Pontifícia Universidade Católica
do Rio de Janeiro

Alireza Danaee

Signal Processing Algorithms for Energy-Efficient Distributed Learning

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica.

Advisor : Prof. Rodrigo Caiado de Lamare
Co-advisor: Prof. Vítor Heloiz Nascimento

Rio de Janeiro
September 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Alireza Danaee

Signal Processing Algorithms for Energy-Efficient Distributed Learning

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the undersigned Examination Committee.

Prof. Rodrigo Caiado de Lamare
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Vítor Heloiz Nascimento
Co-advisor
Universidade de São Paulo – USP

Prof. Elena Veronica Belmega
Université Gustave Eiffel – Paris

Prof. Yuriy Zakharov
University of York – York

Prof. Cássio Guimarães Lopes
Universidade de São Paulo – USP

Prof. Alan Conci Kubrusly
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Lukas Tobias Nepomuk Landau
Departamento de Engenharia Elétrica – PUC-Rio

Dr. Silvio Fernando Bernardes Pinto
Departamento de Engenharia Elétrica – PUC-Rio

Rio de Janeiro, September the 21st, 2022

Alireza Danaee

The author received the B.Sc. and M.Sc degrees in Electrical Engineering from the University of Kurdistan, Sanandaj, in 2008, and the Shahid Rajaee Teacher Training University, Tehran, in 2013, respectively.

Bibliographic data

Danaee, Alireza

Signal Processing Algorithms for Energy-Efficient Distributed Learning / Alireza Danaee; advisor: Rodrigo Caiado de Lamare; co-advisor: Vítor Heloiz Nascimento. – Rio de janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2022.

113 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Processamento de Sinais, Automação e Robótica – Teses. 3. algoritmos adaptativos. 4. quantização severamente. 5. aprendizagem distribuída. 6. aprendizado federativo. 7. processamento de sinais com eficiência energética. I. de Lamare, Rodrigo C.. II. Nascimento, Vítor H.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

This thesis is dedicated to my parents
for their endless love, support and encouragement.

Acknowledgments

First of all, I would like to thank my mother, my siblings, and my father (in memoriam), for their support and encouragement since my undergraduate studies.

I am grateful to have Prof. Rodrigo C. de Lamare as my advisor and Prof. Vítor H. Nascimento as my co-advisor during these four years. This work would not have been possible without their suggestions, guidance, and technical advice.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). I would like to thank CAPES and CNPq for funding my research.

I would also like to thank all my colleagues and professors in the Centro de Estudos de Telecomunicações (CETUC/ PUC-Rio), but especially Prof. Lukas T. N. Landau, which always helped with new ideas and my friends Roberto Brauer Di Renna, Zhichao Shao, Erico Lopes, Diana M. V. Melo, André R Flores, Saeed Mashdour, and Robert Mota who contributed to a friendly environment at CETUC.

Moreover, I would like to thank the members of my dissertation committee, Prof. Yuriy Zakharov, Prof. Cassio Guimaraes Lopes, Prof. E. Veronica Belmega, Prof. Alan Kubrusly, Prof. Lukas T. N. Landau, and Dr. Silvio Fernando Bernardes Pinto for their time and effort dedicated to my thesis.

Finally, I would like to thank Sâmela for her love and support, for always being on my side helping me navigate my way.

Abstract

Danaee, Alireza; de Lamare, Rodrigo C. (Advisor); Nascimento, Vítor H. (Co-Advisor). **Signal Processing Algorithms for Energy-Efficient Distributed Learning**. Rio de Janeiro, 2022. 113p. PhD Dissertation – Departament of Electrical Engineering, Pontifícia Universidade Católica do Rio de Janeiro.

Internet of Things (IoT) networks include smart devices that contain many sensors that allow them to interact with the physical world, collecting and processing streaming data in real time. The total energy-consumption and cost of these sensors affect the energy-consumption and the cost of IoT devices. The type of sensor determines the accuracy of the analog interface and the resolution of the analog-to-digital converters (ADCs). The ADC resolution requirement has a trade-off between sensing performance and energy consumption since the energy consumption of ADCs strongly depends on the number of bits used to represent digital samples.

In this thesis, we present an energy-efficient distributed learning framework using coarsely quantized signals for IoT networks. In particular, we develop a distributed quantization-aware least-mean square (DQA-LMS) and a distributed quantization-aware recursive least-squares (DQA-RLS) algorithms 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 algorithms. We then carry out a statistical analysis of the proposed algorithms along with a computational complexity evaluation of the proposed and existing techniques. Numerical results assess the distributed quantization-aware algorithms against existing techniques for distributed parameter estimation where IoT devices operate in a peer-to-peer mode.

We also introduce an energy-efficient federated learning framework using coarsely quantized signals for IoT networks, where IoT devices exchange their estimates with a server. We then develop the quantization-aware federated averaging LMS (QA-FedAvg-LMS) algorithm to perform parameter estimation at the clients and servers. Furthermore, we devise a bias compensation strategy for QA-FedAvg-LMS, carry out its statistical analysis, and assess its performance against existing techniques with numerical results.

Keywords

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

Resumo

Danaee, Alireza; de Lamare, Rodrigo C.; Nascimento, Vítor H.. **Técnicas de Processamento de Sinais para Aprendizagem Distribuída com Eficiência Energética**. Rio de Janeiro, 2022. 113p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

As redes da Internet das Coisas (IdC) incluem dispositivos inteligentes que contêm muitos sensores que permitem interagir com o mundo físico, coletando e processando dados de streaming em tempo real. O consumo total de energia e o custo desses sensores afetam o consumo de energia e o custo dos dispositivos IdC. O tipo de sensor determina a precisão da interface analógica e a resolução dos conversores analógico-digital (ADCs). A resolução dos ADCs tem um compromisso entre a precisão de inferência e o consumo de energia, uma vez que o consumo de energia dos ADCs depende do número de bits usados para representar amostras digitais.

Nesta tese, apresentamos um esquema de aprendizado distribuído com eficiência energética usando sinais quantizados para redes da IdC. Em particular, desenvolvemos algoritmos de gradiente estocástico com reconhecimento de quantização distribuído (DQA-LMS) e de mínimos quadrados recursivos com reconhecimento de quantização distribuído (DQA-RLS) que podem aprender parâmetros de maneira eficiente em energia usando sinais quantizados com poucos bits, exigindo um baixo custo computacional. Além disso, desenvolvemos uma estratégia de compensação de viés para melhorar ainda mais o desempenho dos algoritmos propostos. Uma análise estatística dos algoritmos propostos juntamente com uma avaliação da complexidade computacional das técnicas propostas e existentes é realizada. Os resultados numéricos avaliam os algoritmos com reconhecimento de quantização distribuída em relação às técnicas existentes para uma tarefa de estimação de parâmetros em que os dispositivos IdC operam em um modo ponto a ponto.

Também apresentamos um esquema de aprendizado federativo com eficiência energética usando sinais quantizados para redes de IdC. Desenvolvemos o algoritmo federated averaging LMS (QA-FedAvg-LMS) com reconhecimento de quantização para redes IdC estruturadas por configuração de aprendizado federativo em que os dispositivos IdC trocam suas estimativas com um servidor. Uma estratégia de compensação de viés para QA-FedAvg-LMS é proposta junto com sua análise estatística e a avaliação de desempenho em relação às técnicas existentes com resultados numéricos.

Palavras-chave

algoritmos adaptativos quantização severamente aprendizagem distribuída aprendizado federativo processamento de sinais com eficiência energética

Table of contents

List of figures

List of tables

List of Abreviations

ADC	–	Analog-to-digital converter
AGC	–	Automatic Gain Control
AWGN	–	Additive White Gaussian Noise
CDF	–	Cumulative Distribution Function
EMSE	–	Excess Mean Squared Error
FedAvg	–	Federated Averaging
IoT	–	Internet of Things
LMS	–	Least Mean Square
DLMS	–	Distributed Least Mean square
DRLS	–	Distributed Recursive Least Squares
MIMO	–	Multiple Input Multiple Output
MSD	–	Mean Squared Deviation
MSE	–	Mean Squared Error
PDF	–	Probability Density Function
QA	–	Quantization-Aware
RLS	–	Recursive Least Squares
WSN	–	Wireless Sensor Network

1
Introduction

This chapter presents the research background and the motivations of this thesis. The main contributions and the structure of this thesis are then provided to readers to access the current state of the art. Moreover, some basic notations used throughout the thesis are introduced. The last section makes a list of publications during the period of development of this thesis.

1.1
Motivation and Prior Works

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, 6, 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 collect and process their local information, and then transmit 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.

The goal of federated learning [10, 11] is to learn a global statistical model from data stored at tens to millions of devices subject to storing the data locally at devices and only communicating the intermediate updates generated by devices to the server. In this context, IoT networks include smart devices such as mobile phones, smart watches, and autonomous vehicles which are generating new data every day [12]. Federated learning offers IoT networks local data storage at devices and transfers network computation to the devices

due to the growing computational capability of these devices. This can mitigate concerns over transmitting private information.

Prior work on distributed signal processing techniques has studied protocols for exchanging information [13, 14, 15], adaptive learning algorithms [16], the exploitation of sparse measurements [17, 18], topology adaptation [19, 20], and robust techniques against interference and noise [21, 22]. 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 [23]. 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 [24, 25, 26, 27].

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 [28, 29, 30] and federated learning [11, 31]. 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 [32, 33]. They integrate various sensors such as temperature,

humidity, accelerometer, gyroscope, magnetometer, altimeter, heart rate, light, microphone, camera, battery monitor, infrared proximity, gas, ultraviolet, capacitive sensors. The total energy-consumption and cost of these sensors affect the energy-consumption and the cost of IoT devices. The type of sensor determines the accuracy of the analog interface and the resolution of the ADC. The ADC resolution requirement varies greatly with the sensing application, ranging from 6 to 16 bits (see [34] Table 1), and has a trade-off between sensing performance and energy consumption since the energy consumption of ADCs strongly depends on the number of bits used to represent digital samples [23]. 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.

