2 + v^k(i),$$ \hspace{1cm} (59)$$

where $v^k(i)$ denotes the observation noise at frequency $f_i$. The noise power $\sigma_{r,k}^2$ can be estimated with high accuracy before the spectrum estimation, using, for example, an energy estimator over an idle band, and then subtracted from (59) [7], [8], [57]. Then, by collecting the output measurements over $N_c$ frequencies, we obtain a data model at every node $k$ for distributed spectrum estimation:

$$d^k(i) = Q_{k,i} w^o + v^k(i),$$ \hspace{1cm} (60)$$

where $Q_{k,i} = [q_{k,i}(f_1), ..., q_{k,i}(f_{N_c})]^T$, $d^k(i) = [d^k_1(i), ..., d^k_{N_c}(i)]^T$, and $v^k(i) = [v^1_k(i), ..., v^{N_c}_k(i)]^T$. 

---

Footnote: 

3. Here the curves of both the R-dRLS and DCD-R-dRLS algorithms are omitted due to their divergence performance in impulsive noise.
Based on this model, we estimate the unknown spectrum $\mathbf{w}_s$ of the signal $s$ using different diffusion algorithms over the network given in Fig. 3a). In the simulation [8], [57], we use $M = 50$ nonoverlapping rectangular basis functions with amplitude equal to one to model the PSD of the signal $s$. The nodes scan $N_c = 100$ frequencies over the normalized frequency axis between 0 and 1. We assume that $\mathbf{w}_s$ has only 8 non-zero elements, meaning that the unknown spectrum is transmitted over 8 basis functions, and the power transmitted over each basis function is set to 0.7. The observation noise $v^s_k(i)$ is an $\alpha$-stable process as in the previous Example 3 [24].

In Fig. 14 we compare the network MSD performance of different algorithms considered for the distributed spectrum estimation. As depicted, the dRLS algorithm can not identify the spectrum coefficients $\mathbf{w}_s$ due to its divergence in an $\alpha$-stable noise environment. In comparison with the dSE-LMS, dLMP, RVWC-dRLS and dRLP algorithms, the proposed R-dRLS and DCD-R-dRLS (with $N_c = 4$) algorithms still obtain better estimation performance. We also notice from this figure that the DCD-R-dRLS algorithm with lower computational complexity approaches the R-dRLS performance. In Fig. 15 we also select the robust dRLS-type algorithms to show their performance in terms of PSD at node 1. From the results, the proposed R-dRLS and DCD-R-dRLS algorithms have lower side lobes in the PSD curves than those of the other two algorithms, thus fitting much better the true spectrum.

**Appendix A**

**Verification of (25)**

From Fig. 16 one can see that the left side of (25) has a good agreement with the right side of that. This reveals that the simplification from (24) to (25) is reasonable.

**Appendix B**

**Convergence of $E[\xi_k(i)]$ to 0**

It is evident from (14) that $\xi_k(i)$ as a function of $i$ is non-increasing in adaptation process, with positive values. So, $\xi_k(i)$ is non-increasing in adaptation process, with positive values. So,
the limit of $E\{\xi_k(i)\}$ at $i \to \infty$ is existent. Applying the expectation operator to (14), we obtain

$$
E\{\xi_k(i)\} = \beta E\{\xi_k(i-1)\} + (1 - \beta) E\{\min\{\|g_{k,i}\|_2^2 \xi_k^2(i), \xi_k(i-1)\}\},
$$
(B.1)

$$
E\{\xi_k(i)\} = \sum_{m \in \mathbb{N}_k} c_{m,k} E\{\zeta_m(i)\},
$$
(B.2)

Again using the assumption that the variance of $\xi_k(i)$ is small enough since $\beta$ closes to 1, we are able to make the approximation,

$$
E\{\min\{\|g_{k,i}\|_2^2 \xi_k^2(i), \xi_k(i-1)\}\} \approx \int_0^{m_k} m_k dF_{k,i}(m_k)
$$

$$
+ E\{\xi_k(i-1)\} P_{k,i}[m_k > E\{\xi_k(i-1)\}]
$$

(B.3)

where $m_k \triangleq \|g_{k,i}\|_2^2 \xi_k^2(i)$ means that both $m_k$ and $\|g_{k,i}\|_2^2 \xi_k^2(i)$ have the same distribution, $P_{k,i}[\cdot]$ denotes the probability of event in the argument, and $F_{k,i}(m_k)$ denotes the distribution function of $m_k$ at time instant $i$.

Let us define the network global vectors as follows:

$$
\xi(i) \triangleq \text{col}\{\xi_1(i), ..., \xi_N(i)\},
$$

$$
\zeta(i) \triangleq \text{col}\{\zeta_1(i), ..., \zeta_N(i)\}.
$$

Therefore, according to (B.3) and (B.4), we reformulate (B.1) and (B.2) for all nodes as:

$$
E\{\xi(i)\} = \mathbf{C}^T \beta E\{\xi(i-1)\} + (1 - \beta) (\mathbf{H}_i E\{\xi(i-1)\} + \mathbf{m}(i))
$$

(B.5)

where

$$
\mathbf{H}_i = \text{diag}\{P_{1,i}[m_1 > E\{\xi_1(i-1)\}], ..., P_{N,i}[m_N > E\{\xi_N(i-1)\}]\},
$$

and

$$
\mathbf{m}_i = \text{col}\left\{ \int_0^{m_1} m_1 dF_{1,i}(m_1), ..., \int_0^{m_N} m_N dF_{N,i}(m_N) \right\}.
$$

(B.6)

