Distributed Quantization-Aware RLS Learning with Bias Compensation and Coarsely Quantized Signals

Alireza Danaee, Rodrigo C. de Lamare and Vitor H. Nascimento

Abstract

In this work, we present an energy-efficient distributed learning framework using coarsely quantized signals for Internet of Things (IoT) networks. In particular, we develop a distributed quantization-aware recursive least-squares (DQA-RLS) algorithm that can learn parameters in an energy-efficient fashion using signals quantized with few bits while requiring a low computational cost. Moreover, we develop a bias compensation strategy to further improve the performance of the proposed DQA-RLS algorithm. We carry out a statistical analysis of the proposed DQA-RLS algorithm and derive analytical expressions for predicting the mean-square deviation. A computational complexity evaluation and a study of the power consumption of the proposed and existing techniques are then presented. Numerical results assess the DQA-RLS algorithm against existing techniques for a distributed parameter estimation task in a scenario where IoT devices operate in peer-to-peer mode.

Index Terms

distributed learning, energy-efficient signal processing, adaptive algorithms, coarse quantization

I. Introduction

DISTRIBUTED signal processing algorithms are of great relevance for statistical inference in wireless networks and applications such as wireless sensor networks (WSNs) [1], [2], the Internet of Things (IoT) [3], [4], distributed optimization [5]–[7] and smart grid implementations [8], [9]. In fact, distributed signal processing techniques deal with the extraction of information from data collected at nodes that are distributed over a geographical area. In this context, for each node a set of neighbor nodes collects and processes their local information, and then transmits their estimates to a specific node. Upon reception of the possibly noisy estimates, each specific node combines the collected information together with its local estimate to generate improved estimates.

A. Prior and Related Work

Prior work on distributed signal processing techniques has studied protocols for exchanging information [10]–[12], adaptive learning algorithms [13], the exploitation of sparse measurements [14], [15], topology adaptation [16], [17], and robust techniques against interference and noise [18], [19]. Even though there have been many studies that evaluated the need for data exchange and signaling among nodes as well as their computational complexity, prior work on energy-efficient techniques is rather limited.

In this context, energy-efficient signal processing techniques have gained a great deal of interest in the last decade or so due to their ability to save energy and promote sustainable development of electronic systems and devices. Electronic devices often exhibit a power consumption that is strongly dependent on the analog-to-digital converters (ADCs) and the number of bits used to represent digital samples [20]. This is of central importance to devices that are battery-operated and to wireless networks that must keep the power consumption to a low level for sustainability reasons. In particular, prior work on energy efficiency has reported many contributions in signal processing for communications and electronic systems that operate with coarsely quantized signals [21]–[24].

Among the methods to reduce the energy consumption of networks are: i) compression of the communication data between neighbor nodes and ii) coarse quantization with ADCs of signals measured by sensors. Communication-efficiency techniques enable IoT devices to reduce their energy consumption with data transmission and reduce the communication bandwidth, and have been reported in adaptive networks [25]–[27]. On the other hand, IoT devices contain many sensors that allow them to interact with the physical world, collecting and processing streaming data in real time. They integrate various sensors such as temperature, humidity, accelerometer, gyroscope, magnetometer, altimeter, heart rate, light, microphone, camera, battery monitor, infrared proximity, gas, ultraviolet and capacitive sensors. The total energy-consumption and cost of these sensors affect the energy-consumption and the cost of IoT devices [28]. The type of sensor determines the accuracy of the analog interface and the resolution of ADCs. The ADC resolution requirement varies greatly with the sensing application, ranging from 6 to 16 bits (see [28] Table 1). This emphasizes the importance of energy-efficient techniques that deal with the coarse quantization of measurement data to enable IoT devices to work with the low energy-consumption sensors. To this end, a distributed quantization-aware least-mean square (DQA-LMS) algorithm was proposed in [29], [30] to reduce the power consumption of ADCs in adaptive IoT networks in an energy-efficient framework.

B. Contributions

In this work, we present an energy-efficient distributed learning framework using coarsely quantized signals for Internet of Things (IoT) networks. In particular, we develop a distributed quantization-aware recursive least-squares (DQA-RLS) algorithm that can learn parameters in an energy-efficient fashion using signals quantized with few bits while requiring a low computational cost. Moreover, we develop a closed-form bias compensation strategy to further improve the performance of the proposed DQA-RLS algorithm. We then carry out a statistical analysis of the proposed DQA-RLS algorithm along with a computational complexity and a power consumption evaluation of the proposed and existing techniques. Numerical results assess the DQA-RLS algorithm against existing techniques for a distributed parameter estimation task in a scenario where IoT devices operate in peer-to-peer mode.

The DQA-LMS has the simplicity and properties of LMS-type algorithms whereas the DQA-RLS algorithm has the fast convergence property of RLS-type algorithms [31] even with high correlated inputs. Moreover, one can easily extend the Quantization-Aware framework to other distributed adaptive algorithms inspired by the results obtained from DQA-LMS and DQA-RLS algorithms. For instance, the dichotomous coordinate descent (DCD) algorithm [32], [33] and the partial-diffusion recursive least-squares (PDRLS) algorithm [25] have been successfully used for significant reduction in the complexity and communication cost of RLS algorithms, respectively, and can be incorporated in the quantization-aware framework presented here to reduce the complexity, the communication cost and the power consumption together.

The main contributions of this work can be summarized as:

- The proposed DQA-RLS algorithm and a bias compensation strategy for processing signals quantized with few bits.
- An analysis of the DQA-RLS algorithm in terms of mean and mean-square performance, computational complexity and power consumption is carried out.
- A simulation study of DQA-RLS, the bias compensation strategies and existing techniques along with a validation of the theoretical expressions obtained in the analysis.

This paper is structured as follows: Section II introduces the signal model and states the problem. Section III details the derivation of the proposed DQA-RLS algorithm. Section IV is devoted to the statistical analysis of the proposed DQA-RLS algorithm, the bias compensation strategy along with their computational complexity and power consumption. Section V shows and discusses the results of simulations, whereas Section VI draws the conclusions of this work. Throughout this paper, we show the scalars, vectors and matrices with lowercase, boldface lowercase and boldface uppercase letters, respectively, and adopt the notation in Table I.

TABLE I
NOTATION

$(.)^T$	Transposition
$(.)^*$	Complex Conjugate (Hermitian) transpose
$\|.\|$	Euclidean norm of a vector
$\|\mathbf{x}\|^2_{\mathbf{Z}} = \mathbf{x}^*\mathbf{Z}\mathbf{x}$	$\mathbf{Z}$-Weighted Euclidean norm of a column vector $\mathbf{x}$
$\|\mathbf{y}\|^2_{\mathbf{Z}} = \mathbf{y}\mathbf{Z}\mathbf{y}^*$	$\mathbf{Z}$-Weighted Euclidean norm of a row vector $\mathbf{y}$
Tr(.)	Trace of a matrix
$\otimes$	Kronecker product of two matrices
vec(Z)	Vectorization operation of $\mathbf{Z}$
col$\{...\}$	Stacks its arguments column-wise
diag$(\mathbf{Z})$	Creates a vector with the diagonal entries of $\mathbf{Z}$
diag$\{...\}$	Creates a diagonal matrix with the entries of a vector or with the diagonal entries of a matrix

II. Signal Model and Problem Statement

In this section, we detail the signal model of the proposed adaptive distributed network with low-resolution ADCs. Furthermore, we also state the problem and review the signal decomposition of coarsely quantized signals performed by Bussgang's theorem, which is central to signal processing with low-resolution ADCs.

A. Adaptive Distributed Network

We consider an IoT network consisting of N nodes or agents which run distributed signal processing techniques to perform the desired tasks, as depicted in Fig. 1. The presented model considers a desired signal $d_k(i)$ at time instant i described by

$$d_k(i) = \mathbf{w}_o^*\mathbf{x}_k(i) + v_k(i), \quad k = 1, 2, \ldots, N, \tag{1}$$

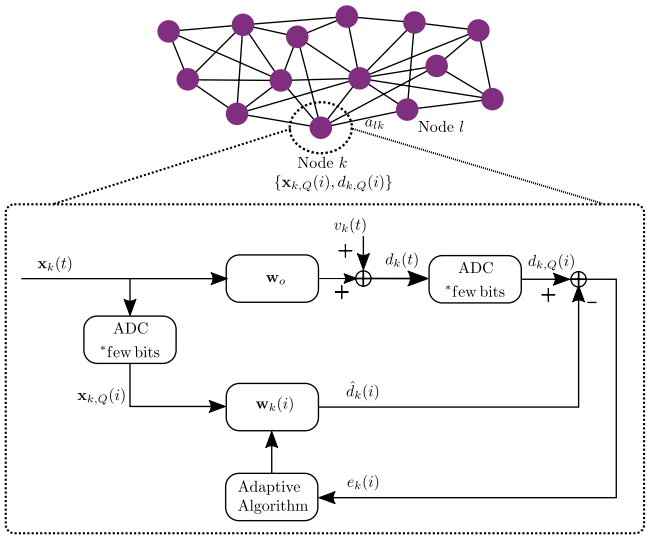


Fig. 1. A distributed adaptive IoT network

where $\mathbf{w}_o \in \mathbb{R}^{M\times 1}$ is the parameter vector that the agents must estimate, $\mathbf{x}_k(i) \in \mathbb{R}^{M\times 1}$ is the regressor at node k, and $v_k(i)$ represents Gaussian noise with zero mean and variance $\sigma^2_{v,k}$ at each node k. We also adopt the Adapt-then-Combine (ATC) diffusion rule because it has been shown a more effective scheme than other previously reported schemes such as incremental and consensus protocols [10], [11].

As shown in Fig. 1, because the measurement data at each node and the unknown system are analog and each agent processes the local data $\{d_k(i), \mathbf{x}_k(i)\}$ digitally, we need two ADCs in each agent. Specifically, a digital signal is acquired from the observation of an analog signal and the use of an ADC with b bits, which employs a scalar quantizer with the set of thresholds $\mathcal{T}_b = \{\tau_1, \tau_2, \ldots, \tau_{2^b-1}\}$ and the set of labels $\mathcal{L}_b = \{l_0, l_1, \ldots, l_{2^b-1}\}$. We summarize the key features of the proposed adaptive distributed network of Fig. 1 as follows:

- The proposed adaptive network consists of N nodes which employ distributed signal processing techniques to perform a desired task.
- Each node has two ADCs to quantize the received signals, i.e., the input regressors and the desired signals, and process them digitally. Moreover, in the proposed bias compensation section, we show that if each node uses one ADC to quantize both the input regressors and the desired signals can reduce the complexity of the proposed adaptive network. Note that for complex data, one ADC must be used for the real part and one ADC for the imaginary part. However, for simplicity, we carry out an analysis of the proposed DQA-RLS algorithm for real data.
- We assume that the analog signals arriving at the ADCs have been adjusted (e.g., by using automatic gain control (AGC)) to have approximately unit power, and that the fixed thresholds and labels of ADCs are calculated with the Lloyd-Max algorithm for a Gaussian signal with zero-mean and unit variance. We show in the numerical results that even imperfect gain control at the analog inputs will not degrade the performance of the proposed algorithms that are based on fixed thresholds and labels.
- Nodes do not have access to the variance of the quantization distortion. Therefore, we estimate the quantization distortion variance alongside the designed thresholds and labels to use this estimation later in the proposed algorithms.

One concern is that as the number of agents increases, the energy consumption might grow substantially when using high-resolution ADCs for each agent. This motivates us to quantize signals using few bits. Therefore, the problem we are interested in solving in this work is how to design energy-efficient distributed learning algorithms that can cost-effectively operate with coarsely quantized signals.

B. Signal Decomposition with Coarse Quantization

In order to provide a clear exposition, we provide a short overview of Bussgang's theorem, which allows one to deal with nonlinearities like the distortion generated by ADCs.

Theorem 1: Given two Gaussian signals, the cross-correlation function taken after a signal has undergone nonlinear amplitude distortion is identical (except for a scaling factor) to the cross-correlation function taken before the distortion [34], [35]. Specifically, according to Bussgang's theorem, for a pair of zero-mean jointly complex Gaussian random variables $x(i) \sim \mathcal{CN}(0, \sigma^2_{x(i)})$ and $x(n) \sim \mathcal{CN}(0, \sigma^2_{x(n)})$, and for the output $x_Q(i)$ of some scalar-valued nonlinear function $x_Q(i) = f\big(x(i)\big)$, where $f(\cdot) : \mathcal{C} \to \mathcal{C}$, it holds that

$$\mathbb{E}_{x_Q(n),x(i)}\big[x_Q(n)x^*(i)\big] = g(n)\mathbb{E}_{x(n),x(i)}\big[x(n)x^*(i)\big] \tag{2}$$

in which

$$g(n) = \frac{1}{\sigma_{x(n)}^2}\mathbb{E}_{x(n)}\big[f\big(x(n)\big)x^*(n)\big]. \tag{3}$$

Let us consider now that the previously mentioned nonlinear function $f(\cdot)$ can be applied element-wise to a zero-mean complex Gaussian random vector $\mathbf{x}(i) = \big[x(1), x(2), \ldots, x(M)\big] \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_x)$ where $\mathbf{R}_x \in \mathbb{C}^{M\times M}$ is the covariance matrix of $\mathbf{x}$, resulting in a vector $\mathbf{x}_Q$, i.e.,

$$\mathbf{x}_Q = f\big(\mathbf{x}\big). \tag{4}$$

It follows from (2) that

$$\mathbf{R}_{x_Q x} = \mathbf{G}\mathbf{R}_x \tag{5}$$

where

$$\mathbf{G} = \operatorname{diag}\big\{g_1, g_2, \ldots, g_M\big\} \tag{6}$$

represents a diagonal $M \times M$ matrix whose mth diagonal entry is computed as in (3). Bussgang's theorem can be used to decompose the output of a nonlinear device as a linear function of the input $\mathbf{x}$ plus a distortion $\mathbf{q} \in \mathbb{C}^M$ that is uncorrelated (but not independent) with the input as [34]

$$\mathbf{x}_Q = \mathbf{G}\mathbf{x} + \mathbf{q} \tag{7}$$

The referred uncorrelation can be viewed as follows

$$\begin{aligned}\mathbb{E}\big[\mathbf{q}\mathbf{x}^*\big] &= \mathbb{E}\big[\big(\mathbf{x}_Q - \mathbf{G}\mathbf{x}\big)\mathbf{x}^*\big] \\ &= \mathbf{R}_{x_Q x} - \mathbf{G}\mathbf{R}_x = \mathbf{0}_{M\times M}\end{aligned} \tag{8}$$

where we made use of (5).