1.2 Contributions

The objective of this thesis is to develop energy-efficient signal processing techniques for distributed adaptive learning and federated learning with application to IoT networks using coarsely quantized signals. In particular, our contributions can be summarized as:

- We develop distributed quantization-aware least-mean square (DQA-LMS) and distributed quantization-aware recursive least-squares (DQA-RLS) algorithms that can learn parameters in an energy-efficient fashion using signals quantized with few bits while requiring a low computational cost. Moreover, we develop closed-form and adaptive bias compensation strategies to further improve the performance of the proposed algorithms. We then carry out a statistical analysis of the proposed algorithms along with a computational complexity evaluation of the proposed and existing techniques. Numerical results compare the distributed quantization-aware algorithms against existing techniques for a distributed parameter estimation task where IoT devices operate in a peer-to-peer mode.

- We also develop an energy-efficient federated learning framework using coarsely quantized measured data for IoT networks. In particular, we present a quantization-aware Federated Averaging Least Mean Square (QA-FedAvg-LMS) algorithm that can learn parameters in an energy-efficient fashion using measurements quantized with few bits, and devise a bias compensation strategy to further improve the performance of the proposed QA-FedAvg-LMS algorithm. We carry out a statistical analysis of the proposed QA-FedAvg-LMS algorithm. Simulations compare the QA-FedAvg-LMS algorithm against existing techniques for a parameter estimation task in a scenario where IoT devices operate with federated learning.

1.3 Outline

The rest of this thesis is organized as follows:

- Chapter 2 gives technical background on this thesis. In the first section, basic concepts of adaptive distributed networks are presented, which includes the model of the network, the structure of nodes, and the communication strategy among nodes and the energy consumption of ADCs in the adaptive distributed networks is demonstrated with an example. In Section 2, the model of federated learning for IoT networks is presented, where the nodes are orchestrated by a server. The concept of signal decomposition, which allows to deal with nonlinearities like the distortion generated by ADCs is presented in Section 3. The design of ADCs in the network is also detailed in section 3.
- Chapter 3 presents the distributed quantization-aware LMS (DQA-LMS) algorithm. In the first section, the derivation of DQA-LMS is presented. The mean performance analysis, the bias compensation term, the adaptive bias compensation strategy and the mean square performance analysis of DQA-LMS are given in Sections 2 to 5. In Section 6, the computational

complexity is evaluated. In Section 7, the results of simulations are shown and discussed.

- Chapter 4 presents the distributed quantization-aware RLS (DQA-RLS) algorithm. In the first section, the derivation of the DQA-RLS algorithm is detailed. In Section 2, the definitions and assumptions that are used for the performance analysis of the RLS-type algorithms are presented. The mean performance analysis, the bias compensation term, and the mean square performance analysis of DQA-RLS are given in Sections 3 to 5. In Section 6, the computational complexity is evaluated. In Section 7, the results of simulations are shown and discussed.
- Chapter 5 presents the quantization-aware Federated Averaging Least Mean Square (QA-FedAvg-LMS) algorithm. In the first section, the derivation of QA-FedAvg-LMS algorithm is detailed. The mean performance analysis, the bias compensation term, and the mean square performance analysis of QA-FedAvg-LMS are given in Sections 2 to 4. In Section 5, the computational complexity is computed. In Section 6, the results of simulations are shown and discussed.
- Chapter 6 presents the conclusions of this thesis and discusses the obtained results, future directions and research opportunities.

1.4 Notation

Throughout this thesis, we use the following notation:

- We show scalars, vectors and matrices with lowercase, boldface lowercase and boldface uppercase letters, respectively.
- The transpose of a vector $\mathbf{a}$ and a matrix $\mathbf{A}$ is denoted by $\mathbf{a}^T$ and $\mathbf{A}^T$, respectively.
- The complex conjugate of a scalar a, the complex conjugate transpose of vector $\mathbf{a}$, and the complex conjugate transpose (Hermitian transpose)

of a matrix $\mathbf{A}$ are shown by a^*, $\mathbf{a}^*$, and $\mathbf{A}^*$, respectively, i.e., for real numbers x and y, and the imaginary unit i, we have

$$
\begin{aligned}
&a = x + iy,\ a^* = x - iy \\
&\mathbf{a} = [a_1, a_2, \dots, a_n],\ \mathbf{a}^* = [a_1^*, a_2^*, \dots, a_n^*]^T \\
&\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},\ \mathbf{A}^* = \begin{bmatrix} a_{11}^* & a_{21}^* & \cdots & a_{n1}^* \\ a_{12}^* & a_{22}^* & \cdots & a_{n2}^* \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n}^* & a_{2n}^* & \cdots & a_{nn}^* \end{bmatrix}
\end{aligned} \tag{1-1}
$$

- $\mathrm{Tr}(\mathbf{A})$ denotes the trace of a matrix $\mathbf{A}$. $\mathbf{A} \otimes \mathbf{B}$ represents the Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$.
- $\mathrm{vec}(A)$ represents the vectorization operation of $\mathbf{A}$, i.e., for $\mathbf{A}$ given by (1-1)

$$
\mathrm{vec}(A) = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \\ a_{12} \\ \vdots \\ a_{nn} \end{bmatrix}. \tag{1-2}
$$

- $\|\mathbf{a}\|$ is the Euclidean norm of a vector $\mathbf{a}$. $\|\mathbf{a}\|_{\mathbf{Z}}^2 = \mathbf{a}^*\mathbf{Z}\mathbf{a}$ is the $\mathbf{Z}$-Weighted Euclidean norm of a column vector $\mathbf{a}$. $\|\mathbf{b}\|_{\mathbf{Z}}^2 = \mathbf{b}\mathbf{Z}\mathbf{b}^*$ denotes the $\mathbf{Z}$-Weighted Euclidean norm of a row vector $\mathbf{b}$.
- $\mathrm{col}\{...\}$ stacks its arguments column-wise as follows:

$$
\mathrm{col}\{a, b, c\} = \begin{bmatrix} a \\ b \\ c \end{bmatrix},\ \mathrm{col}\{\mathbf{a}, \mathbf{b}, \mathbf{c}\} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix},\ \mathrm{col}\{\mathbf{A}, \mathbf{B}, \mathbf{C}\} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{bmatrix}. \tag{1-3}
$$

- diag($\mathbf{A}$) creates a row vector with the diagonal entries of $\mathbf{A}$, i.e., for $\mathbf{A}$ given by (1-1)

$$\text{diag}(\mathbf{A}) = [a_{11}, a_{22}, \ldots, a_{nn}] \tag{1-4}$$

- diag{...} creates a diagonal matrix with the elements of a vector or the diagonal elements of a matrix, i,e., for $\mathbf{a}$ and $\mathbf{A}$ given by (1-1)

$$\text{diag}\{\mathbf{a}\} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \end{bmatrix}, \text{diag}\{\mathbf{A}\} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}. \tag{1-5}$$

- nondiag{$\mathbf{A}$} replaces the diagonal elements of $\mathbf{A}$ with zero, i.e., nondiag{$\mathbf{A}$} = $\mathbf{A}$ − diag{$\mathbf{A}$}.

- blockdiag{...} creates a block diagonal matrix such that the main-diagonal blocks are the entries inside {...}.

- $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. The all-one and all-zero $M \times 1$ column vectors are shown by $\mathbb{1}_M$ and $\mathbf{0}_M$, respectively.

- $\mathbf{x} \sim \mathcal{CN}(\mathbf{m}, \mathbf{R}_x)$ denotes that $\mathbf{x}$ is a complex Gaussian signal with mean $\mathbf{m}$ and covariance matrix $\mathbf{R}_x$ whereas $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{R}_x)$ denotes a real Gaussian signal $\mathbf{x}$ with mean $\mathbf{m}$ and covariance matrix $\mathbf{R}_x$.

- $\mathbb{E}[.]$ denotes the expectation operator.

- $\rho(\mathbf{A})$ represents the spectral radius of a given square matrix $\mathbf{A}$, that is, the maximum of the absolute values of its eigenvalues.

- We use the mathematical accents (diacritical marks) as follows: $\widehat{x}$ is an estimate of a scalar x. $\overline{x}$ denotes the quantized version of a scalar x. These accents are used in the same way for vectors and matrices.

- $Q(.)$ denotes the quantization operator, i.e., $\overline{\mathbf{x}} = Q(\mathbf{x})$ that is performed element-wise.

1.5 Publication List

Some of the results in this thesis have been published, or are under review for publications. The list of publications is presented in the following:

Journal papers:

- Danaee, A.; de Lamare, R. C.; Nascimento, V. H., **Distributed Quantization-Aware RLS Learning with Bias Compensation and Coarsely Quantized Signals**, IEEE Transactions on Signal Processing, vol. 70, pp.3441-3455, 2022.
- Danaee, A.; de Lamare, R. C.; Nascimento, V. H., **Energy-efficient distributed learning with coarsely quantized signals**, IEEE Signal Processing Letters, vol. 28, pp.329-333, 2021.
- Danaee, A.; de Lamare, R. C.; Nascimento, V. H., **Quantization-Aware Federated Learning with Bias Compensation and Coarsely Quantized Measurements**, under preparation.