Taking the $\infty$-norm of both sides of (B.5) and recalling $\|\mathbf{C}^T\|_\infty = 1$, it is found the following inequality:

$$
\|E\{\xi(i)\}\|_\infty \leq \beta \|E\{\xi(i-1)\}\|_\infty + (1 - \beta) \|\mathbf{H}_i E\{\xi(i-1)\} + \mathbf{m}(i)\|_\infty.
$$

(B.8)

Based on the diagonal definition in (B.6), we deduce an equivalent form from (B.8), i.e., for $k = 1, ..., N$,

$$
E\{\xi_k(i)\} \leq \beta E\{\xi_k(i-1)\} + (1 - \beta) E\{\xi_k(i-1)\} P_{k,i}[m_k > E\{\xi_k(i-1)\}] + \int_0^{m_k} m_k dF_{k,i}(m_k).
$$

(B.9)

It is supposed that there is a limit for $F_{k,i}(m_k)$ when $i \to \infty$, (B.9) further reduces to

$$
E\{\xi_k(\infty)\} \cdot P_{k,\infty}[m_k \leq E\{\xi_k(\infty)\}] \leq \int_0^{E\{\xi_k(\infty)\}} m_k dF_{k,\infty}(m_k).
$$

(B.10)

In (B.10), the relation $E\{\xi_k(i)\} = E\{\xi_k(i-1)\}$ as $i \to \infty$ is also used. Herein, we consider the equality case in (B.10), i.e.,

$$
E\{\xi_k(\infty)\} \cdot P_{k,\infty}[m_k \leq E\{\xi_k(\infty)\}] = \int_0^{E\{\xi_k(\infty)\}} m_k dF_{k,\infty}(m_k).
$$

(B.11)

It is shown in Appendix A in [21], for a similar equation (B.11), its solution is $E\{\xi_k(\infty)\} = 0$. Since (B.11) is an upper bound of (B.10), we can conclude that $E\{\xi_k(i)\}$ given by (14) would also converge to zero. Moreover, as its intermediate quantity, $E\{\xi_k(i)\}$ also converges to zero.

REFERENCES


Yi Yu (S’17-M’18) received the B.E. degree at XiHua University, Chengdu, China, in 2011, and the M.S. degree and Ph.D degree at Southwest Jiaotong University, Chengdu, China, in 2014 and 2018, respectively. From Dec. 2016 to Dec. 2017, he was a visiting Ph.D student with the Department of Electronic Engineering, University of York, United Kingdom. He is currently an Associate Professor with the School of Information Engineering, Southwest University of Science and Technology, Mianyang, China. His research interests include adaptive signal processing, distributed estimation, and compressive sensing.

Haiquan Zhao was born in Henan Province, China, in 1974. He received the B.S. degree in applied mathematics, and the M.S. and Ph.D degrees in signal and information processing from Southwest Jiaotong University, Chengdu, China, in 1998, 2005, and 2011, respectively. Since 2012, he was a Professor with the School of Electrical Engineering, Southwest Jiaotong University. His current research interests include adaptive filtering algorithm, nonlinear active noise control and nonlinear system identification.

Rodrigo C. de Lamare was born in Rio de Janeiro, Brazil, in 1975. He received his Diploma in electronic engineering from UFRJ in 1998 and the MSc and PhD degrees in electrical engineering from the Pontifical Catholic University of Rio de Janeiro (PUC-RIO) in 2001 and 2004, respectively. He then worked as a Postdoctoral Fellow from January to June 2005 at the Centre for Telecommunications Studies (CETUC), PUC-RIO and from July 2005 to January 2006 at the Signal Processing Laboratory, UFRJ. Since January 2006, he has been with the Communications Group, Department of Electronics, University of York, United Kingdom, where he is a Professor. Since April 2013, he has also been a Professor at PUC-RIO. Dr de Lamare has participated in numerous projects funded by government agencies and industrial companies. He received a number of awards for his research work and has served as the general chair of the IEEE 7th International Symposium on Wireless Communication Systems (ISWCS) 2010, held in York, UK in September 2010, as the technical programme chair of ISWCS 2013 and WSA 2015 in Ilmenau, Germany, and as the general chair of the 9th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM) in Rio de Janeiro, Brazil, in July 2016. Dr de Lamare is a senior member of the IEEE and an elected member of the IEEE Signal Processing Theory and Method technical committee. He currently serves as editor for IEEE Transactions on Communications, IEEE Wireless Communications Letters, and as a senior area editor for the IEEE Signal Processing Letters. His research interests lie in communications and signal processing, areas in which he has published nearly 400 papers in international journals and conferences.

Yuriy Zakharov (M’01-SM’08) received the M.Sc. and Ph.D. degrees in Electrical Engineering from the Moscow Power Engineering Institute, Moscow, Russia, in 1977 and 1983, respectively. From 1977 to 1983, he was with the Special Design Agency, Moscow Power Engineering Institute. From 1983 to 1999, he was with N. N. Andreev Acoustics Institute, Moscow, Russia. From 1994 to 1999, he was a DSP Group Leader with Nortel. Since 1999, he has been with the Communication Technologies Research Group, University of York, York, U.K., where he is currently a Reader with the Department of Electronic Engineering. His research interests include signal processing, communications, and acoustics.

Lu Lu was born in Chengdu, China, in 1990. He received the Ph.D. degree in the field of signal and information processing at the School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China, in 2018. From 2017 to 2018, he was a visiting Ph.D student with the Electrical and Computer Engineering at McGill University, Montreal, QC, Canada. He is currently a Postdoctoral Fellow with the College of Electronics and Information Engineering, Sichuan University, Chengdu, China. His research interests include adaptive filtering, kernel methods and distributed estimation.