To use this result in the distributed adaptive network in Fig. 1, let $\mathbf{x}_{k,Q} = Q_b(\mathbf{x}_k)$ denote the b-bit quantized output of an ADC at node k, described by a set of $2^b + 1$ thresholds $\mathcal{T}_b = \{\tau_0, \tau_1, ..., \tau_{2^b}\}$, such that $-\infty = \tau_0 < \tau_1 < ... < \tau_{2^b} = \infty$, and the set of 2^b labels $\mathcal{L}_b = \{l_0, l_1, ..., l_{2^b-1}\}$ where $l_p \in (\tau_p, \tau_{p+1}]$, for $p \in [0, 2^b - 1]$ [21]. Let us assume that $\mathbf{x}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{x_k})$ where $\mathbf{R}_{x_k} \in \mathbb{C}^{M\times M}$ is the covariance matrix of $\mathbf{x}_k$. We have thus a relation between $\mathbf{x}_{k,Q}$ and $\mathbf{x}_k$ in the form of (4), with the quantization function $Q_b(\cdot)$ playing the part of the scalar-valued nonlinear function $f(\cdot)$ in (4). We now use (7) to derive a model for the quantized vector $\mathbf{x}_{k,Q}$, which we later use to derive our DQA-RLS algorithm. Employing Bussgang's theorem, $\mathbf{x}_{k,Q}$ can be decomposed as

$$\mathbf{x}_{k,Q} = \mathbf{G}_{k,b}\mathbf{x}_k + \mathbf{q}_{x,k}, \tag{9}$$

where the distortion $\mathbf{q}_{x,k}$ is uncorrelated with $\mathbf{x}_k$, and $\mathbf{G}_{k,b} \in \mathbb{R}^{M\times M}$ is a diagonal matrix given by [21]

$$\begin{aligned}\mathbf{G}_{k,b} = \operatorname{diag}\big\{\mathbf{R}_{x_k}\big\}^{-\frac{1}{2}} \sum_{j=0}^{2^b-1} \frac{l_j}{\sqrt{\pi}} \Bigg[&\exp\big(-\tau_j^2 \operatorname{diag}\big\{\mathbf{R}_{x_k}\big\}^{-1}\big) \\ &- \exp\big(-\tau_{j+1}^2 \operatorname{diag}\big\{\mathbf{R}_{x_k}\big\}^{-1}\big)\Bigg],\end{aligned} \tag{10}$$

where the index b indicates the number of bits (ADC resolution). Since $\mathbf{G}_{k,b}$ is a real-valued diagonal matrix, $\mathbf{G}_{k,b}^H = \mathbf{G}_{k,b}$. Note that, as a simplifying approximation, we also apply this signal decomposition to the desired signal, $d_{k,Q}$, which is the output of the second ADC in the system, and for the particular case that $\mathbf{R}_{x_k} = \mathbb{E}[\mathbf{x}_k\mathbf{x}_k^*] = \sigma_{x,k}^2\mathbf{I}_M$, the matrix $\mathbf{G}_{k,b}$ becomes $g_{k,b}\mathbf{I}_M$ and $g_{k,b} \in \mathbb{R}$ is given by

$$g_{k,b} = \frac{1}{\sqrt{\sigma_{x_k}^2}} \sum_{j=0}^{2^b-1} \frac{l_j}{\sqrt{\pi}} \big(e^{-\frac{\tau_j^2}{\sigma_{x_k}^2}} - e^{-\frac{\tau_{j+1}^2}{\sigma_{x_k}^2}}\big). \tag{11}$$

To compute $\mathbf{G}_{k,b}$, three parameters are needed, namely the set of thresholds $\mathcal{T}_b$, the set of labels $\mathcal{L}_b$ and the covariance matrix of the input signal $\mathbf{R}_{x_k}$. The thresholds and labels are designed for the ADCs and are available at the nodes. However, the nodes have access to $\mathbf{R}_{x_k,Q}$ instead of $\mathbf{R}_{x_k}$. To overcome this, we can estimate the variance of the distortion, i.e., $\widehat{\sigma}_{q,k}^2$ (as this variance is not accessible in practice) and the covariance matrix of the input signal, $\widehat{\mathbf{R}}_{x_k} = \mathbf{R}_{x_k,Q} + \widehat{\sigma}_{q,k}^2\mathbf{I}_M$. In the next section, we show how to design the ADCs and estimate the variance of the distortion.

C. Design of ADCs

In this section, we show how to compute the thresholds and labels to design the ADCs. To minimize the mean square error (MSE) between $\mathbf{x}_k$ and $\mathbf{x}_{k,Q}$, we need to characterize the probability density function (PDF) of $\mathbf{x}_k$ to find the optimal quantization labels. Because choosing these labels based on such PDF is ineffective in practice (since the PDFs are difficult to estimate), we assume the regressor $\mathbf{x}_k(i)$ is Gaussian, then adapt the approach in [21] and compute the thresholds and labels as follows:

1) We generate an auxiliary Gaussian random variable x_{aux} with unit variance and then use the Lloyd-Max algorithm [36], [37] to find a set of thresholds $\widetilde{\mathcal{T}}_b = \{\tau_1, \ldots, \tau_{2^b-1}\}$ and labels $\mathcal{L}_b = \{l_0, \ldots, l_{2^b-1}\}$ that minimize the MSE between the unquantized and the quantized signals.
2) We wrap up the set of thresholds $\mathcal{T}_b$ by adding $\tau_0 = -\infty$ and $\tau_{2^b} = \infty$ to $\widetilde{\mathcal{T}}_b$.
3) We quantize x_{aux} using $\widetilde{\mathcal{T}}_b$ and $\mathcal{L}_b$, generate the quantized signal $x_{\text{aux},Q}$, and estimate the variance of the distortion, $\sigma^2_{q,k}$ with the subtraction of the variance of the quantized auxiliary signal from the variance of the auxiliary signal as follows

$$\widehat{\sigma}^2_{q,k} = \sigma^2_{x_{\text{aux}}} - \sigma^2_{x_{\text{aux},Q}}. \tag{12}$$

4) We estimate the covariance matrix of the distortion as follows

$$\widehat{\mathbf{R}}_{q_x,k} = \widehat{\sigma}^2_{q,k}\mathbf{I}_M. \tag{13}$$

Note that step 1 designs the ADC thresholds and labels, step 2 completes the thresholds needed for (10), and steps 3 and 4 are useful to estimate $\mathbf{R}_{x_k}$.

III. Proposed DQA-RLS Algorithm

In this section, we present the derivation of the proposed DQA-RLS algorithm. We consider $\mathbf{x}_k(t)$ and $d_k(t)$ as the (band-limited before sampling) analog input and output of the unknown system $\mathbf{w}_o$ at node k. Let $\mathbf{x}_k(i)$ and $d_k(i)$ denote the digital versions of $\mathbf{x}_k(t)$ and $d_k(t)$, and $\mathbf{x}_{k,Q}(i)$ and $d_{k,Q}(i)$ denote the coarsely quantized versions of $\mathbf{x}_k(i)$ and $d_k(i)$, respectively. We assume that the input signal at each node is Gaussian with zero mean and covariance matrix $\mathbf{R}_{x_k}$ for $k = 1, 2, ..., N$. We can now decompose $\mathbf{x}_{k,Q}(i)$ and $d_{k,Q}(i)$ as

$$\mathbf{x}_{k,Q}(i) = \mathbf{G}_{k,b}\mathbf{x}_k(i) + \mathbf{q}_{x,k}(i), \tag{14}$$

and

$$\begin{aligned} d_{k,Q}(i) &= Q(d_k(i)) = g_{k,b}d_k(i) + q_{d,k}(i) \\ &= g_{k,b}\mathbf{w}_o^*\mathbf{x}_k(i) + \bar{q}_k(i), \end{aligned} \tag{15}$$

where $\bar{q}_k(i) = g_{k,b}v_k(i) + q_{d,k}(i)$ is uncorrelated with $\mathbf{x}_k(i)$. Note that $\mathbf{G}_{k,b}$ and $g_{k,b}$ are built based on $\mathbf{x}_k(i)$ and $d_k(i)$, respectively.

We show next that a learning algorithm based directly on (15) is biased for estimating $\mathbf{w}_o$ and show how to correct for this bias. To this end, let $\boldsymbol{\beta}_{k,b}(i)$ be an $M \times M$ bias-compensation matrix to be computed and define $\widehat{d}_k(i) = \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)$. Thus, the error is given by

$$\begin{aligned} \bar{e}_k(i) &= d_{k,Q}(i) - \widehat{d}_k(i) \\ &= d_{k,Q}(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i). \end{aligned} \tag{16}$$

As seen in (16), $\bar{e}_k(i)$ is different from $e_k(i) = d_k(i) - \mathbf{w}_k^*(i-1)\mathbf{x}_k(i)$ in the diffusion RLS (DRLS) [12].

Let us consider a network of N nodes distributed over an area as in Fig. 1. At time i, we globally collect the quantized input regressors into a matrix $\mathbf{X}_{i,Q}$, the quantized desired signal into vector $\mathbf{d}_{i,Q}$, the noise samples into vector $\mathbf{v}_i$, and the bias-compensation coefficients into the matrix $\boldsymbol{\beta}_{i,b}$ over all nodes as follows:

$$\begin{aligned} \mathbf{X}_{i,Q} &= \text{bdiag}\{\mathbf{x}_{1,Q}(i), \ldots, \mathbf{x}_{N,Q}(i)\} && (MN \times N) \\ \mathbf{d}_{i,Q} &= \text{col}\{d_{1,Q}(i), \ldots, d_{N,Q}(i)\} && (N \times 1) \\ \mathbf{v}_i &= \text{col}\{v_1(i), \ldots, v_N(i)\} && (N \times 1) \\ \boldsymbol{\beta}_{i,b} &= [\boldsymbol{\beta}_{1,b}(i), \ldots, \boldsymbol{\beta}_{N,b}(i)] && (M \times MN). \end{aligned} \tag{17}$$

We can write down the covariance matrix of the noise vector as follows

$$\mathbf{R}_v = \mathbb{E}[\mathbf{v}_i\mathbf{v}_i^*] = \text{diag}\{\sigma^2_{v_1}, \ldots, \sigma^2_{v_N}\} \qquad (N \times N). \tag{18}$$

Now we collect these data from time 0 to time i as follows

$$\begin{aligned}
\boldsymbol{X}_{i,Q} &= \mathrm{bdiag}\{\mathbf{X}_{i,Q},\ldots,\mathbf{X}_{0,Q}\} && (MN(i+1)\times N(i+1)),\\
\boldsymbol{d}_{i,Q} &= \mathrm{col}\{\mathbf{d}_{i,Q},\ldots,\mathbf{d}_{0,Q}\}^T && (1\times N(i+1)),\\
\boldsymbol{v}_{i} &= \mathrm{col}\{\mathbf{v}_{i},\ldots,\mathbf{v}_{0}\}^T && (1\times N(i+1)),\\
\boldsymbol{B}_{i,b} &= [\boldsymbol{\beta}_{i,b},\ldots,\boldsymbol{\beta}_{0,b}] && (M\times MN(i+1)),
\end{aligned} \tag{19}$$

where $\boldsymbol{R}_{v,i}=\mathbb{E}[\boldsymbol{v}_i^*\boldsymbol{v}_i]$. Note that for the globally collected quantities we denote time by a subscript, whereas for node-wise quantities we denote time by parenthesis. Then, we estimate the $M\times 1$ vector $\mathbf{w}_o$ by solving the weighted regularized least-squares problem given by

$$\min_{\mathbf{w}}\Big[\big\|\mathbf{w}-\overline{\mathbf{w}}\big\|^2_{\boldsymbol{\Pi}_i}+\big\|\boldsymbol{d}_{i,Q}-\mathbf{w}^*\boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\big\|^2_{\boldsymbol{\Sigma}_i}\Big], \tag{20}$$

where $\overline{\mathbf{w}}$ is a given column vector, usually $\overline{\mathbf{w}}=\mathbf{0}$, $\boldsymbol{\Pi}_i>0$ is an $M\times M$ positive-definite matrix that incorporates a regularization term $\|\mathbf{w}-\overline{\mathbf{w}}\|^2_{\boldsymbol{\Pi}_i}$ into the least-squares problem, and $\boldsymbol{\Sigma}_i>0$ is an $N(i+1)\times N(i+1)$ Hermitian positive-definite matrix that incorporates weighting into the least-squares problem. The regularization and weighting matrices are given as follows

$$\boldsymbol{\Pi}_i=\lambda^{i+1}\boldsymbol{\Pi} \qquad \text{and} \qquad \boldsymbol{\Sigma}_i=\boldsymbol{\Lambda}_i, \tag{21}$$

where $0\ll\lambda<1$, $\boldsymbol{\Pi}=\delta^{-1}\mathbf{I}_M$ with $\delta>0$ as a positive constant, and $\boldsymbol{\Lambda}_i\triangleq\mathrm{bdiag}\{\mathbf{I}_N,\lambda\mathbf{I}_N,\ldots,\lambda^i\mathbf{I}_N\}$. Note that $\boldsymbol{\Sigma}_i=\boldsymbol{R}_{v,i}^{-1}\boldsymbol{\Lambda}_i$ and since the noise variances are often unknown, we choose the weighting matrix as in (21).