Conference papers:

- Danaee, A.; de Lamare, R. C.; Nascimento, V. H., **Quantization-Aware Federated Learning with Coarsely Quantized Measurements**, In: (to appear) European Signal Processing Conference (EUSIPCO) 2022.
- Danaee, A.; de Lamare, R. C.; Nascimento, V. H., **Energy-Efficient Distributed Learning with Adaptive Bias Compensation for Coarsely Quantized Signals**, In: IEEE Statistical Signal Processing Workshop 2021.
- Danaee, A.; de Lamare, R. C.; Nascimento, V. H., **Energy-Efficient Distributed Recursive Least Squares Learning with Coarsely Quantized Signals**, In: 2020 54th Asilomar Conference on Signals, Systems, and Computers, pp.1533-1537, 2020.

2
Distributed Learning and Problem Statement

In this section, we present the fundamentals of distributed and federated learning and state the fundamental problem that we are interested in this thesis using simple mathematical models with the help of linear algebra tools. We start this exposition by detailing the signal model of the proposed distributed network with low-resolution ADCs. Then, we describe two common structures for IoT networks, i.e., adaptive distributed networks and federated learning. In adaptive networks, nodes are connected within their neighborhood and cooperate with neighbors to complete a task whereas in federated learning setup, nodes are connected to a server and complete a task with the supervision of the server. 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. We then show the design of ADCs in such networks.

2.1
Adaptive Distributed Networks

Adaptive networks are connected networks with N agents as illustrated in Figure 2.1. In a connected network, there always exists at least one path connecting any two agents: the agents may be connected directly by an edge if they are neighbors, or they may be connected by a path that passes through other intermediate agents.

- A network of size N is generally represented by a graph containing N vertices, which we call nodes or agents, and a set of edges connecting the vertices to each other. An edge that connects a vertex to itself is called a self-loop and vertices connected by edges are called neighbors.

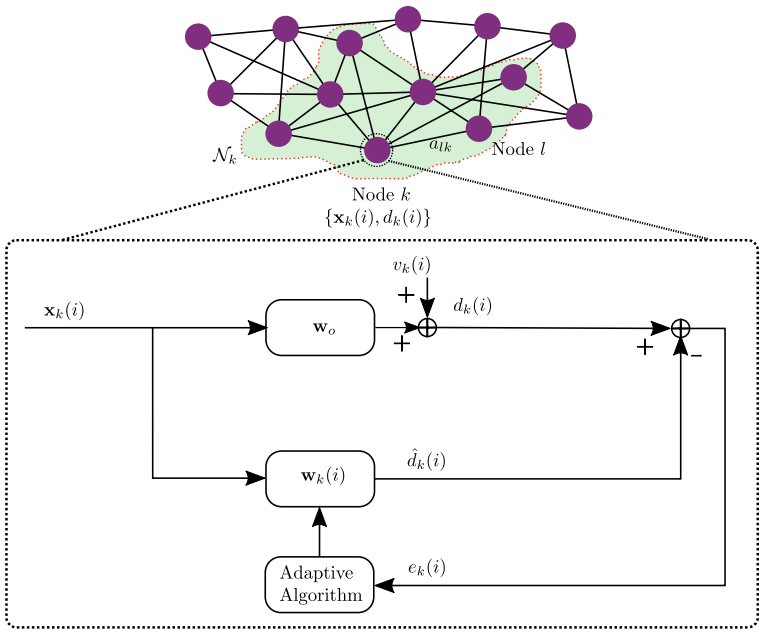


Figure 2.1: A distributed adaptive network

- The scalar a_{lk} will be used by agent k to scale data it receives from agent l. The weight a_{lk} is non-negative.
- The neighborhood of an agent k is denoted by $\mathcal{N}_k$ and it consists of all agents that are connected to agent k by an edge, in addition to agent k itself.
- In general, any two neighboring agents k and l have the ability to share information over the edge connecting them. Whether this exchange of information occurs, and whether it is uni-directional, bi-directional, or non-existent, will depend on the values of the weighting scalars a_{kl}, a_{lk} assigned to the edge.
- We assume the graph is undirected so that if agent k is a neighbor of agent l, then agent l is also a neighbor of agent k. Any two neighbors can share information both ways over the edge connecting them.

We associate with each agent a twice-differentiable individual cost function, denoted by $j_k(\mathbf{w})$ to estimate an $M \times 1$ unknown vector $\mathbf{w}_o \in \mathbb{C}^M$. This function $j_k(\mathbf{w})$ is sometimes called the utility function in applications involving resource management issues and the risk function in machine learning applications.

The goal of the network of agents is still to seek the unique minimizer of the aggregate cost function, $j(\mathbf{w})$ as follows:

$$\mathbf{w}_o \triangleq \min_{\mathbf{w}\in\mathbb{C}^M} j(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^{N} j_k(\mathbf{w}_k), \tag{2-1}$$

and the local objective functions are given by:

$$j_k(\mathbf{w}) = \frac{1}{n_k}\sum_{i=1}^{n_k} \ell(\mathbf{w}_k; \mathbf{x}_k(i), d_k(i)), \tag{2-2}$$

where $\ell(\mathbf{w}_k; \mathbf{x}_k(i), d_k(i))$ quantifies the loss of the model parameter $\mathbf{w}_k$ on data sample $\{\mathbf{x}_k(i), d_k(i)\}$. The presented model considers a desired signal $d_k(i)$, at each time i described by the model:

$$d_k(i) = \mathbf{w}_o^* \mathbf{x}_k(i) + v_k(i), \quad k = 1, 2, \ldots, N, \tag{2-3}$$

where $v_k(i)$ represents Gaussian noise with zero mean and variance $\sigma_{v_k}^2$ at each agent k that is uncorrelated with $\mathbf{x}_k(i)$.

To solve (2-1), distributed algorithms that are based on gradient-descent methods [35, 36, 37, 38] are effective and easy to implement. Two prominent variants of distributed algorithms are the consensus strategy [13, 39, 40] and the diffusion strategy [41, 42, 43]. We adopt the diffusion strategy because it has been shown to be a more effective scheme than other previously reported schemes [44]. The diffusion strategy is listed in Algorithm 1 where each agent k, first computes its local gradient and updates the intermediate estimated parameter $\mathbf{h}_k(i)$ as follows:

$$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) - \mu_k \nabla j_k\Big(\mathbf{w}_k(i-1)\Big), \tag{2-4}$$

and in the next step, executes weighted averaging for the received intermediate updates $\mathbf{h}_l(i)$ from its neighbors $l \in \mathcal{N}_k$ as follows:

$$\mathbf{w}_k(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\mathbf{h}_l(i). \tag{2-5}$$

The combination coefficients $\{a_{lk}\}_{l=1,k=1}^{N}$ are non-negative and satisfy

Algorithm 1: Diffusion strategy for distributed learning at node k

input : Initial values $\mathbf{w}_k(-1) \in \mathbb{R}^M$, combination matrix $\mathbf{A}$, and step-size μ_k

output : $\mathbf{w}_k(i)$

for $i = 0, 1, 2, \ldots$ **do**

$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) - \mu_k \nabla j_k\Big(\mathbf{w}_k(i-1)\Big)$

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

end for

$$a_{lk} \begin{cases} = 0, & l \notin \mathcal{N}_k \\ \geq 0, & l \in \mathcal{N}_k \end{cases}, \quad a_{lk} = a_{kl}, \quad \sum_{l \in \mathcal{N}_k} a_{lk} = 1, \tag{2-6}$$

which implies that the combination matrix $\mathbf{A} = [a_{lk}] \in \mathbb{R}^{N \times N}$ is a symmetric and doubly-stochastic matrix, i.e.,

$$\mathbf{A} = \mathbf{A}^T \quad \text{and} \quad \mathbf{A}\mathbb{1}_N = \mathbb{1}_N. \tag{2-7}$$

One example of doubly-stochastic combination coefficients is to use the Metropolis rule [41] as follows:

$$a_{lk} = \begin{cases} \frac{1}{\max\{n_k, n_l\}} & , l \in \mathcal{N}_k \setminus \{k\} \\ 1 - \sum_{m \in \mathcal{N}_k \setminus \{k\}} a_{mk} & , l = k \\ 0 & , l \notin \mathcal{N}_k \end{cases}. \tag{2-8}$$

We consider the mean-square error (MSE) as the local objective function (2-2) given by:

$$\begin{aligned} j_k(\mathbf{w}_k(i-1)) &= \mathbb{E}\Big[\big\|e_k(i)\big\|^2\Big] \triangleq \mathbb{E}\Big[\big\|d_k(i) - \hat{d}_k(i)\big\|^2\Big] \\ &= \mathbb{E}\Big[\big\|d_k(i) - \mathbf{w}_k^*(i-1)\mathbf{x}_k(i)\big\|^2\Big], \end{aligned} \tag{2-9}$$

where

$$\hat{d}_k(i) = \mathbf{w}_k^*(i-1)\mathbf{x}_k(i) \tag{2-10}$$

is the output of the adaptive filter and $e_k(i) = d_k(i) - \mathbf{w}_k^*(i-1)\mathbf{x}_k(i)$ is the estimation error (Figure 2.1). The gradient of (2-9) with respect to $\mathbf{w}_k^*(i-1)$ is $\nabla j_k(\mathbf{w}_k(i-1)) = -\mathbb{E}[\mathbf{x}_k(i)e_k^*(i)]$. Replacing an instantaneous approximation of it into (2-4), we arrive at the distributed least-mean square (DLMS) algorithm at agent k as follows:

$$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mu_k \mathbf{x}_k(i)e_k^*(i) \tag{2-11a}$$

$$\mathbf{w}_k(i) = \sum_{l \in \mathcal{N}_k} a_{lk}\mathbf{h}_l(i). \tag{2-11b}$$

In order to ensure convergence of $\mathbf{w}_k(i)$ in the mean, the step-size of the DLMS algorithm must be chosen such that [42]

$$0 < \mu_k < \frac{2}{\lambda_{\max}(\mathbf{R}_{x_k})}, \tag{2-12}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}_{x_k}$, and $\mathbf{R}_{x_k} \in \mathbb{C}^{M\times M}$ is the covariance matrix of $\mathbf{x}_k$. Note that for LMS-type algorithms [36, 37, 38], if the eigenvalue spread $(\lambda_{\max}/\lambda_{\min})$ of the covariance matrix of the input is large, it is appropriate to choose the step-size much smaller than its upper bound [45]. Therefore, the convergence rate of LMS-type algorithms varies substantially with the statistics of the input signal $\mathbf{x}_k(i)$. If $\mathbf{x}_k(i)$ is a white noise, the convergence rate is high. On the other hand, if the correlation between successive samples of $\mathbf{x}_k(i)$ is high, LMS-type algorithms converge slowly.