We can rewrite (20) in the equivalent form

$$\begin{aligned}
\min_{\mathbf{w}}\Big[&(\mathbf{w}-\overline{\mathbf{w}})^*\boldsymbol{\Pi}_i(\mathbf{w}-\overline{\mathbf{w}})+(\boldsymbol{d}_{i,Q}-\mathbf{w}^*\boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q})\\
&\qquad\boldsymbol{\Sigma}_i(\boldsymbol{d}_{i,Q}-\mathbf{w}^*\boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q})^*\Big]
\end{aligned} \tag{22}$$

To solve (22), we reduce it to the standard least-squares form by introducing the eigendecompositions of $\boldsymbol{\Pi}_i$ as follows

$$\boldsymbol{\Pi}_i=\boldsymbol{\Psi}_i\boldsymbol{\Delta}_i\boldsymbol{\Psi}_i^*, \tag{23}$$

where $\boldsymbol{\Delta}_i$ is diagonal with positive entries, and unitary, i.e., it satisfies $\boldsymbol{\Psi}_i\boldsymbol{\Psi}_i^*=\boldsymbol{\Psi}_i^*\boldsymbol{\Psi}_i=\mathbf{I}_M$. Let $\boldsymbol{\Lambda}_i^{1/2}$ denote the diagonal matrix whose entries are equal to the positive square roots of the entries of $\boldsymbol{\Lambda}_i$, and define the change of variables

$$\mathbf{a}\triangleq\boldsymbol{d}_{i,Q}\boldsymbol{\Lambda}_i^{1/2}, \qquad \text{and} \qquad \mathbf{L}\triangleq\boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\boldsymbol{\Lambda}_i^{1/2}. \tag{24}$$

Using (24), we can rewrite (22) as follows

$$\min_{\mathbf{w}}\big[(\mathbf{w}-\overline{\mathbf{w}})^*\boldsymbol{\Pi}_i(\mathbf{w}-\overline{\mathbf{w}})+\|\mathbf{a}-\mathbf{w}^*\mathbf{L}\|^2\big]. \tag{25}$$

Defining the change of variables

$$\mathbf{z}\triangleq\mathbf{w}-\overline{\mathbf{w}} \qquad \text{and} \qquad \mathbf{f}\triangleq\mathbf{a}-\overline{\mathbf{w}}^*\mathbf{L}, \tag{26}$$

we can rewrite (25) as follows

$$\min_{\mathbf{z}}\big[\mathbf{z}^*\boldsymbol{\Pi}_i\mathbf{z}+\|\mathbf{f}-\mathbf{z}^*\mathbf{L}\|^2\big]. \tag{27}$$

Let $\boldsymbol{\Delta}_i^{1/2}$ denote the diagonal matrix whose entries are equal to the positive square roots of the entries of $\boldsymbol{\Delta}_i$. Then using the eigendecomposition in (23), we can write the equivalent form of (27) as follows

$$\min_{\mathbf{z}}\left[\left\|\begin{bmatrix}\mathbf{0} & \mathbf{f}\end{bmatrix}-\mathbf{z}^*\begin{bmatrix}\boldsymbol{\Psi}_i^*\boldsymbol{\Delta}_i^{1/2} & \mathbf{L}\end{bmatrix}\right\|^2\right], \tag{28}$$

which is a form of the standard least-squares

$$\min_{\mathbf{w}}\left[\big\|\mathbf{y}-\mathbf{w}^*\mathbf{H}\big\|^2\right], \tag{29}$$

in which the roles of $\mathbf{y}$ and $\mathbf{H}$ are played by $\begin{bmatrix}\mathbf{0} & \mathbf{f}\end{bmatrix}$ and $\begin{bmatrix}\boldsymbol{\Psi}_i^*\boldsymbol{\Delta}_i^{1/2} & \mathbf{L}\end{bmatrix}$, respectively. All solutions $\widehat{\mathbf{w}}$ to the least-squares problem (29) are characterized as solutions to the linear system of equations

$$\widehat{\mathbf{w}}^*\mathbf{H}\mathbf{H}^*=\mathbf{y}\mathbf{H}^*. \tag{30}$$

We replace $\mathbf{y}$ and $\mathbf{H}$ with $\begin{bmatrix}\mathbf{0} & \mathbf{f}\end{bmatrix}$ and $\begin{bmatrix}\boldsymbol{\Psi}_i^*\boldsymbol{\Delta}_i^{1/2} & \mathbf{L}\end{bmatrix}$ in (30) to form the solutions $\widehat{\mathbf{z}}$ to (27) as follows

$$\left(\begin{bmatrix}\mathbf{0} & \mathbf{f}\end{bmatrix}-\widehat{\mathbf{z}}^*\begin{bmatrix}\boldsymbol{\Psi}_i^*\boldsymbol{\Delta}_i^{1/2} & \mathbf{L}\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{\Psi}_i^*\boldsymbol{\Delta}_i^{1/2} & \mathbf{L}\end{bmatrix}^*=0. \tag{31}$$

Using $\widehat{\mathbf{z}} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ and (26) in (31), we can write the solution $\widehat{\mathbf{w}}$ to (27) in a system of equations as follows

$$(\widehat{\mathbf{w}} - \overline{\mathbf{w}})^*[\mathbf{\Pi}_i + \mathbf{L}\mathbf{L}^*] = (\mathbf{a} - \overline{\mathbf{w}}^*\mathbf{L})\mathbf{L}^*, \tag{32}$$

or in an equivalent form the solution $\mathbf{w}(i)$ is given by

$$\mathbf{w}(i) = \overline{\mathbf{w}} + [\mathbf{\Pi}_i + \mathbf{L}\mathbf{L}^*]^{-1}\mathbf{L}(\mathbf{a} - \overline{\mathbf{w}}^*\mathbf{L})^*. \tag{33}$$

Using (24), the solution $\mathbf{w}(i)$ to the weighted regularized least-squares (20) is given by

$$\begin{aligned}\mathbf{w}(i) = \overline{\mathbf{w}} + &(\mathbf{\Pi}_i + \boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\mathbf{\Sigma}_i\boldsymbol{X}_{i,Q}^*\boldsymbol{B}_{i,b}^*)^{-1} \\ &\boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\mathbf{\Sigma}_i(\boldsymbol{d}_{i,Q} - \overline{\mathbf{w}}^*\boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q})^*.\end{aligned} \tag{34}$$

To simplify, we use (21), assume the given column vector $\overline{\mathbf{w}} = \mathbf{0}$, and write an exponentially weighted version of (34) as follows

$$\begin{aligned}\mathbf{w}(i) = (\lambda^{i+1}\mathbf{\Pi} + \boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\mathbf{\Lambda}_i\boldsymbol{X}_{i,Q} * \boldsymbol{B}_{i,b}^*)^{-1} \\ \boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\mathbf{\Lambda}_i\boldsymbol{d}_{i,Q}^*.\end{aligned} \tag{35}$$

To form the recursions, we introduce $\boldsymbol{P}_i$ as follows

$$\boldsymbol{P}_i = (\lambda^{i+1}\mathbf{\Pi} + \boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\mathbf{\Lambda}_i\boldsymbol{X}_{i,Q}^*\boldsymbol{B}_{i,b}^*)^{-1}, \tag{36}$$

and write down (35) as follows

$$\mathbf{w}(i) = \boldsymbol{P}_i\boldsymbol{B}_{i,b}\boldsymbol{X}_{i,Q}\mathbf{\Lambda}_i\boldsymbol{d}_{i,Q}^*. \tag{37}$$

We then reformulate the global least-squares solution in (37) as a local least-squares solution where nodes have access to limited data from their neighbors in the diffusion scheme [12]. In this scheme, nodes update their local intermediate estimates $\mathbf{h}_k(i)$ following (37) which yields

$$\mathbf{h}_k(i) = \mathbf{P}_k(i)\boldsymbol{B}_{k,b}(i)\boldsymbol{X}_{k,Q}(i)\mathbf{\Lambda}_k(i)\boldsymbol{d}_{k,Q}^*(i), \tag{38}$$

where

$$\mathbf{P}_k(i) = (\lambda^{i+1}\mathbf{\Pi} + \boldsymbol{B}_{k,b}(i)\boldsymbol{X}_{k,Q}(i)\mathbf{\Lambda}_k(i)\boldsymbol{X}_{k,Q}^*(i)\boldsymbol{B}_{k,b}^*(i))^{-1},$$

and $\boldsymbol{X}_{k,Q}(i)$, $\boldsymbol{d}_{k,Q}(i)$, and $\boldsymbol{B}_{k,b}(i)$ are the collected quantities from time 0 to time i at node k.

At the combination step, the nodes communicate their local intermediate estimates with their neighbors and provide a combined estimate $\mathbf{w}_k(i)$ as follows

$$\mathbf{w}_k(i) = \sum_{l\in\mathcal{N}_k} a_{l,k}\mathbf{h}_l(i), \tag{39}$$

where $\mathcal{N}_k$ is the neighborhood of node k (possibly including itself) and the combination coefficients, $a_{l,k}$, of neighbor nodes on node k are chosen such that

$$a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k, a_{l,k} > 0 \text{ if } l \in \mathcal{N}_k, \text{ and } \sum_{l\in\mathcal{N}_k} a_{l,k} = 1. \tag{40}$$

To form the recursion, we compute $\mathbf{P}_k(i)$ from $\mathbf{P}_k(i-1)$ as follows

$$\begin{aligned}\mathbf{P}_k^{-1}(i) &= \lambda[\lambda^i\mathbf{\Pi} + \boldsymbol{B}_{k,b}(i-1)\boldsymbol{X}_{k,Q}(i-1)\mathbf{\Lambda}_k(i-1) \\ &\quad \boldsymbol{X}_{k,Q}^*(i-1)\boldsymbol{B}_{k,b}^*(i-1)] + \boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^*(i) \\ &= \lambda\mathbf{P}_k^{-1}(i-1) + \boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^*(i).\end{aligned} \tag{41}$$

Using the matrix inversion lemma

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1},$$

and the assignments

$$\begin{aligned} A &\leftarrow \lambda\mathbf{P}_k^{-1}(i), & B &\leftarrow \boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i), \\ C &\leftarrow 1, & D &\leftarrow \mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^*(i), \end{aligned}$$

we obtain the recursion for updating $\mathbf{P}_k(i)$ given by

$$\begin{aligned}\mathbf{P}_k(i) = \lambda^{-1}\Bigg(&\mathbf{P}_k(i-1) - \\ &\frac{\lambda^{-1}\mathbf{P}_k(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^*(i)\mathbf{P}_k(i-1)}{1 + \lambda^{-1}\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^*(i)\mathbf{P}_k(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)}\Bigg).\end{aligned} \tag{42}$$

To form the recursion for $\mathbf{h}_k(i)$, we rewrite (38) as follows

$$\begin{aligned}\mathbf{h}_k(i) &= \mathbf{P}_k(i)\Big[\boldsymbol{B}_{k,b}(i-1)\boldsymbol{X}_{k,Q}(i-1)\lambda\boldsymbol{\Lambda}_k(i-1)\\ &\qquad \boldsymbol{d}^*_{k,Q}(i-1)+\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)d_{k,Q}(i)\Big]\\ &= \mathbf{P}_k(i)\Big[\mathbf{P}_k^{-1}(i-1)\mathbf{P}_k(i-1)\boldsymbol{B}_{k,b}(i-1)\boldsymbol{X}_{k,Q}(i-1)\\ &\qquad \lambda\boldsymbol{\Lambda}_k(i-1)\boldsymbol{d}^*_{k,Q}(i-1)+\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)d^*_{k,Q}(i)\Big]\end{aligned} \tag{43}$$

From (41), we have $\mathbf{P}_k^{-1}(i-1)=\lambda^{-1}\big(\mathbf{P}_k^{-1}(i)-\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)\boldsymbol{\beta}^*_{k,b}(i)\big)$. Then considering $\mathbf{h}_k(i-1)=\mathbf{P}_k(i-1)\boldsymbol{B}_{k,b}(i-1)\boldsymbol{X}_{k,Q}(i-1)\boldsymbol{\Lambda}_k(i-1)\boldsymbol{d}^*_{k,Q}(i-1)$ from (38) and the fact that $\mathbf{P}_k$ is a symmetric matrix, we can write (43) as follows

$$\begin{aligned}\mathbf{h}_k(i) &= \mathbf{P}_k(i)\mathbf{P}_k^{-1}(i-1)\lambda\mathbf{h}_k(i-1)\\ &\qquad +\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)d^*_{k,Q}(i)\\ &= \mathbf{h}_k(i-1)-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)\boldsymbol{\beta}^*_{k,b}(i)\mathbf{h}_k(i-1)\\ &\qquad +\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)d^*_{k,Q}(i)\\ &= \mathbf{h}_k(i-1)+\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\\ &\qquad \big(d^*_{k,Q}(i)-\mathbf{x}^*_{k,Q}(i)\boldsymbol{\beta}^*_{k,b}(i)\mathbf{h}_k(i-1)\big).\end{aligned} \tag{44}$$

Note that with diffusion, at time instant i, each node uses its combined estimate $\mathbf{w}_k(i-1)$ instead of intermediate estimate $\mathbf{h}_k(i-1)$ to update $\mathbf{h}_k(i)$. This allows the combined estimates diffuse into the network. We then update $\mathbf{h}_k(i)$ as follows

$$\mathbf{h}_k(i)=\mathbf{w}_k(i-1)+\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\bar{e}^*_k(i), \tag{45}$$

where

$$\bar{e}_k(i)=d_{k,Q}(i)-\mathbf{w}^*_k(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i) \tag{46}$$

The next section shows how the bias compensation $\boldsymbol{\beta}_{k,b}(i)$ should be chosen such that (39) is asymptotically unbiased in the mean.

IV. Analysis and Bias Compensation

In this section, we present a statistical analysis of the proposed DQA-RLS algorithm that focuses on the mean and mean-square performances. In addition, we devise a bias compensation strategy and investigate the computational complexity and the power consumption of the proposed and existing algorithms.

A. Assumptions and Definitions

Combining (21) with (41), we can write $\mathbf{P}_k^{-1}(i)$ as follows:

$$\begin{aligned}\mathbf{P}_k^{-1}(i) &= \lambda\mathbf{P}_k^{-1}(i-1)+\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)\boldsymbol{\beta}^*_{k,b}(i)\\ &= \lambda^{i+1}\boldsymbol{\Pi}+\sum_{j=0}^{i}\lambda^{i-j}\boldsymbol{\beta}_{k,b}(j)\mathbf{x}_{k,Q}(j)\mathbf{x}^*_{k,Q}(j)\boldsymbol{\beta}^*_{k,b}(j).\end{aligned} \tag{47}$$

We aim to evaluate the mean behavior of the matrix $\mathbf{P}_k(i)$. Choosing $0\ll\lambda<1$, as $i\to\infty$, the steady-state mean value of $\mathbf{P}_k^{-1}(i)$ is given by

$$\begin{aligned}\lim_{i\to+\infty}\mathbb{E}[\mathbf{P}_k^{-1}(i)] &=\\ &\frac{1}{1-\lambda}\mathbb{E}[\boldsymbol{\beta}_{k,b}(j)\mathbf{x}_{k,Q}(j)\mathbf{x}^*_{k,Q}(j)\boldsymbol{\beta}^*_{k,b}(j)].\end{aligned} \tag{48}$$

In order to simplify the analysis, we use the definition

$$\mathbf{u}_k(j)\triangleq\boldsymbol{\beta}_{k,b}(j)\mathbf{x}_{k,Q}(j), \tag{49}$$

that leads to the following expression for (48):

$$\lim_{i\to+\infty}\mathbb{E}[\mathbf{P}_k^{-1}(i)]=\frac{\mathbf{R}_{u_k}}{1-\lambda}\triangleq\mathbf{P}_k^{-1}. \tag{50}$$

TABLE II
PSEUDO CODE FOR THE DQA-RLS ALGORITHM

Initialization:

Generate $\mathcal{T}_b$ and $\mathcal{L}_b$

Compute $\widehat{\sigma}^2_{q,k}$ from (12)

$\widehat{\sigma}^2_{x_k,Q}(-1) = 0$, $\mathbf{w}_k(-1) = 0$ and $\mathbf{P}_k(-1) = \mathbf{\Pi}^{-1}$

$0 \ll \gamma < 1$ and $0 \ll \lambda < 1$

At each time instant i **and node** k

Receive data

$d_{k,Q}(i)$

$\mathbf{x}_{k,Q}(i) = [x_{k,Q}(i), x_{k,Q}(i-1), \ldots, x_{k,Q}(i-M+1)]^T$

Repeat

$\widehat{\sigma}^2_{x_k,Q}(i) = \gamma\widehat{\sigma}^2_{x_k,Q}(i-1) + (1-\gamma)|x_{k,Q}(i)|^2$

$\widehat{\sigma}^2_{x_k}(i) = \widehat{\sigma}^2_{x_k,Q}(i) + \widehat{\sigma}^2_{q,k}$

$\widehat{g}_{x_k,b}(i) = \frac{1}{\sqrt{\widehat{\sigma}^2_{x_k}(i)}} \sum_{j=0}^{2^b-1} \frac{l_j}{\sqrt{\pi}} \left(e^{-\frac{\tau_j^2}{\widehat{\sigma}^2_{x_k}(i)}} - e^{-\frac{\tau_{j+1}^2}{\widehat{\sigma}^2_{x_k}(i)}} \right)$

$\beta_{k,b}(i) = \frac{\widehat{g}^2_{x_k,b}(i)\widehat{\sigma}^2_{x_k}(i)}{\widehat{g}^2_{x_k,b}(i)\widehat{\sigma}^2_{x_k}(i) + \sigma^2_{q,k}}$

$\bar{e}_k(i) = d_{k,Q}(i) - \mathbf{w}_k^*(i-1)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)$

$$\mathbf{P}_k(i) = \frac{1}{\lambda}\left(\mathbf{P}_k(i-1) - \frac{\mathbf{P}_k(i-1)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)\beta_{k,b}(i)\mathbf{P}_k(i-1)}{\lambda + \mathbf{x}^*_{k,Q}(i)\beta_{k,b}(i)\mathbf{P}_k(i-1)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)} \right)$$

$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mathbf{P}_k(i)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)\bar{e}^*_k(i)$

$\mathbf{w}_k(i) = \sum_{l \in \mathcal{N}_k} a_{l,k}\mathbf{h}_l(i)$

where $\mathbf{R}_{u_k} = \mathbb{E}[\boldsymbol{\beta}_{k,b}(j)\mathbf{x}_{k,Q}(j)\mathbf{x}^*_{k,Q}(j)\boldsymbol{\beta}^*_{k,b}(j)]$.

To start the analysis, we need to introduce some assumptions similar to those often used in the analysis of adaptive algorithms in the literature. Simulation results show that the results obtained under the following assumptions correspond well with real performance of the DQA-RLS algorithm, for stationary data and for a forgetting factor close to unity.

Assumption 1: The input regressors $\mathbf{x}_k(i)$ are zero-mean with covariance matrices $\mathbf{R}_{x_k} = \mathbb{E}[\mathbf{x}_k(i)\mathbf{x}^*_k(i)]$ and temporally independent. This assumption also applies to the additive noise sequences $v_k(i)$ with variance $\sigma^2_{v,k}$ and the quantized regressors $\mathbf{x}_{k,Q}(i)$ with covariance matrices $\mathbf{R}_{x_k,Q} = \mathbb{E}[\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)]$. Moreover, covariance matrices are time-invariant and all data is assumed spatially independent.

Assumption 2: The matrix $\mathbf{P}_k(i)$ varies slowly in relation to $\widetilde{\mathbf{w}}_k(i)$ and $\mathbf{w}_k(i)$. Thus, when they appear inside the expectations we decouple their expected values [38], [39]. Note that this a common assumption for the performance analysis of RLS-type algorithms for λ close to unity. This also applies to $\mathbf{G}_{k,b}$ since (10) is based on the statistics of $\mathbf{x}_k(i)$.

Assumption 3: There is an iteration number i_0 such that for all $i > i_0$, the time-averaged matrices $\mathbf{P}_k(i)$ and $\mathbf{P}^{-1}_k(i)$ can be replaced by their expected values, $\mathbb{E}[\mathbf{P}_k(i)]$ and $\mathbb{E}[\mathbf{P}^{-1}_k(i)]$. Note that such replacements are commonly used in the performance analysis of RLS-type algorithms, see [12], [18], [25], [31], i.e.,

$$\mathbb{E}[\mathbf{P}_k(i)] \approx \mathbf{P}_k \text{ and } \mathbb{E}[\mathbf{P}^{-1}_k(i)] \approx \mathbf{P}^{-1}_k. \tag{51}$$

To analyze the mean and mean-square performance of DQA-RLS, we use the weight-error vectors

$$\widetilde{\mathbf{w}}_k(i) = \mathbf{w}_o - \mathbf{w}_k(i), \qquad \text{and} \qquad \widetilde{\mathbf{h}}_k(i) = \mathbf{w}_o - \mathbf{h}_k(i). \tag{52}$$

The performance of bias-compensated-type adaptive filters is usually compared in terms of the mean-square deviation (MSD) [19], defined by

$$\text{MSD}_k \triangleq \lim_{i \to +\infty} \mathbb{E}[\|\widetilde{\mathbf{w}}_k(i)\|^2] = \mathbb{E}\big[\,\text{Tr}\left(\widetilde{\mathbf{w}}_k(i)\widetilde{\mathbf{w}}^*_k(i)\right)\big]. \tag{53}$$

We define the $N \times N$ matrix $\mathbf{A} = [a_{l,k}]$, and the following matrices:

$$\begin{aligned} \boldsymbol{\mathcal{A}} &= \mathbf{A} \otimes \mathbf{I}_M \qquad (MN \times MN) \\ \widetilde{\mathbf{W}}_i &= \operatorname{col}\left\{\widetilde{\mathbf{w}}_1(i), \ldots, \widetilde{\mathbf{w}}_N(i)\right\} \quad (MN \times 1). \end{aligned} \tag{54}$$

B. Mean Performance Analysis

Using the decomposition model in (14), we can rewrite the error in (16) as follows

$$\begin{aligned} \bar{e}_k(i) &= d_{k,Q}(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{x}_{k,Q}(i) \\ &= g_{k,b}\mathbf{w}_o^*\mathbf{x}_k(i) + \bar{q}_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_{k,b}(i) \\ &\quad \big(\mathbf{G}_{k,b}\mathbf{x}_k(i) + \mathbf{q}_{x,k}(i)\big) \\ &= g_{k,b}\mathbf{w}_o^*\mathbf{x}_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i) \\ &\quad - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_{k,b}(i)\mathbf{q}_{x,k}(i) + \bar{q}_k(i). \end{aligned} \tag{55}$$

Combining (55) with (45) and subtracting from $\mathbf{w}_o$ yields

$$\begin{aligned} \widetilde{\mathbf{h}}_k(i) &= \widetilde{\mathbf{w}}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\Big(\mathbf{G}_{k,b}\mathbf{x}_k(i) + \mathbf{q}_{x,k}(i)\Big) \\ &\quad \Big(\mathbf{x}_k^*(i)g_{k,b}\mathbf{w}_o - \mathbf{x}_k^*(i)\mathbf{G}_{k,b}\boldsymbol{\beta}_{k,b}^*(i)\mathbf{w}_k(i-1) \\ &\quad - \mathbf{q}_{x,k}^*(i)\boldsymbol{\beta}_{k,b}^*(i)\mathbf{w}_k(i-1) + \bar{q}_k^*(i)\Big) \\ &= \widetilde{\mathbf{w}}_k(i-1) + \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i) \\ &\quad \mathbf{x}_k^*(i)\mathbf{G}_{k,b}\boldsymbol{\beta}_{k,b}^*(i)\mathbf{w}_k(i-1) \\ &\quad + \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{q}_{x,k}(i)\mathbf{q}_{x,k}^*(i)\boldsymbol{\beta}_{k,b}^*(i)\mathbf{w}_k(i-1) \\ &\quad - \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{k,b}\mathbf{w}_o \\ &\quad + \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{q}_{x,k}^*(i)\boldsymbol{\beta}_{k,b}^*(i)\mathbf{w}_k(i-1) \\ &\quad - \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i)\bar{q}_k^*(i) \\ &\quad + \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{q}_{x,k}(i)\mathbf{x}_k^*(i)\mathbf{G}_{k,b}\boldsymbol{\beta}_{k,b}^*(i)\mathbf{w}_k(i-1) \\ &\quad - \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{q}_{x,k}(i)\mathbf{x}_k^*(i)g_{k,b}\mathbf{w}_o \\ &\quad - \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{q}_{x,k}(i)\bar{q}_k^*(i). \end{aligned} \tag{56}$$

The errors $\bar{q}_k$, $\mathbf{q}_{x,k}$, and the regressors $\mathbf{x}_k$ are assumed statistically independent, so that the expectation of the cross terms vanishes. Taking the expectation from the remaining terms in (56), we obtain

$$\begin{aligned} \mathbb{E}\big[\widetilde{\mathbf{h}}_k(i)\big] &= \mathbb{E}\big[\widetilde{\mathbf{w}}_k(i-1)\big] + \mathbb{E}\Big[\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\Big(\mathbf{G}_{k,b}\mathbf{x}_k(i) \\ &\quad \mathbf{x}_k^*(i)\mathbf{G}_{k,b} + \mathbf{q}_{x,k}(i)\mathbf{q}_{x,k}^*(i)\Big)\boldsymbol{\beta}_{k,b}^*(i)\Big]\mathbb{E}\big[\mathbf{w}_k(i-1)\big] \\ &\quad - \mathbb{E}\big[\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{k,b}\big]\mathbf{w}_o. \end{aligned} \tag{57}$$

Let us define two $M \times M$ matrices that include terms multiplied by $\mathbf{w}_k(i-1)$ and $\mathbf{w}_o$ in (57) as follows:

$$\begin{aligned} \boldsymbol{\Theta}_k(i) &\triangleq \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\big(\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{G}_{k,b} \\ &\quad + \mathbf{q}_{x,k}(i)\mathbf{q}_{x,k}^*(i)\big)\boldsymbol{\beta}_{k,b}^*(i) \\ \boldsymbol{\Gamma}_k(i) &\triangleq \mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{k,b}. \end{aligned} \tag{58}$$