The recursive least-squares (RLS) algorithms are known to pursue fast convergence even when the eigenvalue spread of the covariance matrix of the input is large. The RLS algorithm can be derived in different ways. For the adaptive networks, let us consider a network of N nodes distributed over an area as in Figure 2.1. At time i, we globally collect the input regressors into a matrix $\mathbf{X}_i$ and the desired signal into vector $\mathbf{d}_i$ as follows:

$$\begin{aligned} \mathbf{X}_i &= \text{blockdiag}\{\mathbf{x}_1(i), \ldots, \mathbf{x}_N(i)\} \quad (MN \times N) \\ \mathbf{d}_i &= \text{col}\{d_1(i), \ldots, d_N(i)\} \qquad\qquad (N \times 1). \end{aligned} \tag{2-13}$$

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

$$\begin{aligned} \boldsymbol{X}_i &= \text{blockdiag}\{\mathbf{X}_i, \ldots, \mathbf{X}_0\} \quad (MN(i+1) \times N(i+1)), \\ \boldsymbol{d}_i &= \text{col}\{\mathbf{d}_i, \ldots, \mathbf{d}_0\}^T \qquad\quad (1 \times N(i+1)). \end{aligned} \tag{2-14}$$

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 [15]:

$$\min_{\mathbf{w}} \left[\left\| \mathbf{w} - \check{\mathbf{w}} \right\|_{\boldsymbol{\Pi}_i}^2 + \left\| \boldsymbol{d}_i - \mathbf{w}^* \boldsymbol{X}_i \right\|_{\boldsymbol{\Sigma}_i}^2 \right], \tag{2-15}$$

where $\check{\mathbf{w}}$ is a given column vector, $\boldsymbol{\Pi}_i > 0$ is an $M \times M$ positive-definite matrix that incorporates a regularization term $\|\mathbf{w} - \check{\mathbf{w}}\|_{\boldsymbol{\Pi}_i}^2$ 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. We give more details on these vectors and matrices later in Section 4.1. The solution to (2-15) is given by diffusion RLS (DRLS) [15] as follows:

$$\mathbf{P}_k(i) = \frac{1}{\lambda} \left(\mathbf{P}_k(i-1) - \frac{\mathbf{P}_k(i-1)\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{P}_k(i-1)}{\lambda + \mathbf{x}_k^*(i)\mathbf{P}_k(i-1)\mathbf{x}_k(i)} \right) \tag{2-16a}$$

$$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mathbf{P}_k(i)\mathbf{x}_k(i)e_k^*(i) \tag{2-16b}$$

$$\mathbf{w}_k(i) = \sum_{l \in \mathcal{N}_k} a_{lk}\mathbf{h}_l(i), \tag{2-16c}$$

where $e_k(i) = d_k(i) - \mathbf{w}_k^*(i-1)\mathbf{x}_k(i)$ is the estimation error, $0 \ll \lambda < 1$ is known as the forgetting factor, $\mathbf{P}_k(-1) = \boldsymbol{\Pi}^{-1}$, and $\boldsymbol{\Pi} = \delta^{-1}\mathbf{I}_M$. Although DRLS algorithm converges very quickly when the eigenvalue spread of the covariance matrix of the input is large, it has a considerable increase in the computational cost when compared to the DLMS algorithm. We compare our proposed DQA-LMS and DQA-RLS algorithms that are based on the DLMS and DRLS, respectively, in Section 4.7, and discuss the use case of each algorithm.

2.1.1 Adaptive IoT Networks

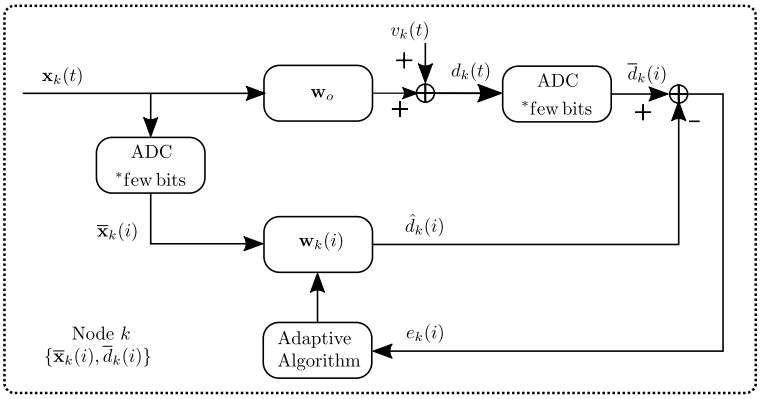


Figure 2.2: An IoT device k with quantized measurements

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 Figure 2.1. 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 high-precision sampled (or so-called full resolution) versions of $\mathbf{x}_k(t)$ and $d_k(t)$, and $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ denote the coarsely quantized versions of $\mathbf{x}_k(i)$ and $d_k(i)$, respectively. Note that we denote the continuous time (analog) data with (t) whereas we use (i) for discrete-time data.

As shown in Figure 2.2, 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 $\mathcal{T}_b = \{-\infty = \tau_0, \tau_1, \ldots, \tau_{2^b-1}, \tau_{2^b} = \infty\}$ 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 IoT network given in Figure 2.1 as follows:

- The proposed adaptive network consists of N nodes which run 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 discuss one special case where each node has one ADC to quantize both the input regressors and the desired signals.
- The fixed thresholds and labels of ADCs are calculated with the Lloyd-Max algorithm [46, 47] for a Gaussian signal with zero-mean and unit variance. Although there are practical methods for the inputs to the ADCs to approach the unit variance such as automatic gain control (AGC), we show in the numerical results that inputs with different variances do not degrade the performance of the proposed algorithms that are based on fixed thresholds and labels.

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.

2.1.2 An Example of Energy Saving by Coarse Quantization

In order to assess the power savings by low-resolution quantization, we consider a network with N IoT devices in which each device uses two ADCs. The power consumption of each ADC in millimeter wave systems is $P_{ADC}(b) = cB2^b$ [48], 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

$$P_{ADC,T}(b) = 2NcB2^b \qquad \text{(watts)}. \tag{2-17}$$

Figure 2.3 shows an example of the total power consumption of ADCs in a narrowband IoT (NB-IoT) network consisting of 20 devices with bandwidth $B = 200\,\text{kHz}$ [49] and considering the power consumption per conversion step of each ADC, $c = 494$ fJ, as in [50].

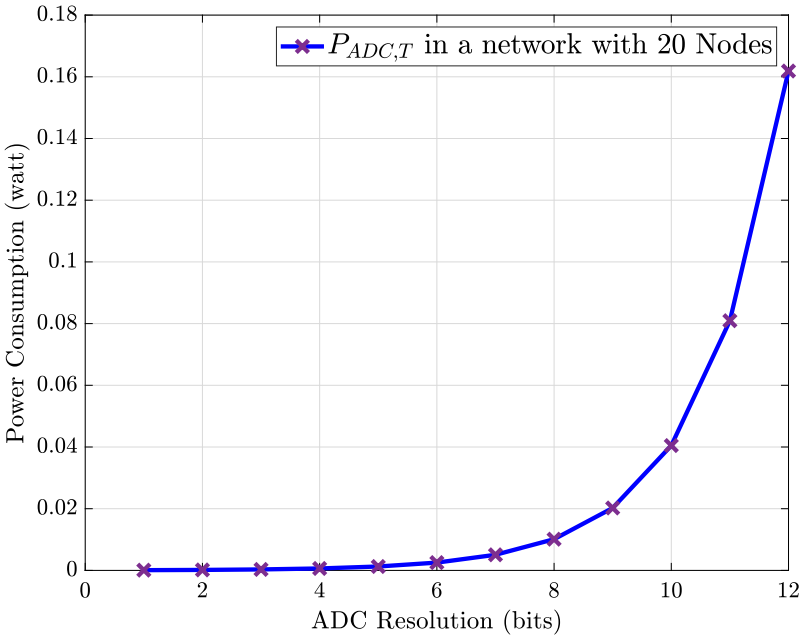


Figure 2.3: Power consumption of the ADCs in an adaptive IoT network.

2.2 Federated Learning for IoT Networks

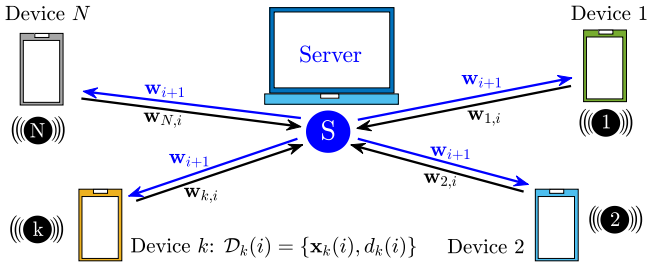


Figure 2.4: A federated IoT network

Figure 2.4 shows the architecture of an IoT network consisting of N IoT edge devices orchestrated by a server under federated learning strategy. Each device k, $k = 1, \ldots, N$, has access to local training data $\mathcal{D}_k$ including $n_k = |\mathcal{D}_k|$ data samples. The data sample i is represented by $\mathcal{D}_k(i) = \{\mathbf{x}_k(i), d_k(i)\}$ where $\mathbf{x}_k(i) \in \mathbb{C}^M$ and $d_k(i)$ are the ith input data vector and the associated output

response at device k, respectively. The goal of the learning task of the network is typically determined by the optimization problem:

$$\mathbf{w}_o \triangleq \min_{\mathbf{w}\in\mathbb{R}^M} j(\mathbf{w}) = \sum_{k=1}^{N} a_k j_k(\mathbf{w}), \tag{2-18}$$

where $j(\mathbf{w})$ is the global objective function, $j_k(\mathbf{w})$ is the local objective function, $a_k \geq 0$ denote weights and $\sum_k a_k = 1$. It is common to set $a_k = \frac{1}{N}$ in homogeneous networks to give an equal weight to every device k. The local objective function is given by:

$$j_k(\mathbf{w}) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell(\mathbf{w}; \mathbf{x}_k(i), d_k(i)), \tag{2-19}$$

where $\ell(\mathbf{w}; \mathbf{x}_k(i), d_k(i))$ quantifies the loss of the model parameter $\mathbf{w}$ on the data samples $\{\mathbf{x}_k(i), d_k(i)\}$. One solution of (2-18) can be obtained by applying iterative methods such as stochastic gradient descent techniques to (2-19) as follows:

$$\mathbf{w}_k(i) = \mathbf{w}(i-1) - \mu \nabla j_k(\mathbf{w}(i-1)), \tag{2-20}$$

where μ is the step size, and the server receives the updated parameter $\mathbf{w}_k(i)$ from devices and sends the following updated global parameter to N devices:

$$\mathbf{w}(i) = \frac{1}{N} \sum_{k=1}^{N} \mathbf{w}_k(i). \tag{2-21}$$

The server update (2-21) and local update (2-20) are key steps of FedAvg [10]. The data collected at IoT devices are described by the model:

$$d_k(i) = \mathbf{w}_o^* \mathbf{x}_k(i) + v_k(i),\ k = 1, \ldots, N, \text{and}\, i = 1, \ldots, n_k, \tag{2-22}$$

where $v_k(i)$ represents Gaussian noise with zero mean and variance $\sigma_{v_k}^2$ at each device k, that is uncorrelated with $\mathbf{x}_k(i)$. We consider the MSE as the local objective function (2-19) to estimate $\mathbf{w}(i)$ as defined by:

$$\begin{aligned} j_k(\mathbf{w}(i-1)) &= \mathbb{E}\left[\left\|e_k(i)\right\|^2\right] \triangleq \mathbb{E}\left[\left\|d_k(i) - \hat{d}_k(i)\right\|^2\right] \\ &= \mathbb{E}\left[\left\|d_k(i) - \mathbf{w}^*(i-1)\mathbf{x}_k(i)\right\|^2\right], \end{aligned} \tag{2-23}$$

where $\hat{d}_k(i) = \mathbf{w}^*(i-1)\mathbf{x}_k(i)$ is the output of the estimated parameters and $e_k(i) = d_k(i) - \mathbf{w}^*(i-1)\mathbf{x}_k(i)$ is the estimation error. The gradient of (2-23) with respect to $\mathbf{w}^*(i-1)$ is $\nabla j_k(\mathbf{w}(i-1)) = -\mathbf{x}_k(i)e_k^*(i)$. Replacing it into (2-20), we arrive at the Federated Averaging-LMS (FedAvg-LMS) algorithm as follows:

$$\mathbf{w}_k(i) = \mathbf{w}(i-1) + \mu\mathbf{x}_k(i)(d_k(i) - \mathbf{w}^*(i-1)\mathbf{x}_k(i))^* \tag{2-24a}$$

$$\mathbf{w}(i) = \frac{1}{N}\sum_{k=1}^{N}\mathbf{w}_k(i), \tag{2-24b}$$

which has been used in adaptive federated learning tasks [51, 52].

As shown in Fig. 2.4, each IoT device uses sensors and hence the measurement data are analog and should be converted to digital data for processing, as illustrated in Fig. 2.2. The average ADC resolution for different types of sensors in IoT devices varies from 5 to 16 bits (see Table. 1 in [34]). One concern is that the cost and power consumption of ADCs increase exponentially with the number of quantization bits [23] for each device. This motivates us to quantize the measurement data with few bits to support the low-cost and low-power consumption features of IoT sensors.

2.3 Signal Decomposition with Coarse Quantization

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

Teorema 2.1 *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 [53, 54]. 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*

$$\mathbb{E}_{x_Q(n),x(i)}\Big[x_Q(n)x^*(i)\Big] = g(n)\mathbb{E}_{x(n),x(i)}\Big[x(n)x^*(i)\Big] \tag{2-25}$$

in which

$$g(n) = \frac{1}{\sigma^2_{x(n)}}\mathbb{E}_{x(n)}\Big[f\Big(x(n)\Big)x^*(n)\Big]. \tag{2-26}$$

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.,

$$\mathbf{x}_Q = f\Big(\mathbf{x}\Big). \tag{2-27}$$

It follows from (2-25) that

$$\mathbf{R}_{x_Q x} = \mathbf{G}\mathbf{R}_x \tag{2-28}$$

where

$$\mathbf{G} = \text{diag}\Big\{g_1, g_2, \ldots, g_M\Big\} \tag{2-29}$$

represents a diagonal $M \times M$ matrix whose mth diagonal entry is computed as in (2-26). 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 [53]

$$\mathbf{x}_Q = \mathbf{G}\mathbf{x} + \mathbf{q} \tag{2-30}$$

The referred uncorrelation can be viewed as follows:

$$\mathbb{E}\Big[\mathbf{q}\mathbf{x}^*\Big] = \mathbb{E}\Big[\Big(\mathbf{x}_Q - \mathbf{G}\mathbf{x}\Big)\mathbf{x}^*\Big] = \mathbf{R}_{x_Q x} - \mathbf{G}\mathbf{R}_x = \mathbf{0}_{M\times M} \tag{2-31}$$

where we made use of (2-28).

2.3.1 Signal Decomposition for IoT Networks

To use this result in the distributed adaptive network in Fig. 2.1, let $\overline{\mathbf{x}}_k = Q_b(\mathbf{x}_k)$ 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 $p \in [0, 2^b-1]$ [24]. 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 $\overline{\mathbf{x}}_k$ and $\mathbf{x}_k$ in the form of (2-27), with the quantization function $Q_b(\cdot)$ playing the part of the scalar-valued nonlinear function $f(\cdot)$ in (2-27). We now use (2-30) to derive a model for the quantized vector $\overline{\mathbf{x}}_k$, which we later use to derive our quantization-aware algorithms. Employing Bussgang's theorem, $\overline{\mathbf{x}}_k$ can be decomposed as

$$\overline{\mathbf{x}}_k = \mathbf{G}_k\mathbf{x}_k + \mathbf{q}_k, \tag{2-32}$$

where the distortion $\mathbf{q}_k$ is uncorrelated with $\mathbf{x}_k$, and $\mathbf{G}_k \in \mathbb{R}^{M\times M}$ is a diagonal matrix given by [24]

$$\begin{aligned}\mathbf{G}_k = \mathrm{diag}(\mathbf{R}_{x_k})^{-\frac{1}{2}} \sum_{j=0}^{2^b-1} \frac{l_j}{\sqrt{\pi}} \Bigg[&\exp\Big(-\tau_j^2 \mathrm{diag}(\mathbf{R}_{x_k})^{-1}\Big) \\ &- \exp\Big(-\tau_{j+1}^2 \mathrm{diag}(\mathbf{R}_{x_k})^{-1}\Big)\Bigg],\end{aligned} \tag{2-33}$$

where the index b indicates the number of bits (ADC resolution). 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 $\mathbf{G}_k$ becomes $g_k\mathbf{I}_M$ and g_k is given by

$$g_k = \frac{1}{\sqrt{\sigma_{x_k}^2}} \sum_{j=0}^{2^b-1} \frac{l_j}{\sqrt{\pi}} \Big(e^{-\frac{\tau_j^2}{\sigma_{x_k}^2}} - e^{-\frac{\tau_{j+1}^2}{\sigma_{x_k}^2}}\Big). \tag{2-34}$$

Note that, as a simplifying approximation, we also apply this signal decomposition to the quantized desired signal, $\overline{d}_k$, which is the output of the second ADC in the system. To compute $\mathbf{G}_k$, 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 $\mathbf{R}_{\overline{x}_k}$ instead of $\mathbf{R}_{x_k}$. To overcome this, we can estimate the variance of the distortion $\mathbf{q}_k$, i.e., $\widehat{\sigma}^2_{q_k}$ (since the variance of the distortion is not accessible in practice, we estimate it) and the covariance matrix of the input signal, $\widehat{\mathbf{R}}_{x_k} = \mathbf{R}_{\overline{x}_k} + \widehat{\sigma}^2_{q_k}\mathbf{I}_M$. In the next section, we show how to design the ADCs and estimate the variance of the distortion.

2.3.2 ADC Design

In this section, we show how to compute the thresholds and labels to design the ADCs. To minimize the MSE between $\mathbf{x}_k$ and $\overline{\mathbf{x}}_k$, 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 [24] and compute the thresholds and labels as follows:

1. We generate an auxiliary Gaussian random variable, $\mathbf{x}_{aux}$, with unit variance and then use the Lloyd-Max algorithm [46, 47] to find a set of thresholds $\mathcal{T}_b = \{\tau_1, \ldots, \tau_{2^b-1}\}$ 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 $\tau_{2^b} = \infty$, i.e., $\mathcal{T}_b = \{-\infty = \tau_0, \tau_1, \ldots, \tau_{2^b-1}, \tau_{2^b} = \infty\}$.