We show next that a necessary but not sufficient condition to have an asymptotically unbiased solution in the mean is that

$$\mathbb{E}\big[\boldsymbol{\Theta}_k(i)\big] = \mathbb{E}\big[\boldsymbol{\Gamma}_k(i)\big], \tag{59}$$

and we show in the next section that this condition is possible by appropriately choosing $\boldsymbol{\beta}_{k,b}(i)$. Assuming (59), we can write (57) as follows

$$\mathbb{E}\big[\widetilde{\mathbf{h}}_k(i)\big] = \mathbb{E}\big[\widetilde{\mathbf{w}}_k(i-1)\big] - \mathbb{E}\big[\boldsymbol{\Theta}_k(i)\big]\mathbb{E}\big[\widetilde{\mathbf{w}}_k(i-1)\big]. \tag{60}$$

From (47) and (50), we can verify that for sufficiently large i under Assumption 3, we have

$$\mathbb{E}\big[\boldsymbol{\Theta}_k(i)\big] = 1 - \lambda. \tag{61}$$

We now apply the weight-error vectors to the combined estimates (39) and obtain

$$\widetilde{\mathbf{w}}_k(i) = \sum_{l\in\mathcal{N}_k} a_{l,k}\widetilde{\mathbf{h}}_l(i), \tag{62}$$

and taking the expectation from both sides of it and using (60) and (61), we arrive at

$$\mathbb{E}\big[\widetilde{\mathbf{w}}_k(i)\big] = \sum_{l\in\mathcal{N}_k} a_{l,k}(1-\lambda)\mathbb{E}\big[\widetilde{\mathbf{w}}_k(i-1)\big]. \tag{63}$$

For the global estimates and $i > i_0$, we have

$$\mathbb{E}[\widetilde{\mathbf{W}}_i] = (1-\lambda)\boldsymbol{\mathcal{A}}\mathbb{E}[\widetilde{\mathbf{W}}_{i-1}] = (1-\lambda)^{i-i_0}\boldsymbol{\mathcal{A}}^{i-i_0}\mathbb{E}[\widetilde{\mathbf{W}}_{i_0}]. \tag{64}$$

Observing (64) and recalling that the spectral radius of $\boldsymbol{\mathcal{A}}$ (i.e., the largest eigenvalue in modulus) is equal to one [40], we can state that for sufficiently large i_0 (or equivalently in adaptive filtering, when the algorithm reaches the steady-state), assuming $\widetilde{\mathbf{W}}_{i_0}$ is element-wise bounded by some finite constant and regarding (40), for $0 \ll \lambda < 1$, the term on the right-hand side of (64) converges to zero and DQA-RLS is asymptotically unbiased in the mean.

C. Proposed Bias Compensation

In this section, we derive an expression for the bias compensation term $\boldsymbol{\beta}_{k,b}(i)$ such that (59) is true and (39) is asymptotically unbiased in the mean. From (58) and (59), we must have

$$\begin{aligned}\mathbb{E}\Big[&\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\big(\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{G}_{k,b}\\ &+\mathbf{q}_{x,k}(i)\mathbf{q}_{x,k}^*(i)\big)\boldsymbol{\beta}_{k,b}^*(i)\Big]\\ &=\mathbb{E}\Big[\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}(i)\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{k,b}\Big].\end{aligned} \tag{65}$$

Under Assumptions 2 and 3, we can write (65) as follows

$$\begin{aligned}\Big(&\mathbf{G}_{k,b}\mathbb{E}\big[\mathbf{x}_k(i)\mathbf{x}_k^*(i)\big]\mathbf{G}_{k,b}\\ &+\mathbb{E}\big[\mathbf{q}_{x,k}(i)\mathbf{q}_{x,k}^*(i)\big]\Big)\boldsymbol{\beta}_{k,b}^*(i)\\ &=\mathbf{G}_{k,b}\mathbb{E}\big[\mathbf{x}_k(i)\mathbf{x}_k^*(i)\big]g_{k,b}.\end{aligned} \tag{66}$$

Therefore, the bias compensation term is expressed by

$$\boldsymbol{\beta}_{k,b}(i) = \boldsymbol{\beta}_{k,b} = g_{k,b}\mathbf{R}_{x_k}\mathbf{G}_{k,b}\Big(\mathbf{G}_{k,b}\mathbf{R}_{x_k}\mathbf{G}_{k,b} + \mathbf{R}_{q_x,k}\Big)^{-1}, \tag{67}$$

which needs an $M\times M$ matrix inversion at each time instant i if $\mathbf{x}_k(i)$ are nonstationary and, moreover, we realize $\boldsymbol{\beta}_{k,b}\in\mathbb{R}^{M\times M}$ is time-invariant for stationary inputs. In what follows, we show how to compute the bias compensation term to reduce the complexity of our proposed algorithm.

Remark 1: *(Stationary data).* When the input regressors $\mathbf{x}_k(i)$ are wide-sense stationary, we use $\mathbf{R}_{x_k} = \mathbb{E}[\mathbf{x}_k(i)\mathbf{x}_k^*(i)] \approx \sigma_{x,k}^2\mathbf{I}_M$ and the matrix $\mathbf{G}_{k,b}$ reduces to $g_{x_k,b}\mathbf{I}_M$ with $g_{x_k,b}$ estimated as follows

$$\widehat{g}_{x_k,b}(i) = \frac{1}{\sqrt{\widehat{\sigma}_{x_k}^2(i)}}\sum_{j=0}^{2^b-1}\frac{l_j}{\sqrt{\pi}}\left(e^{-\frac{\tau_j^2}{\widehat{\sigma}_{x_k}^2(i)}} - e^{-\frac{\tau_{j+1}^2}{\widehat{\sigma}_{x_k}^2(i)}}\right), \tag{68}$$

where $\widehat{\sigma}_{x_k}^2(i)$ is an instantaneous approximation to $\sigma_{x,k}^2$ given by the following recursions:

$$\begin{aligned}\widehat{\sigma}_{x_k,Q}^2(i) &= \gamma\widehat{\sigma}_{x_k,Q}^2(i-1) + (1-\gamma)|x_{k,Q}(i)|^2,\\ \widehat{\sigma}_{x_k}^2(i) &= \widehat{\sigma}_{x_k,Q}^2(i) + \widehat{\sigma}_{q,k}^2,\end{aligned} \tag{69}$$

where $\widehat{\sigma}_{x_k,Q}^2(-1) = 0$ and $0 \ll \gamma < 1$ and $\widehat{\sigma}_{q,k}^2$ is given by (12). Therefore, the bias compensation term is given by

$$\boldsymbol{\beta}_{k,b}(i) = \beta_{k,b}(i)\mathbf{I}_M = \frac{\widehat{g}_{x_k,b}(i)\widehat{g}_{k,b}(i)\widehat{\sigma}_{x_k}^2(i)}{\widehat{g}_{x_k,b}^2(i)\widehat{\sigma}_{x_k}^2(i) + \widehat{\sigma}_{q,k}^2}\mathbf{I}_M. \tag{70}$$

Remark 2: *(One ADC for each node).* To reduce the complexity of our algorithm, we design only one ADC to quantize both the input regressors and desired signals. This also covers the network with nodes in which the two ADCs use the same set of thresholds and the same set of labels. Then $g_{x_k,b}$ and $g_{k,b}$ can be considered equal and the bias compensation term is given by

$$\boldsymbol{\beta}_{k,b}(i) = \frac{\widehat{g}_{x_k,b}^2(i)\widehat{\sigma}_{x_k}^2(i)}{\widehat{g}_{x_k,b}^2(i)\widehat{\sigma}_{x_k}^2(i) + \widehat{\sigma}_{q,k}^2}\mathbf{I}_M. \tag{71}$$

This is the simple version of the bias compensation term that we use in the proposed DQA-RLS algorithm which is detailed in table II, and as we show in the simulation results, its MSD performance is better than that of the DRLS algorithm with coarsely quantized signals even for colored inputs and imperfect gain control ($\sigma_{x_k}^2 \neq 1$).

D. Mean-Square Performance Analysis

In this section, we carry out a mean-square performance analysis and discuss the steady-state behavior of the DQA-RLS algorithm. We first write (56) as

$$\begin{aligned}\widetilde{\mathbf{h}}_k(i) = &\mathbf{P}_k(i)\mathbf{P}_k^{-1}(i)\widetilde{\mathbf{w}}_k(i-1)\\ &-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{x}_{k,Q}(i)d_{k,Q}^*(i)\\ &+\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{x}_{k,Q}(i)\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^T\mathbf{w}_k(i-1).\end{aligned} \tag{72}$$

Using (47), we obtain

$$\begin{aligned}\widetilde{\mathbf{h}}_k(i) &= \mathbf{P}_k(i)\lambda\mathbf{P}_k^{-1}(i-1)\widetilde{\mathbf{w}}_k(i-1)\\ &\quad-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{x}_{k,Q}(i)d_{k,Q}^*(i)\\ &\quad+\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{x}_{k,Q}(i)\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^T\\ &\qquad\Big(\widetilde{\mathbf{w}}_k(i-1)+\mathbf{w}_k(i-1)\Big)\\ &= \lambda\widetilde{\mathbf{w}}_k(i-1)-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{x}_{k,Q}(i)d_{k,Q}^*(i)\\ &\quad+\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{x}_{k,Q}(i)\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^T\mathbf{w}_o.\end{aligned} \tag{73}$$

We now use (14) and (15) to write (73) as follows

$$\begin{aligned}\widetilde{\mathbf{h}}_k(i) = &\lambda\widetilde{\mathbf{w}}_k(i-1)\\ &-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{G}_{k,b}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{k,b}\mathbf{w}_o\\ &-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{G}_{k,b}\mathbf{x}_k(i)\bar{q}_k^*(i)\\ &-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{q}_{x,k}(i)\mathbf{x}_k^*(i)g_{k,b}\mathbf{w}_o\\ &-\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{q}_{x,k}(i)\bar{q}_k^*(i)\\ &+\mathbf{P}_k(i)\boldsymbol{\beta}_{k,b}\mathbf{x}_{k,Q}(i)\mathbf{x}_{k,Q}^*(i)\boldsymbol{\beta}_{k,b}^T\mathbf{w}_o.\end{aligned} \tag{74}$$

In order to simplify this expression, we assume that the choice of $\boldsymbol{\beta}_{k,b}$ in (67) is such that the instantaneous values in the second and sixth terms on the RHS of (74) are equal with different signs and vanish for sufficiently large i. Therefore, the weight-error vectors of the combined estimates in (62) are given by

$$\begin{aligned}\widetilde{\mathbf{w}}_k(i) = &\lambda\sum_{l\in\mathcal{N}_k}a_{l,k}\widetilde{\mathbf{w}}_l(i-1)\\ &-\sum_{l\in\mathcal{N}_k}a_{l,k}\mathbf{P}_l\boldsymbol{\beta}_{l,b}\mathbf{G}_{l,b}\mathbf{x}_l(i)\bar{q}_l^*(i)\\ &-\sum_{l\in\mathcal{N}_k}a_{l,k}\mathbf{P}_l\boldsymbol{\beta}_{l,b}\mathbf{q}_{x,l}(i)\mathbf{x}_l^*(i)g_{l,b}\mathbf{w}_o\\ &-\sum_{l\in\mathcal{N}_k}a_{l,k}\mathbf{P}_l\boldsymbol{\beta}_{l,b}\mathbf{q}_{x,l}(i)\bar{q}_l^*(i).\end{aligned} \tag{75}$$

Let us now define

$$\begin{aligned}\boldsymbol{\mathcal{P}} &= \text{bdiag}\left\{\mathbf{P}_1,\ldots,\mathbf{P}_N\right\} && (MN\times MN)\\ \boldsymbol{\mathcal{B}}_b &= \text{bdiag}\left\{\boldsymbol{\beta}_{1,b},\ldots,\boldsymbol{\beta}_{N,b}\right\} && (MN\times MN)\\ \boldsymbol{\mathcal{G}}_b &= \text{bdiag}\left\{\mathbf{G}_{1,b},\ldots,\mathbf{G}_{N,b}\right\} && (MN\times MN)\\ \boldsymbol{\Upsilon}_i &= \text{bdiag}\left\{\mathbf{q}_{x,1}(i)\mathbf{x}_1^*(i),\ldots,\right.\\ &\qquad\left.\mathbf{q}_{x,N}(i)\mathbf{x}_N^*(i)\right\} && (MN\times MN)\\ \boldsymbol{\xi}_i &= \text{col}\left\{\mathbf{x}_1(i)\bar{q}_1^*(i),\ldots,\mathbf{x}_N(i)\bar{q}_N^*(i)\right\} && (MN\times 1)\\ \boldsymbol{\zeta}_i &= \text{col}\left\{\mathbf{q}_{x,1}(i)\bar{q}_1^*(i),\ldots,\mathbf{q}_{x,N}(i)\bar{q}_N^*(i)\right\} && (MN\times 1)\\ \boldsymbol{\eta} &= \text{col}\left\{g_{1,b}\mathbf{w}_o,\ldots,g_{N,b}\mathbf{w}_o\right\} && (MN\times 1),\end{aligned}$$

and write $\widetilde{\mathbf{W}}_i$ in a more compact form as

$$\widetilde{\mathbf{W}}_i = \lambda\boldsymbol{\mathcal{A}}\widetilde{\mathbf{W}}_{i-1}-\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\mathcal{G}}_b\boldsymbol{\xi}_i-\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\Upsilon}_i\boldsymbol{\eta}-\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\zeta}_i.$$