3. We quantize $\mathbf{x}_{aux}$ using $\mathcal{T}_b$ and $\mathcal{L}_b$, generate the quantized signal $\overline{\mathbf{x}}_{aux}$, and estimate the variance of the quantization noise, $\sigma^2_{q_k}$ with the subtraction

of the variance of the quantized auxiliary signal from the variance of the auxiliary signal [24]

$$\widehat{\sigma}^2_{q_k} = \sigma^2_{x_{aux}} - \sigma^2_{\overline{x}_{aux}}. \tag{2-35}$$

4. We define the estimate of the covariance matrix of the quantization noise as follows:

$$\widehat{\mathbf{R}}_{q_k} = \widehat{\sigma}^2_{q_k} \mathbf{I}_M. \tag{2-36}$$

Note that steps 1 and 2 design the thresholds and labels of the ADCs, and steps 3 and 4 are useful to estimate $\mathbf{R}_{x_k}$.

2.4 Chapter Summary

This chapter has reviewed the technical background of adaptive IoT networks in order to introduce two methods of learning in IoT networks, i.e., distributed learning, where the agents operate in a peer-to-peer mode, and federated learning, where the agents work in a client-server mode. Furthermore, two key distributed adaptive algorithms for adaptive IoT networks were presented. The signal and system model used in distributed learning were described in detail, and the need for the saving energy by using low-resolution ADCs in these networks was identified. Subsequently, a signal decomposition method was introduced which is a fundamental tool in energy-efficient signal processing with low-resolution ADCs. Moreover, a design method of ADCs for IoT networks was detailed.

3
Distributed Quantization-Aware LMS Algorithm

In this chapter, we present the derivation of the proposed DQA-LMS algorithm and a statistical analysis of the proposed DQA-LMS algorithm. In addition, we devise a bias compensation strategy, investigate the computational complexity of the proposed and existing algorithms, and assess the estimation performance of the DQA-LMS algorithm for a parameter estimation setup with numerical results.

3.1
Derivation of DQA-LMS

Let us consider $\mathbf{x}_k(t)$ and $d_k(t)$ as the analog input and output of the unknown system $\mathbf{w}_o$ at node k (Fig. 2.1). Let $\mathbf{x}_k(i)$ and $d_k(i)$ denote the high-precision sampled versions of $\mathbf{x}_k(t)$ and $d_k(t)$, and $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ denote the coarsely quantized versions of $\mathbf{x}_k(i)$ and $d_k(i)$, respectively.

We show next that a learning algorithm based directly on (2-10) is biased for estimating $\mathbf{w}_o$, and show how to correct for this bias. To this end, let $\beta_k(i)$ be the bias compensation term to be chosen shortly, and define $\widehat{d}_k(i) = \beta_k(i)\mathbf{w}_k^*(i-1)\overline{\mathbf{x}}_k(i)$ to construct an MSE cost function as follows:

$$\begin{aligned} J_k(\mathbf{w}_k(i)) &= \mathbb{E}[|e_k(i)|^2] = \mathbb{E}[|\overline{d}_k(i) - \widehat{d}_k(i)|^2] \\ &= \mathbb{E}[|\overline{d}_k(i) - \beta_k(i)\mathbf{w}_k^*(i-1)\overline{\mathbf{x}}_k(i)|^2], \end{aligned} \tag{3-1}$$

which depends only on the received quantized quantities $\overline{d}_k(i)$, $\overline{\mathbf{x}}_k(i)$, and the bias compensation term $\beta_k(i)$. For $\beta_k(i) = 1$ (as in DLMS (2-9)), the quantization of $d_k(i)$ would result in biased estimates of $\mathbf{w}_o$. The gradient of (3-1) with respect to $\mathbf{w}_k^*(i-1)$ is given by:

$$\nabla j_k\Big(\mathbf{w}_k(i-1)\Big) = -\beta_k(i)\overline{\mathbf{x}}_k(i)\Big(\overline{d}_k(i) - \beta_k(i)\mathbf{w}_k^*(i-1)\overline{\mathbf{x}}_k(i)\Big)^*. \tag{3-2}$$

Replacing (3-2) in 2-4 and using (2-5), we obtain the DQA-LMS algorithm as follows:

$$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mu_k\beta_k(i)\overline{\mathbf{x}}_k(i)e_k^*(i) \tag{3-3a}$$

$$\mathbf{w}_k(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\mathbf{h}_l(i), \tag{3-3b}$$

where $e_k(i) = \overline{d}_k(i) - \beta_k(i)\mathbf{w}_k^*(i-1)\overline{\mathbf{x}}_k(i)$ is the estimation error. In the following we show how to optimally choose $\beta_k(i)$ to reduce the bias.

3.2 Mean Performance Analysis

To start the mean performance analysis, we use the following assumption that is very common in parameter estimation [55, 56, 57] and adaptive signal processing [58].

Assumption 1: The input data 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 $\sigma_{v_k}^2$ and the quantized regressors $\overline{\mathbf{x}}_k(i)$ with covariance matrices $\mathbf{R}_{\overline{x}_k} = \mathbb{E}[\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)]$. Moreover, covariance matrices are time-invariant, and $\mathbf{x}_k(i)$, $v_k(i)$ and $\overline{\mathbf{x}}_k(i)$ are assumed spatially independent.

To analyze the performance of DQA-LMS, we use the weight-error vectors defined as:

$$\widetilde{\mathbf{w}}_k(i) = \mathbf{w}_o - \mathbf{w}_k(i), \qquad \text{and} \qquad \widetilde{\mathbf{h}}_k(i) = \mathbf{w}_o - \mathbf{h}_k(i). \tag{3-4}$$

Under Assumption 1, if the entries in the input regressors $\mathbf{x}_k(i)$ are uncorrelated and with equal variance, we have $\mathbf{R}_{x_k} = \mathbb{E}[\mathbf{x}_k(i)\mathbf{x}_k^*(i)] \approx \sigma_{x_k}^2\mathbf{I}_M$ and the matrix $\mathbf{G}_{x_k}$ reduces to $g_{x_k}\mathbf{I}_M$. Let us denote $\mathbf{R}_{q_k} = E[\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)] \approx \widehat{\sigma}_{q_k}^2\mathbf{I}_M$ where $\widehat{\sigma}_{q_k}^2$ is given by (2-35).

Using (2-32), we can decompose $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ as follows:

$$\overline{\mathbf{x}}_k(i) = g_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i) \tag{3-5a}$$

$$\overline{d}_k(i) = g_{d_k}d_k(i) + q_{d_k}(i) = g_{d_k}\mathbf{w}_o^*\mathbf{x}_k(i) + p_k(i), \tag{3-5b}$$

where $p_k(i) = g_{d_k}v_k(i) + q_{d_k}(i)$. Using this decomposition, we write the error $e_k(i)$ as follows:

$$\begin{aligned} e_k(i) &= \overline{d}_k(i) - \beta_k(i)\mathbf{w}_k^*(i-1)\overline{\mathbf{x}}_k(i) \\ &= g_{d_k}\mathbf{w}_o^*\mathbf{x}_k(i) + p_k(i) - \beta_k(i)\mathbf{w}_k^*(i-1)\Big(g_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i)\Big). \end{aligned} \tag{3-6}$$

Replacing (3-6) into (3-3a) and subtracting from $\mathbf{w}_o$ yields

$$\begin{aligned} \widetilde{\mathbf{h}}_k(i) &= \widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k(i)\overline{\mathbf{x}}_k(i)e_k^*(i) = \widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k(i)(g_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i))e_k^*(i) \\ &= \widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k(i)\Big(g_{x_k}g_{d_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{w}_o + g_{x_k}\mathbf{x}_k(i)p_k(i) \\ &\quad - g_{x_k}^2\beta_k(i)\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{w}_k(i-1) - g_{x_k}\beta_k(i)\mathbf{x}_k(i)\mathbf{q}_{x_k}^*(i)\mathbf{w}_k(i-1) \\ &\quad + g_{d_k}\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\mathbf{w}_o + \mathbf{q}_{x_k}(i)p_k(i) \\ &\quad - g_{x_k}\beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\mathbf{w}_k(i-1) - \beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\mathbf{w}_k(i-1)\Big) \\ &= \widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k(i)\Big(\Big(g_{x_k}g_{d_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i) + g_{d_k}\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\Big)\mathbf{w}_o \\ &\quad - \Big(g_{x_k}^2\beta_k(i)\mathbf{x}_k(i)\mathbf{x}_k^*(i) + g_{x_k}\beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i) + g_{x_k}\beta_k(i)\mathbf{x}_k(i)\mathbf{q}_{x_k}^*(i) \\ &\quad + \beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big)\mathbf{w}_k(i-1) + g_{x_k}\mathbf{x}_k(i)p_k(i) + \mathbf{q}_{x_k}(i)p_k(i)\Big). \end{aligned} \tag{3-7}$$

We take the expectation from both sides of (3-7). Since $\mathbf{x}_k(i)$, $\mathbf{q}_{x_k}(i)$, and $p_k(i)$ are uncorrelated pairwise, the expectations of these cross terms vanish. Considering this, we obtain

$$\begin{aligned}\mathbb{E}[\widetilde{\mathbf{h}}_k(i)] &= \mathbb{E}[\widetilde{\mathbf{w}}_k(i-1)] - \mu_k\Big(\mathbb{E}\Big[g_{x_k}g_{d_k}\beta_k(i)\mathbf{x}_k(i)\mathbf{x}_k^*(i)\Big]\mathbf{w}_o \\ &\quad - \mathbb{E}\Big[\Big(g_{x_k}^2\beta_k^2(i)\mathbf{x}_k(i)\mathbf{x}_k^*(i) + \beta_k^2(i)\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big)\mathbf{w}_k(i-1)\Big]\Big) \\ &= \mathbb{E}[\widetilde{\mathbf{w}}_k(i-1)] - \mu_k\Big(g_{x_k}g_{d_k}\mathbf{R}_{x_k}\mathbf{w}_o \\ &\quad - \Big(g_{x_k}^2\beta_k(i)\mathbf{R}_{x_k} + \beta_k(i)\mathbf{R}_{q_k}\Big)\mathbb{E}\Big[\mathbf{w}_k(i-1)\Big]\Big).\end{aligned} \tag{3-8}$$

In the last line of (3-8), we use a common assumption that states that $\mathbf{x}_k(i)$ varies slowly in relation to $\widetilde{\mathbf{w}}_k(i-1)$ [58]. Thus, when they appear inside the expectations we decouple their expected values. This also applies to $\mathbf{q}_{x_k}(i)$ in relation to $\widetilde{\mathbf{w}}_k(i-1)$.