Taking the expectation of $\widetilde{\mathbf{W}}_i\widetilde{\mathbf{W}}_i^*$, we obtain

$$\begin{aligned}\mathbb{E}\big[\widetilde{\mathbf{W}}_i\widetilde{\mathbf{W}}_i^*\big] &= \lambda^2\mathbb{E}\big[\boldsymbol{\mathcal{A}}\widetilde{\mathbf{W}}_{i-1}\widetilde{\mathbf{W}}_{i-1}^*\boldsymbol{\mathcal{A}}^T\big]\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\mathcal{G}}_b\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}_b^T\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big],\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big],\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big]\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\mathcal{G}}_b\boldsymbol{\xi}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big],\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}_b^T\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big],\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\mathcal{G}}_b\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big],\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}_b^T\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big]\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big],\\ &+ \mathbb{E}\big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}_b\boldsymbol{\zeta}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}_b^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\big].\end{aligned} \tag{76}$$

We now use the commutative property of the expectation and vectorization operations, and the relationship between the vectorization operation and the Kronecker product, $\mathrm{vec}(ABC) = (C^T \otimes A)\,\mathrm{vec}(B)$, to write (76) as follows

$$\mathrm{vec}(\boldsymbol{\Omega}_i) = \lambda^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\,\mathrm{vec}(\boldsymbol{\Omega}_{i-1}) + \mathrm{vec}(\boldsymbol{\Xi}_i), \tag{77}$$

where $\boldsymbol{\Omega}_i = \mathbb{E}\big[\widetilde{\mathbf{W}}_i\widetilde{\mathbf{W}}_i^*\big]$ and $\boldsymbol{\Xi}_i$ denotes the summation of the second term to the last one on the RHS of (76). Note that $\mathrm{vec}(\boldsymbol{\Omega}_i)$ is stable if and only if the spectral radius of $\lambda^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})$ is strictly smaller than 1 or $|\lambda| < 1$. Therefore, we obtain

$$\lim_{i\to+\infty}\mathrm{vec}(\boldsymbol{\Omega}_i) = \big(\mathbf{I}_{M^2N^2} - \lambda^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\big)^{-1}\,\mathrm{vec}(\boldsymbol{\Xi}_{+\infty}). \tag{78}$$

For sufficiently large i, taking into account Assumption 3 and (50), we have

$$\mathbf{P}_k = (1-\lambda)\big[\boldsymbol{\beta}_{k,b}\mathbf{R}_{x_k,Q}\boldsymbol{\beta}_{k,b}\big]^{-1},$$

thus $\mathrm{vec}(\boldsymbol{\Xi}_{+\infty})$ is given by

$$\begin{aligned}\mathrm{vec}(\boldsymbol{\Xi}_{+\infty}) = \lim_{i\to+\infty}\mathrm{vec}(\boldsymbol{\Xi}_i) &= (1-\lambda)^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\\ &\mathrm{vec}\Big(\boldsymbol{\mathcal{B}}_b^{-1}\boldsymbol{\mathcal{R}}_{x_Q}^{-1}\Big(\boldsymbol{\mathcal{G}}_b\mathbb{E}\big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\big]\boldsymbol{\mathcal{G}}_b^T + \mathbb{E}\big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\big]\\ &+ \mathbb{E}\big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\big] + \boldsymbol{\mathcal{G}}_b\mathbb{E}\big[\boldsymbol{\xi}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\big] + \mathbb{E}\big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\xi}_i^*\big]\boldsymbol{\mathcal{G}}_b^T\\ &+ \boldsymbol{\mathcal{G}}_b\mathbb{E}\big[\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\big] + \mathbb{E}\big[\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\big]\boldsymbol{\mathcal{G}}_b^T + \mathbb{E}\big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\zeta}_i^*\big]\\ &+ \mathbb{E}\big[\boldsymbol{\zeta}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\big]\Big)\boldsymbol{\mathcal{R}}_{x_Q}^{-1}\boldsymbol{\mathcal{B}}_b^{-1}\Big)\\ &\approx (1-\lambda)^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\\ &\quad\mathrm{vec}\Big(\boldsymbol{\mathcal{B}}_b^{-1}\boldsymbol{\mathcal{R}}_{x_Q}^{-1}\Big(\boldsymbol{\mathcal{G}}_b\mathbb{E}\big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\big]\boldsymbol{\mathcal{G}}_b^T\\ &\quad+ \mathbb{E}\big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\big] + \mathbb{E}\big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\big]\Big)\boldsymbol{\mathcal{R}}_{x_Q}^{-1}\boldsymbol{\mathcal{B}}_b^{-1}\Big),\end{aligned} \tag{79}$$

where under Assumptions 2 and 3, the expectations of the cross-terms vanish and

$$\begin{aligned}\boldsymbol{\mathcal{R}}_{x_Q} &= \mathrm{bdiag}\big\{\mathbf{R}_{x_1,Q},\ldots,\mathbf{R}_{x_N,Q}\big\}\\ \mathbb{E}\big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\big] &= \mathrm{bdiag}\big\{\mathbf{R}_{x_1}\sigma^2_{\bar{q}_1},\ldots,\mathbf{R}_{x_N}\sigma^2_{\bar{q}_N}\big\}\\ \mathbb{E}\big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\big] &= \mathrm{bdiag}\big\{g_{1,b}^2\mathbf{R}_{q_x,1}\big(\mathbf{w}_o^*\mathbf{R}_{x_1}\mathbf{w}_o\big),\ldots,\\ &\qquad g_{N,b}^2\mathbf{R}_{q_x,N}\big(\mathbf{w}_o^*\mathbf{R}_{x_N}\mathbf{w}_o\big)\big\}\\ \mathbb{E}\big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\big] &= \mathrm{bdiag}\big\{\mathbf{R}_{q_x,1}\sigma^2_{\bar{q}_1},\ldots,\mathbf{R}_{q_x,N}\sigma^2_{\bar{q}_N}\big\}.\end{aligned} \tag{80}$$

The analytical computation of $\mathbf{R}_{x_k,Q}$, $\mathbf{R}_{q_x,k}$, and $\sigma^2_{\bar{q}_k}$ is detailed in the Appendix.

Therefore, the steady-state MSD at node k is given by

$$\begin{aligned}\mathrm{MSD}_k &= \lim_{i\to+\infty}\mathbb{E}[\|\widetilde{\mathbf{w}}_k(i)\|^2] = \mathrm{vec}^T\left(\mathbf{C}_k\otimes\mathbf{I}_M\right)\mathrm{vec}(\boldsymbol{\Omega}_{+\infty})\\ &= (1-\lambda)^2\,\mathrm{vec}^T\left(\mathbf{C}_k\otimes\mathbf{I}_M\right)\left(\mathbf{I}_{M^2N^2}-\lambda^2(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\right)^{-1}\\ &\quad(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\,\mathrm{vec}\Bigg(\boldsymbol{\mathcal{B}}_b^{-1}\boldsymbol{\mathcal{R}}_{x_Q}^{-1}\Big(\boldsymbol{\mathcal{G}}_b\mathbb{E}\big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\big]\boldsymbol{\mathcal{G}}_b^T\\ &\quad+\mathbb{E}\big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\big]+\mathbb{E}\big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\big]\Big)\boldsymbol{\mathcal{R}}_{x_Q}^{-1}\boldsymbol{\mathcal{B}}_b^{-1}\Bigg),\end{aligned}\tag{81}$$

where $\mathbf{C}_k$ is an $N\times N$ matrix with zero entries and a unity on its kth diagonal entry that selects the part of $\widetilde{\mathbf{W}}_i\widetilde{\mathbf{W}}_i^*$ corresponding to the kth node. It can be seen from (77) that, for $0\ll\lambda<1$ and $\boldsymbol{\mathcal{A}}$ with the entries $a_{l,k}$ subject to (40), the eigenvalues of $\lambda^2\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}}$ remain in the interval $(-1,1)$, and thus DQA-RLS is stable in the mean-square sense and $\left(\mathbf{I}_{M^2N^2}-\lambda^2(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\right)$ in (81) is nonsingular. The global MSD over all the nodes is given by

$$\mathrm{MSD}_{\mathrm{global}} = \frac{1}{N}\sum_{k=1}^{N}\mathrm{MSD}_k = \frac{1}{N}\,\mathrm{Tr}(\boldsymbol{\Omega}_{+\infty}).\tag{82}$$

Remark 3: *(High precision signals, $b=\infty$).* Increasing the number of quantization bits, the diagonal entries of $\mathbf{G}_{k,b}$ approach unity where for high precision signals ($b=\infty$) with $\mathbf{x}_{k,Q}=\mathbf{x}_k$, we have $\mathbf{G}_{k,b}=\mathbf{I}_M$ according to (10) and $\mathbf{R}_{q_x,k}=\mathbf{0}$ according to (14), and consequently $\boldsymbol{\beta}_{k,b}(i)=\mathbf{I}_M$ from (71). For $b=\infty$ we also have $q_{d,k}=0$ and $g_{k,b}=1$, and since $\bar{q}_k(i)=g_{k,b}v_k(i)+q_{d,k}(i)$, thus $\sigma^2_{\bar{q}_k}=\sigma^2_{v,k}$. Therefore, for high precision signals, the third and fourth definitions in (80) vanish, and (81) reduces to

$$\begin{aligned}\mathrm{MSD}_k &= (1-\lambda)^2\,\mathrm{vec}^T\left(\mathbf{C}_k\otimes\mathbf{I}_M\right)\\ &\quad\left(\mathbf{I}_{M^2N^2}-\lambda^2(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\right)^{-1}(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\\ &\quad\mathrm{vec}\Big(\mathrm{bdiag}\left\{\mathbf{R}_{x_1}^{-1}\sigma^2_{v,1},\ldots,\mathbf{R}_{x_N}^{-1}\sigma^2_{v,N}\right\}\Big),\end{aligned}\tag{83}$$

which is equal to the theoretical MSD of the standard DRLS. So, as we expected, the MSD performance of DQA-RLS becomes closer to that of the standard DRLS with the increase of the resolution of ADCs.

E. Complexity and Power Consumption

Table III shows the computational complexity of the DQA-RLS algorithm in terms of the number of multiplications and additions at node k per time instant, where n_k is the number of neighbor nodes connected to node k. At each time instant, DQA-RLS performs a few more operations ($O(M+2^b)$) than DRLS, which does not change much the computational complexity that is in the order of $O(M^2)$. Fig. 2a shows a comparison of the computational complexity of the DQA-RLS and DRLS algorithms for different filter lengths in terms of the number of multiplications/divisions and additions/subtractions assuming $n_k=3$ in average for each node k. As we can see, by increasing the filter length the number of operations will increase while the computational complexity of DQA-RLS still remains close to that of DRLS. For instance, for $M=32$ in Fig. 2a, DQA-RLS with low-resolution quantized signals adds 1% extra multiplications/divisions to DRLS with full resolution signals while the number of additions/subtractions operated by DQA-RLS remains very close to that of DRLS. Note that we compute $g_{k,b}$ online since this is more appropriate for non-stationary input data. However, one can compute $\mathbf{G}_{k,b}$ offline if an estimate of $\mathbf{R}_{x_k}$ in (10) is available.

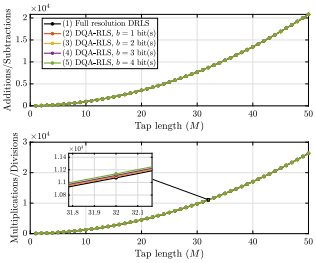
However, the extra complexity in DQA-RLS allows the system to work in an energy-efficient way and enables the algorithm to be robust against variations and imprecise knowledge of $\mathbf{R}_{x_k}$. In order to assess the power savings by low-resolution quantization, let us consider a network with N nodes in which each node uses two ADCs. The power consumption of each ADC is $P_{ADC}(b)=cB2^b$ [41], where B is the bandwidth (related to the sampling rate), b is the number of quantization bits of the ADC, and c is the power consumption per conversion step. Therefore, the total power consumption of the ADCs in the network is

$$P_{ADC,T}(b) = 2NcB2^b \qquad (\text{watts}).\tag{84}$$

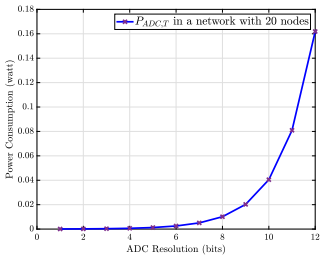
Fig. 2b shows an example of the total power consumption of ADCs in a narrowband IoT (NB-IoT) network running diffusion adaptation consisting of 20 nodes with bandwidth $B=200\,\mathrm{kHz}$ [42] and considering the power consumption per conversion step of each ADC, $c=494$ fJ, as in [43].