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

$$g_{x_k}g_{d_k}\beta_k(i)\mathbf{R}_{x_k} = g_{x_k}^2\beta_k^2(i)\mathbf{R}_{x_k} + \beta_k^2(i)\mathbf{R}_{q_k}, \tag{3-9}$$

and we show in the next section that this condition is possible by appropriately choosing $\beta_k(i)$. Assuming (3-9) and using (3-4), we can write (3-8) as follows:

$$\mathbb{E}[\widetilde{\mathbf{h}}_k(i)] = \Big(\mathbf{I}_M - \mu_k g_{x_k}g_{d_k}\beta_k(i)\mathbf{R}_{x_k}\Big)\mathbb{E}[\widetilde{\mathbf{w}}_k(i-1)]. \tag{3-10}$$

Subtracting $\mathbf{w}_o$ from both sides of (3-3b) and using (3-10), we obtain

$$\mathbb{E}[\widetilde{\mathbf{w}}_k(i)] = \sum_{l\in\mathcal{N}_k} a_{lk}\widetilde{\mathbf{h}}_l(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\Big(\Big(\mathbf{I}_M - \mu_k g_{x_k}g_{d_k}\beta_k(i)\mathbf{R}_{x_k}\Big)\mathbb{E}[\widetilde{\mathbf{w}}_k(i-1)]\Big). \tag{3-11}$$

We define the following matrices:

$$\begin{aligned}
&\boldsymbol{\mathcal{A}} = \mathbf{A}\otimes\mathbf{I}_M && (MN\times MN)\\
&\boldsymbol{\mathcal{M}} = \text{blockdiag}\{\mu_1\mathbf{I}_M,\ldots,\mu_N\mathbf{I}_M\} && (MN\times MN)\\
&\boldsymbol{\mathcal{R}}_x = \text{blockdiag}\{g_{x_1}g_{d_1}\beta_1(i)\mathbf{R}_{x_1},\ldots,g_{x_N}g_{d_N}\beta_N(i)\mathbf{R}_{x_N}\} && (MN\times MN)\\
&\widetilde{\mathbf{W}}_i = \text{col}\Big\{\widetilde{\mathbf{w}}_1(i),\ldots,\widetilde{\mathbf{w}}_N(i)\Big\} && (MN\times 1),
\end{aligned} \tag{3-12}$$

and write the global form of (3-11) as follows:

$$\mathbb{E}[\widetilde{\mathbf{W}}_i] = \boldsymbol{\mathcal{A}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_x\Big)\mathbb{E}[\widetilde{\mathbf{W}}_{i-1}]. \tag{3-13}$$

The necessary and sufficient condition to ensure the mean stability of the network, i.e., $\mathbb{E}[\widetilde{\mathbf{W}}_i] \to 0$ as $i \to \infty$ is to have $\rho\Big(\boldsymbol{\mathcal{A}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_x\Big)\Big) < 1$. Since the spectral radius of $\boldsymbol{\mathcal{A}}$ is equal to one [41], we must have $\rho\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_x\Big) < 1$. Therefore, the stability condition for DQA-LMS is given by:

$$0 < \mu_k < \frac{2}{\lambda_{\max}(\boldsymbol{\mathcal{R}}_x)}, \tag{3-14}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{\mathcal{R}}_x$.

3.3 Bias Compensation

From (3-9), we must have

$$\beta_k(i)\mathbf{I}_M = g_{x_k} g_{d_k} \mathbf{R}_{x_k}\Big(g_{x_k}^2 \mathbf{R}_{x_k} + \mathbf{R}_{q_k}\Big)^{-1}. \tag{3-15}$$

Therefore, the bias compensation term is expressed by

$$\beta_k(i) = \beta_k = \frac{g_{x_k} g_{d_k} \sigma_{x_k}^2}{g_{x_k}^2 \sigma_{x_k}^2 + \widehat{\sigma}_{q_k}^2}. \tag{3-16}$$

Remark 1: *(One ADC for each sensor).* To reduce the cost and energy consumption of sensors, we consider one ADC to quantize the measurement data $\{\mathbf{x}_k(i), d_k(i)\}$. Then g_{x_k} and g_{d_k} can be considered equal and this reduces the complexity of our algorithm as well.

Remark 2: *(Approximation of data variance).* Since the devices receive quantized data and have access to the covariance of the quantized data, $\mathbf{R}_{\overline{x}_k} = \mathbb{E}[\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)] \approx \sigma_{\overline{x}_k}^2 \mathbf{I}_M$, we approximate the variance of high precision data as follows:

$$\widehat{\sigma}_{x_k}^2 = \widehat{\sigma}_{\overline{x}_k}^2 + \widehat{\sigma}_{q_k}^2, \tag{3-17}$$

and the variance of the quantized input is given by:

Algorithm 2: DQA-LMS algorithm at agent k

input : Initial values $\mathbf{w}_k(-1) \in \mathbb{R}^M$, combination matrix $\mathbf{A}$, step-size μ_k, $\widehat{\sigma}^2_{\overline{x}_k}(-1) = 0$ and $0 \ll \gamma < 1$
Generate $\mathcal{T}_b$ and $\mathcal{L}_b$, and Compute $\widehat{\sigma}^2_{q_k}$ from (2-35)

output : $\mathbf{w}_k(i)$

for $i = 0, 1, 2, \ldots$ **do**

data: $\overline{d}_k(i)$ and $\overline{\mathbf{x}}_k(i) = [\overline{x}_k(i), \overline{x}_k(i-1), \ldots, \overline{x}_k(i-M+1)]^T$

$\widehat{\sigma}^2_{\overline{x}_k}(i) = \gamma \widehat{\sigma}^2_{\overline{x}_k}(i-1) + (1-\gamma)|\overline{x}_k(i)|^2$

$\widehat{\sigma}^2_{x_k} = \widehat{\sigma}^2_{\overline{x}_k}(i) + \widehat{\sigma}^2_{q_k}$

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

$\beta_k(i) = \frac{g^2_{x_k} \widehat{\sigma}^2_{x_k}}{g^2_{x_k} \widehat{\sigma}^2_{x_k} + \widehat{\sigma}^2_{q_k}}$

$e_k(i) = \overline{d}_k(i) - \beta_k(i) \mathbf{w}_k^*(i-1) \overline{\mathbf{x}}_k(i)$

$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mu_k \beta_k(i) \mathbf{x}_k(i) e_k^*(i)$

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

end for

$$\widehat{\sigma}^2_{\overline{x}_k} = \frac{1}{M} \sum_{l=1}^{M} (\overline{x}_k(l) - \text{mean}\{\overline{x}_k\})^2, \tag{3-18}$$

where $\overline{x}_k(l)$ are the lth entry of vector $\overline{\mathbf{x}}_k(i)$. To reduce the complexity of computation of $\widehat{\sigma}^2_{\overline{x}_k}$, we use the iterative recursion at each time instant i as follows:

$$\widehat{\sigma}^2_{\overline{x}_k}(i) = \gamma \widehat{\sigma}^2_{\overline{x}_k}(i-1) + (1-\gamma)|\overline{x}_k(i)|^2, \tag{3-19}$$

where $\widehat{\sigma}^2_{\overline{x}_k}(-1) = 0$ and $0 \ll \gamma < 1$. Therefore, at each data sample i, the bias correction term is given by:

$$\beta_k = \frac{g^2_{x_k} \widehat{\sigma}^2_{x_k}}{g^2_{x_k} \widehat{\sigma}^2_{x_k} + \widehat{\sigma}^2_{q_k}}. \tag{3-20}$$

The DQA-LMS algorithm is summarized in Algorithm 2.