TABLE III
COMPUTATIONAL COMPLEXITY PER TIME INSTANT

Task	$+-$	$\times$	$\div$	exp
$\widehat{\sigma}^2_{x_k,Q}(i) = \gamma\widehat{\sigma}^2_{x_k,Q}(i-1) + (1-\gamma)\lvert x_{k,Q}(i)\rvert^2$	2	4	0	0
$\widehat{\sigma}^2_{x_k}(i) = \widehat{\sigma}^2_{x_k,Q}(i) + \widehat{\sigma}^2_{q,k}$	1	0	0	0
$g_{x_k,b}(i) = \frac{1}{\sqrt{\widehat{\sigma}^2_{x_k}(i)}} \sum_{j=0}^{2^b-1} \frac{l_j}{\sqrt{\pi}} \left(e^{-\frac{\tau_j^2}{\widehat{\sigma}^2_{x_k}(i)}} - e^{-\frac{\tau_{j+1}^2}{\widehat{\sigma}^2_{x_k}(i)}} \right)$	$2^b - 1$	$2^{b+1} + 1$	$2^b + 1$	2^b
$\beta_{k,b}(i) = \frac{g^2_{x_k,b}(i)\widehat{\sigma}^2_{x_k}(i)}{g^2_{x_k,b}(i)\widehat{\sigma}^2_{x_k}(i) + \sigma^2_{q,k}}$	1	2	1	0
$\bar{e}_k(i) = d_{k,Q}(i) - \mathbf{w}_k^*(i-1)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)$	$4M$	$4M+2$	0	0
$\mathbf{P}_k(i) = \frac{1}{\lambda}\big(\mathbf{P}_k(i-1) - \frac{\mathbf{P}_k(i-1)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)\beta_{k,b}(i)\mathbf{P}_k(i-1)}{\lambda + \mathbf{x}^*_{k,Q}(i)\beta_{k,b}(i)\mathbf{P}_k(i-1)\mathbf{x}_{k,Q}(i)\beta_{k,b}(i)}\big)$	$8M^2 + 2M - 1$	$10M^2 + 6M + 2$	1	0
$\mathbf{h}_k(i) = \mathbf{h}_k(i-1) + \mathbf{P}_k(i)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)\bar{e}^*_k(i)$	$4M$	$4M+2$	0	0
$\mathbf{w}_k(i) = \sum_{l\in\mathcal{N}_k} a_{l,k}\mathbf{h}_l(i)$	$2n_kM$	$4n_kM$	0	0
Total (DQA-RLS at node k)	$8M^2 + 2^b + 2$ $(2n_k+10)M$	$10M^2 + 2^{b+1} +$ $(4n_k+14)M + 13$	$2^b + 3$	2^b
Total (DRLS [12] at node k)	$8M^2 - 1 +$ $(2n_k+10)M$	$10M^2 +$ $(4n_k+14)M$	1	0



(a)

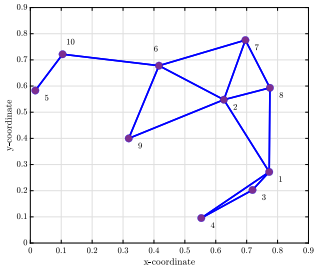


(b)

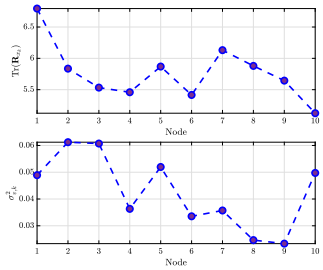
Fig. 2. (a) Number of operations per node versus the filter length for $n_k = 3$. and (b) Power consumption of the ADCs in an adaptive IoT network.

V. SIMULATION RESULTS

In this section, we assess the estimation performance of DQA-RLS for a system identification setup in a network with $N = 10$ nodes. The impulse response of the unknown system has $M = 8$ taps, is generated randomly and normalized to one. We design the ADCs for all nodes with a set of thresholds $\widetilde{\mathcal{T}}_b = \{\tau_1, \ldots, \tau_{2^b-1}\}$ and labels $\mathcal{L}_b = \{l_0, \ldots, l_{2^b-1}\}$ using the Lloyd-Max algorithm [36], [37]. The input signals $\mathbf{x}_k(i)$ at each node are chosen as Gaussian i.i.d. with a covariance matrix $\mathbf{R}_{x_k} = \mathbf{U}_k \,\mathrm{diag}\{\mathbf{s}_k\}\mathbf{U}_k^*$ where $\mathbf{U}_k$ is an $M \times M$ random unitary matrix and $\mathbf{s}_k$ is an $M \times 1$ vector with random entries between 0.5 and 1. The noise samples of each node are drawn from a zero mean white Gaussian process with variance $\sigma^2_{v,k}$. The input regressors and desired signals are quantized with $\widetilde{\mathcal{T}}_b$ and $\mathcal{L}_b$ to generate $\mathbf{x}_{k,Q}(i)$ and $d_{k,Q}(i)$. Fig. 3 plots the network details.



(a) Distributed network structure



(b) Network statistical settings

Fig. 3. A wireless network with $N = 10$ nodes.

The simulated MSD learning curves are obtained by ensemble averaging over 100 independent trials. The steady-state MSD values are obtained by ensemble averaging over 100 independent trials over the last 200 samples. The combining coefficients $a_{l,k}$ are computed by the Metropolis rule [40], $\gamma = 0.9$ and $\lambda = 0.99$. We have compared the proposed DQA-RLS in table II with DRLS [12], DLMS [11], and DQA-LMS [29]. The curves with full resolution DRLS and full resolution DLMS legends refer to the case where the input signals $\mathbf{x}_k(i)$ and desired outputs $d_k(i)$ are not quantized and used as high precision data for the estimation task by DRLS and DLMS algorithms, respectively, whereas other curves are generated with b-bit quantized $\mathbf{x}_{k,Q}(i)$ and $d_{k,Q}(i)$ as the coarsely quantized data.

Figs. 4 and 5 show the global MSD learning curve (average MSD among nodes) and the node-wise steady state MSD values obtained from simulations for DRLS and DQA-RLS using different numbers of bits. Curve 1 shows the standard DRLS performance assuming full resolution ADCs to perform estimation. Curves 2, 4 and 6 show the MSD evolution of the standard DRLS with signals coarsely quantized with b=1, 2 and 3 bits, respectively. Curves 3, 5 and 7 show the MSD performance of the proposed DQA-RLS algorithm that improves the MSD performance for coarsely quantized signals. The performance of the proposed DQA-RLS algorithm is closer to the DRLS while it reduces about 90% of the power consumption related to the ADCs in the network (see Fig. 2b).

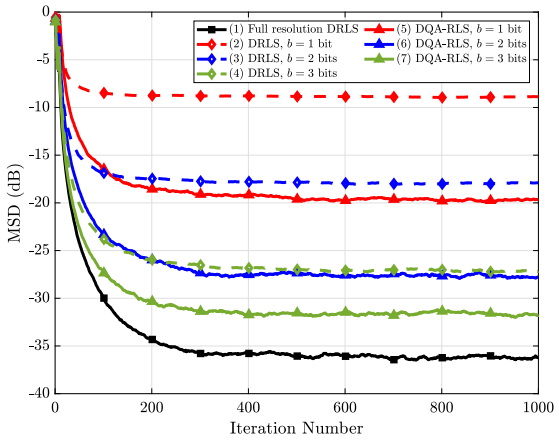


Fig. 4. MSD curves for the DRLS [12] and DQA-RLS algorithms.

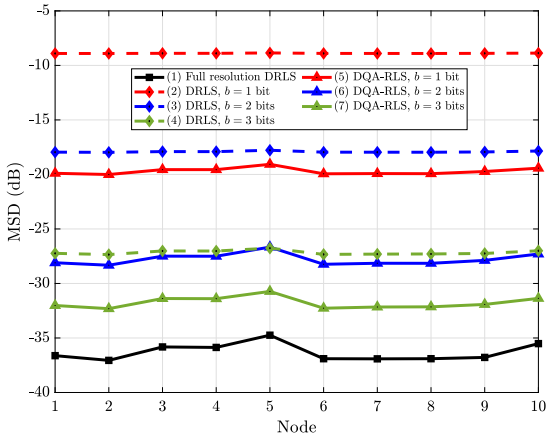


Fig. 5. Steady-state MSD values for the DRLS [12] and DQA-RLS algorithms.

In the next two examples, the MSD performance is evaluated for different signal-to-noise ratios (SNRs) adjusted for the network while nodes are working under different SNRs, i.e., SNR at node k equals the SNR adjusted to the network $\pm 20\,\%$ and the results are shown in Figs. 6 and 7. In particular, Fig. 7 compares the simulation results with those obtained by the analytical expression in (82). The results in Fig. 7 indicate that the theoretical and simulated results agree well especially for $b = 3$ and $b = 4$ bits, and low to moderate values of SNR, confirming the validity of the theoretical development. The node-wise theoretical and experimental MSD values for different nodes for a moderate SNR are compared in Fig. 8 and authenticate the validation of the MSD theoretical expression (81).

In Fig. 9, the MSD learning curves of the proposed DQA-RLS is compared with DQA-LMS [29], standard DRLS [12] and DLMS algorithms [11]. We choose the same step sizes for all agents, i.e., $\mu_k = 0.05$ for DLMS and DQA-LMS algorithms.

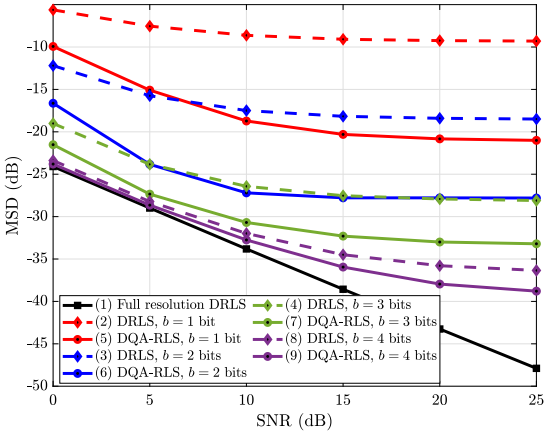


Fig. 6. Steady-state MSD values for the DRLS [12] and DQA-RLS algorithms for different SNR values.

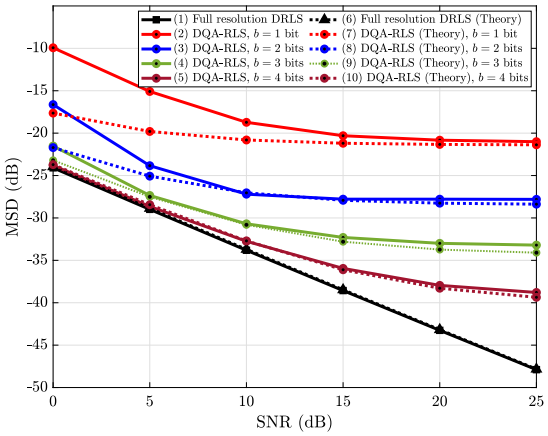


Fig. 7. Steady-state and theoretical MSD values for the DRLS [12] and DQA-RLS algorithms for different SNR values.

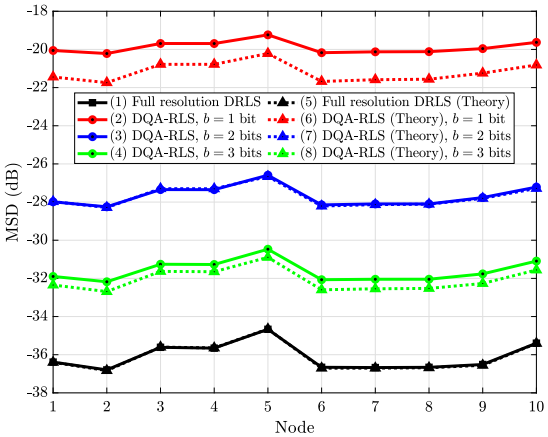


Fig. 8. Node-wise Steady-state and theoretical MSD values for the DRLS [12] and DQA-RLS algorithms.

It can be seen that the DQA-RLS algorithm improves the estimation performance of the DQA-LMS algorithm while both outperform the standard DRLS and DLMS, respectively, with coarsely quantized signals. According to curves (5) and (6) in Fig. 9, in applications in which computational complexity is not a bottleneck, one can use DQA-RLS with 2-bit quantization to achieve the estimation performance of full resolution DLMS and save a large amount of energy consumption of the ADCs.

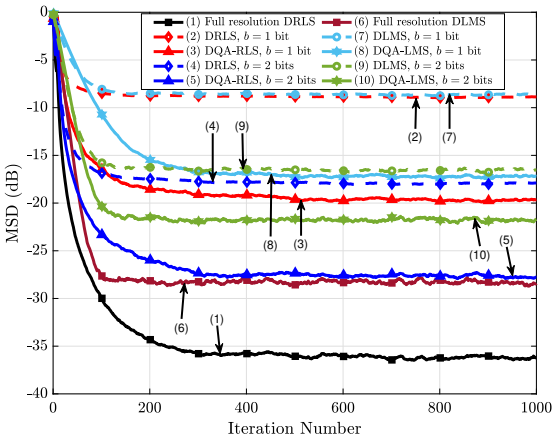


Fig. 9. MSD curves for the DRLS [12], DLMS [11], DQA-RLS and DQA-LMS [29] algorithms.

VI. Conclusion

In this paper, we have proposed an energy-efficient framework for distributed learning and developed the DQA-RLS algorithm along with bias compensation strategies for IoT networks. The DQA-RLS algorithm has comparable computational complexity to the standard DRLS algorithm while it reduces the power consumption of the ADCs in the network by more than 90%. We have also carried out a statistical analysis of the DQA-RLS algorithm along with a study of the power consumption of the proposed and existing techniques. The derived analytical expressions have been shown to accurately predict the MSD of the DQA-RLS algorithm. Numerical results have shown the excellent performance of DQA-RLS algorithm as compared to the standard DRLS algorithm for coarsely quantized signals.

Appendix

In this Appendix, we obtain the theoretical expressions for $\mathbf{R}_{x_k,Q}$, $\mathbf{R}_{q_x,k}$, and $\sigma^2_{\bar{q}_k}$ based on $\mathbf{R}_{x_k}$, $\sigma^2_{v,k}$ for the designed thresholds and labels. We consider $\sigma_{\alpha(i)\beta(j)} = \mathbb{E}[\alpha(i)\beta^*(j)]$ and $\sigma^2_{\alpha}(i) = \mathbb{E}[\alpha(i)\alpha^*(i)]$. From (14) and (8), $\mathbf{R}_{q_x,k}$ can be obtained as follows:

$$\mathbf{R}_{q_x,k} = \mathbf{R}_{x_k,Q} - \mathbf{G}_{k,b}\mathbf{R}_{x_k}\mathbf{G}_{k,b}, \tag{85}$$

where $\mathbf{G}_{k,b}$ is computed as in (10) with $\mathbf{R}_{x_k}$. We evaluate the covariance matrix $\mathbf{R}_{x_k,Q}$ whose diagonal entries are given by

$$\begin{aligned}
[\mathbf{R}_{x_k,Q}]_{m,m} &= \mathbb{E}[|\mathbf{x}_{k,Q}(m)|^2] = h\sum_{j=0}^{2^b-1} l_j^2 \mathbb{P}[\mathbf{x}_{k,Q}(m) = l_j] \\
&= h\sum_{j=0}^{2^b-1} l_j^2 \mathbb{P}[\tau_j \le \mathbf{x}_k(m) = l_j < \tau_{j+1}] \\
&= h\sum_{j=0}^{2^b-1} l_j^2 \Big(\Phi\Big(\frac{\sqrt{h}\tau_{j+1}^2}{\sigma_{x_k}(m)}\Big) - \Phi\Big(\frac{\sqrt{h}\tau_j^2}{\sigma_{x_k}(m)}\Big)\Big),
\end{aligned} \tag{86}$$

where $\sigma^2_{x_k}(m) = \mathbb{E}[|\mathbf{x}_k(m)|^2] = [\mathbf{R}_{x_k}]_{m,m}$, $\Phi\big(\,\cdot\,\big)$ refers to the cumulative distribution function and the variable $h = 1$ for real data and $h = 2$ for complex data. The off-diagonal entries of $\mathbf{R}_{x_k,Q}$ for $1 < m, n < M$ and $m \neq n$ are given by

$$\begin{aligned}[\mathbf{R}_{x_k,Q}]_{m,n} &= \mathbb{E}[\mathbf{x}_{k,Q}(m)\mathbf{x}^*_{k,Q}(n)] \\ &= \sum_{j=0}^{2^b-1}\sum_{p=0}^{2^b-1} l_j l_p \mathbb{P}[\mathbf{x}_{k,Q}(m) = l_j, \mathbf{x}_{k,Q}(n) = l_p] \\ &= \sum_{j=0}^{2^b-1}\sum_{p=0}^{2^b-1} l_j l_p \mathbb{P}[\tau_j \leq \mathbf{x}_k(m) < \tau_{j+1}, \\ &\qquad \tau_p \leq \mathbf{x}_k(n) < \tau_{p+1}].\end{aligned} \tag{87}$$

Unfortunately, (87) does not have a known closed-form expression and hence, has to be evaluated using numerical methods [22]. However, in what follows, we shall present a closed-form approximation for the off-diagonal entries of $\mathbf{R}_{x_k,Q}$, following [23]. Let us rewrite (14) as follows:

$$\mathbf{x}_{k,Q}(i) = \mathbf{G}_{k,b}\mathbf{x}_k(i) + \mathbf{q}_{x,k}(i) = \mathbf{x}_k(i) + \boldsymbol{\epsilon}_k(i), \tag{88}$$

where $\boldsymbol{\epsilon}_k(i)$ is the quantization error which by definition is the difference between an input value and its quantized value. Each quantization process is assigned a distortion factor $\rho_{q,b}(i)$ to indicate the relative amount of quantization error generated, which is defined as follows:

$$\rho_{q,b}(i) = \frac{\sigma^2_{\epsilon_k}(i)}{\sigma^2_{x_k}(i)}, \tag{89}$$

where $\sigma^2_{x_k}(i)$ is the variance of the input and the distortion factor $\rho_{q,b}(i)$ depends on the number of quantization bits b, the quantizer type (uniform or non-uniform) and the probability density function of $\mathbf{x}_k(i)$ [23]. For Gaussian inputs and a scalar non-uniform quantizer, e.g., Lloyd-Max quantizer, the distortion factor $\rho_{q,b}(i) = \rho_{q,b}$ can be obtained from Table IV in [37] for different b and asymptotically approximated by $\rho_{q,b} = \frac{\pi\sqrt{3}}{2}2^{-2b}$ for $b > 5$ [44]. Based on this, we obtain an approximation of $\mathbf{R}_{x_k,Q}$ as follows [23]:

$$\widehat{\mathbf{R}}_{x_k,Q} \approx (1 - \rho_{q,b})\big(\mathbf{R}_{x_k} - \rho_{q,b}\,\mathrm{nondiag}\{\mathbf{R}_{x_k}\}\big), \tag{90}$$

where the operator $\mathrm{nondiag}\{\mathbf{A}\} = \mathbf{A} - \mathrm{diag}\{\mathbf{A}\}$. Using this, we approximate the off-diagonal entries $[\mathbf{R}_{x_k,Q}]_{m,n}$ as follows:

$$\mathrm{nondiag}\{\mathbf{R}_{x_k,Q}\} = \widehat{\mathbf{R}}_{x_k,Q} - \mathrm{diag}\{\widehat{\mathbf{R}}_{x_k,Q}\}. \tag{91}$$

Note that we compute the diagonal elements of $\mathbf{R}_{x_k,Q}$ directly using (86) instead of using the approximation in (90), to get an accurate expression for $[\mathbf{R}_{x_k,Q}]_{m,m}$, and approximate the off-diagonal entries $[\mathbf{R}_{x_k,Q}]_{m,n}$ from (91).

We now find a closed-form expression for $\sigma^2_{\bar{q}_k}$, where $\bar{q}_k(i) = g_{k,b}v_k(i) + q_{d,k}(i)$ and

$$\sigma^2_{\bar{q}_k} = \mathbb{E}[\bar{q}_k(i)\bar{q}^*_k(i)] = g^2_{k,b}\sigma^2_{v,k} + \mathbb{E}[q_{d,k}(i)q^*_{d,k}(i)]. \tag{92}$$

From (15), we have

$$\sigma^2_{q_{d,k}} = \mathbb{E}[q_{d,k}(i)q^*_{d,k}(i)] = \sigma^2_{d_k,Q} - g^2_{k,b}\sigma^2_{d,k}, \tag{93}$$

where considering the data model in (1), $\sigma^2_{d,k}$ is given by

$$\sigma^2_{d,k} = \mathbb{E}[d_k(i)d^*_k(i)] = \mathbf{w}^*_o\mathbf{R}_{x_k}\mathbf{w}_o + \sigma^2_{v,k}. \tag{94}$$

Finally, using the same evaluation as in (86), the variance of $d_{k,Q}$ is given by

$$\begin{aligned}\sigma^2_{d_k,Q} &= \mathbb{E}[d_{k,Q}(i)d^*_{k,Q}(i)] = h\sum_{j=0}^{2^b-1} l_j^2 \mathbb{P}[\mathbf{x}_{k,Q}(m) = l_j] \\ &= h\sum_{j=0}^{2^b-1} l_j^2 \mathbb{P}[\tau_j \leq d_k = l_j < \tau_{j+1}] \\ &= h\sum_{j=0}^{2^b-1} l_j^2 \bigg(\Phi\Big(\frac{\sqrt{h}\tau^2_{j+1}}{\sigma_{d,k}}\Big) - \Phi\Big(\frac{\sqrt{h}\tau^2_{j}}{\sigma_{d,k}}\Big)\bigg).\end{aligned} \tag{95}$$

TABLE IV
DISTORTION FACTOR $\rho_{q,b}$ FOR DIFFERENT ADC RESOLUTIONS b [37]

b	1	2	3	4	5
$\rho_{q,b}$	0.3634	0.1175	0.03454	0.009497	0.002499

REFERENCES

[1] J. B. Predd, S. B. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[2] L. A. Rossi, B. Krishnamachari, and C. C. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.* IEEE, 2004, pp. 460–469.

[3] M. M. Rana, W. Xiang, and E. Wang, "IoT-based state estimation for microgrids," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1345–1346, 2018.

[4] Yifei Zou, Minghui Xu, Hao Sheng, Xiaoshuang Xing, Yicheng Xu, and Yong Zhang, "Crowd density computation and diffusion via internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8111–8121, 2020.

[5] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.

[6] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2014.

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

[8] C. Ibars, M. Navarro, and L. Giupponi, "Distributed demand management in smart grid with a congestion game," in *2010 First IEEE International Conference on Smart Grid Communications*. IEEE, 2010, pp. 495–500.

[9] G. B. Giannakis, V. Kekatos, N. Gatsis, S. J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 107–128, 2013.

[10] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

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

[12] F. S. 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, 2008.

[13] S. Xu, R. C. de Lamare, and H. V. Poor, "Distributed estimation over sensor networks based on distributed conjugate gradient strategies," *IET Signal Processing*, vol. 10, no. 3, pp. 291–301, 2016.

[14] T. G. Miller, S. Xu, R. C. de Lamare, V. H. Nascimento, and Y. Zakharov, "Sparsity-aware distributed conjugate gradient algorithms for parameter estimation over sensor networks," in *2015 49th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2015, pp. 1556–1560.

[15] Zhen Qin, Jun Tao, and Yili Xia, "A proportionate recursive least squares algorithm and its performance analysis," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020.

[16] S. Xu, R. C. de Lamare, and H. V. Poor, "Adaptive link selection algorithms for distributed estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–22, 2015.

[17] Feng Chen, Limei Hu, Pengfei Liu, and Minyu Feng, "A robust diffusion estimation algorithm for asynchronous networks in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9103–9115, 2020.

[18] Y. Yu, H. Zhao, R. C. de Lamare, Y. Zakharov, and L. Lu, "Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1566–1581, 2019.

[19] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5212–5224, 2011.

[20] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on selected areas in communications*, vol. 17, no. 4, pp. 539–550, 1999.

[21] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 4038–4051, 2017.

[22] S. Jacobsson, G. Durisi, M. Coldrey, and C. Studer, "Linear precoding with low-resolution dacs for massive mu-mimo-ofdm downlink," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1595–1609, 2019.

[23] A. Mezghani, M.-S. Khoufi, and J. A. Nossek, "A modified MMSE receiver for quantized MIMO systems," *Proc. ITG/IEEE WSA, Vienna, Austria*, pp. 1–5, 2007.

[24] Z. Shao, L. Landau, and R. C. de Lamare, "Adaptive RLS channel estimation and sic for large-scale antenna systems with 1-bit adcs," in *WSA 2018; 22nd International ITG Workshop on Smart Antennas*. VDE, 2018, pp. 1–4.

[25] R. Arablouei, K. Doğançay, S. Werner, and Y. F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE transactions on signal processing*, vol. 62, no. 14, pp. 3510–3522, 2014.

[26] R. Arablouei, S. Werner, Y. Huang, and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, 2013.

[27] Wanyu Lin, Jiannong Cao, and Xuefeng Liu, "E^3: Towards energy-efficient distributed least squares estimation in sensor networks," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. IEEE, 2014, pp. 21–30.

[28] M. Alioto and M. Shahghasemi, "The internet of things on its edge: Trends toward its tipping point," *IEEE Consumer Electronics Magazine*, vol. 7, no. 1, pp. 77–87, 2017.

[29] A. Danaee, R. C. de Lamare, and V. H. Nascimento, "Energy-efficient distributed learning with coarsely quantized signals," *IEEE Signal Processing Letters*, vol. 28, pp. 329–333, 2021.

[30] A. Danaee, R. C. de Lamare, and V. H. Nascimento, "Energy-efficient distributed learning with adaptive bias compensation for coarsely quantized signals," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2021, pp. 61–65.

[31] Ali H Sayed, *Fundamentals of adaptive filtering*, John Wiley & Sons, 2003.

[32] Yuriy V Zakharov, Vitor H Nascimento, Rodrigo C De Lamare, and Fernando Goncalves De Almeida Neto, "Low-complexity DCD-based sparse recovery algorithms," *IEEE Access*, vol. 5, pp. 12737–12750, 2017.

[33] Yi Yu, Lu Lu, Zongsheng Zheng, Wenyuan Wang, Yuriy Zakharov, and Rodrigo C de Lamare, "DCD-based recursive adaptive algorithms robust against impulsive noise," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 7, pp. 1359–1363, 2019.

[34] H. E. Rowe, "Memoryless nonlinearities with Gaussian inputs: Elementary results," *The BELL system technical Journal*, vol. 61, no. 7, pp. 1519–1525, 1982.

[35] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted gaussian signals," Tech. Rep. 216, Research Laboratory of Electronics, Massachusetts Institute of Technology, 1952.

[36] S. Lloyd, "Least squares quantization in PCM," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[37] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, 1960.

[38] Paulo SR Diniz et al., *Adaptive filtering*, vol. 4, Springer, 1997.

[39] V. H Nascimento and M. T. M. Silva, "Adaptive filters," in *Academic Press Library in Signal Processing*, vol. 1, pp. 619–761. Elsevier, 2014.

[40] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[41] O. Orhan, E. Erkip, and S. Rangan, "Low power analog-to-digital conversion in millimeter wave systems: Impact of resolution and bandwidth on performance," in *2015 Information Theory and Applications Workshop (ITA)*. IEEE, 2015, pp. 191–198.

[42] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," in *2016 IEEE wireless communications and networking conference*. IEEE, 2016, pp. 1–5.

[43] H. Chung, A. Rylyakov, Z. T. Deniz, J. Bulzacchelli, G Wei, and D. Friedman, "A 7.5-GS/s 3.8-ENOB 52-mW flash ADC with clock duty cycle control in 65nm CMOS," in *2009 Symposium on VLSI Circuits*. IEEE, 2009, pp. 268–269.

[44] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, vol. 159, Springer Science & Business Media, 2012.