3.4 Adaptive Bias Compensation

In this section, we propose an adaptive strategy to compute the bias compensation term $\beta_k(i)$ for "delay line" inputs, which uses an approximation for $\mathbf{R}_{x_k}$ by the instantaneous values of the input vector $\mathbf{x}_k$. A simple approach to estimate $\mathbf{R}_{x_k}$ is obtained by employing the instantaneous values of $\mathbf{x}_k\mathbf{x}_k^*$ as follows

$$\widehat{\mathbf{R}}_{x_k}(i) = \mathbf{x}_k(i)\mathbf{x}_k^*(i) \qquad M \times M \tag{3-21}$$

and consequently for estimating $\mathbf{R}_{\overline{x}_k}(i)$

$$\widehat{\mathbf{R}}_{\overline{x}_k}(i) = \overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i) \qquad M \times M. \tag{3-22}$$

Since we do not have access to the instantaneous values of $\mathbf{x}_k(i)$ and process $\overline{\mathbf{x}}_k(i)$, we use the following estimate

$$\widehat{\mathbf{R}}_{x_k}(i) = \widehat{\mathbf{R}}_{\overline{x}_k}(i) + \widehat{\sigma}_{q_k}^2\mathbf{I}_M. \tag{3-23}$$

Note that we estimate $\mathbf{R}_{x_k}$ online to form the diagonal entries of the matrix $\mathbf{G}_k(i)$ considering (2-33) and since $\mathbf{G}_k(i)$ is a diagonal matrix whose entries are built based on the diagonal entries of the covariance matrix $\mathbf{R}_{x_k}$, we only consider the diagonal entries of $\widehat{\mathbf{R}}_{x_k}(i)$ which can be shown as follows:

$$\widehat{\mathbf{R}}_{x_k}(i) = \begin{bmatrix} \widehat{r}_{x_k}(i) & & \\ & \ddots & \\ & & \widehat{r}_{x_k}(i-M+1) \end{bmatrix},$$

where $\widehat{r}_{x_k}(i) = \overline{x}_k(i)\overline{x}_k^*(i) + \widehat{\sigma}_{q_k}^2$. Algorithm 3 summarizes the AdDQA-LMS algorithm.

3.5 Mean Square Performance Analysis

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

Algorithm 3: AdDQA-LMS algorithm at agent k

input : Initial values $\mathbf{w}_k(-1) \in \mathbb{R}^M$, combination matrix $\mathbf{A}$, step-size μ_k, $\boldsymbol{\beta}_k(-1) = \text{diag}\{b_k(-1), \ldots, b_k(-M)\} = \mathbf{0}_{M\times M}$
Generate $\mathcal{T}_b$ and $\mathcal{L}_b$, and Compute $\widehat{\sigma}^2_{q_k}$ from (2-35)

output : $\mathbf{w}_k(i)$

for $i = 0, 1, 2, \ldots$ **do**

data: $\overline{d}_k(i)$ and $\overline{\mathbf{x}}_k(i) = [\overline{x}_k(i), \overline{x}_k(i-1), \ldots, \overline{x}_k(i-M+1)]^T$

$\widehat{r}_{x_k}(i) = \overline{x}_k(i)\overline{x}^*_k(i) + \widehat{\sigma}^2_{q_k}$

$g_k(i) = \frac{1}{\sqrt{\widehat{r}_{x_k}(i)}} \sum_{j=0}^{2^b-1} \frac{l_j}{\sqrt{\pi}} \Big(e^{-\frac{\tau_j^2}{\widehat{r}_{x_k}(i)}} - e^{-\frac{\tau_{j+1}^2}{\widehat{r}_{x_k}(i)}} \Big)$

$b_k(i) = \frac{g_k(i)\widehat{r}_{x_k}(i)}{g_k(i)\widehat{r}_{x_k}(i) + \widehat{\sigma}^2_{q_k}}$

update $\boldsymbol{\beta}_k(i)$

$e_k(i) = \overline{d}_k(i) - \mathbf{w}^*_k(i-1)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)$

$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mu_k \boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)e^*_k(i)$

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

end for

(3-7) as

$$
\begin{aligned}
\widetilde{\mathbf{h}}_k(i) &= \widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k\overline{\mathbf{x}}_k(i)\Big(\overline{d}_k(i) - \beta_k\mathbf{w}^*_k(i-1)\overline{\mathbf{x}}_k(i)\Big)^* \\
&= \widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k\overline{\mathbf{x}}_k(i)\overline{d}^*_k(i) + \mu_k\beta_k^2\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}^*_k(i)\mathbf{w}_k(i-1) \\
&= \Big(\mathbf{I}_M - \mu_k\beta_k^2\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}^*_k(i)\Big)\widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k\overline{\mathbf{x}}_k(i)\Big(g_{d_k}\mathbf{x}^*_k(i)\mathbf{w}_o + p^*_k(i)\Big) \\
&\quad + \mu_k\beta_k^2\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}^*_k(i)\mathbf{w}_o \\
&= \Big(\mathbf{I}_M - \mu_k\beta_k^2\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}^*_k(i)\Big)\widetilde{\mathbf{w}}_k(i-1) - \underbrace{\mu_k\beta_k g_{x_k}g_{d_k}\mathbf{x}_k(i)\mathbf{x}^*_k(i)\mathbf{w}_o}_{I} \\
&\quad - \mu_k\beta_k g_{d_k}\mathbf{q}_{x_k}(i)\mathbf{x}^*_k(i)\mathbf{w}_o - \mu_k\beta_k g_{x_k}\mathbf{x}_k(i)p^*_k(i) - \mu_k\beta_k\mathbf{q}_{x_k}(i)p^*_k(i) \\
&\quad + \underbrace{\mu_k\beta_k^2 g^2_{x_k}\mathbf{x}_k(i)\mathbf{x}^*_k(i)\mathbf{w}_o + \mu_k\beta_k^2\mathbf{q}_{x_k}(i)\mathbf{q}^*_{x_k}(i)\mathbf{w}_o}_{II}.
\end{aligned}
\tag{3-24}
$$

The choice of β_k in (3-20) makes (3-9) approximately true. In order to reduce the complexity of the model, we assume in the sequel that (3-9) is exactly true,

so the instantaneous values of terms I and II in (3-24) are equal with different signs and vanish for sufficiently large i (our simulations in Sec. 3.7 show that this approximation is reasonable). Then (3-24) simplifies to

$$\begin{aligned}\widetilde{\mathbf{h}}_k(i) = \Big(\mathbf{I}_M - \mu_k\beta_k^2\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\Big)\widetilde{\mathbf{w}}_k(i-1) - \mu_k\beta_k g_{d_k}\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\mathbf{w}_o \\ - \mu_k\beta_k g_{x_k}\mathbf{x}_k(i)p_k^*(i) - \mu_k\beta_k\mathbf{q}_{x_k}(i)p_k^*(i).\end{aligned} \tag{3-25}$$

Therefore, the weight-error vectors of the combined estimates are given by

$$\begin{aligned}\widetilde{\mathbf{w}}_k(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\widetilde{\mathbf{h}}_l(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\Big(\mathbf{I}_M - \mu_l\beta_l^2\overline{\mathbf{x}}_l(i)\overline{\mathbf{x}}_l^*(i)\Big)\widetilde{\mathbf{w}}_l(i-1) \\ - \sum_{l\in\mathcal{N}_k} a_{lk}\mu_l\beta_l g_{d_l}\mathbf{q}_{x_l}(i)\mathbf{x}_l^*(i)\mathbf{w}_o - \sum_{l\in\mathcal{N}_k} a_{lk}\mu_l\beta_l g_{x_l}\mathbf{x}_l(i)p_l^*(i) \\ - \sum_{l\in\mathcal{N}_k} a_{lk}\mu_l\beta_l\mathbf{q}_{x_l}(i)p_l^*(i)\end{aligned} \tag{3-26}$$

Let us now define the following quantities

$$\begin{aligned}\boldsymbol{\mathcal{R}}_{\overline{x}} &= \text{blockdiag}\left\{\mathbf{R}_{\overline{x}_1},\ldots,\mathbf{R}_{\overline{x}_N}\right\} && (MN\times MN)\\ \boldsymbol{\mathcal{B}} &= \text{blockdiag}\left\{\beta_1\mathbf{I}_M,\ldots,\beta_N\mathbf{I}_M\right\} && (MN\times MN)\\ \mathfrak{B} &= \text{blockdiag}\left\{\beta_1^2\mathbf{I}_M,\ldots,\beta_N^2\mathbf{I}_M\right\} && (MN\times MN)\\ \boldsymbol{\mathcal{G}} &= \text{blockdiag}\left\{g_{x_1}\mathbf{I}_M,\ldots,g_{x_N}\mathbf{I}_M\right\} && (MN\times MN)\\ \boldsymbol{\Upsilon}_i &= \text{blockdiag}\left\{\mathbf{q}_{x_1}(i)\mathbf{x}_1^*(i),\ldots,\mathbf{q}_{x_N}(i)\mathbf{x}_N^*(i)\right\} && (MN\times MN)\\ \boldsymbol{\xi}_i &= \text{col}\left\{\mathbf{x}_1(i)p_1^*(i),\ldots,\mathbf{x}_N(i)p_N^*(i)\right\} && (MN\times 1)\\ \boldsymbol{\zeta}_i &= \text{col}\left\{\mathbf{q}_{x_1}(i)p_1^*(i),\ldots,\mathbf{q}_{x_N}(i)p_N^*(i)\right\} && (MN\times 1)\\ \boldsymbol{\eta} &= \text{col}\left\{g_{d_1}\mathbf{w}_o,\ldots,g_{d_N}\mathbf{w}_o\right\} && (MN\times 1),\end{aligned}$$

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

$$\widetilde{\mathbf{W}}_i = \boldsymbol{\mathcal{A}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\mathfrak{B}\boldsymbol{\mathcal{R}}_{\overline{x}}\Big)\widetilde{\mathbf{W}}_{i-1} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i.$$

Defining an $MN\times MN$ matrix $\boldsymbol{\mathcal{D}}$ as follows:

$$\begin{aligned}\boldsymbol{\mathcal{D}} &= \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\Big(\mathbf{I}_{MN} - \text{blockdiag}\{\mu_1\beta_1^2\overline{\mathbf{x}}_1(i)\overline{\mathbf{x}}_1^*(i), \ldots, \mu_N\beta_N^2\overline{\mathbf{x}}_N(i)\overline{\mathbf{x}}_N^*(i)\}\Big)\Big] \\ &= \boldsymbol{\mathcal{A}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\mathfrak{B}\boldsymbol{\mathcal{R}}_{\overline{x}}\Big)\end{aligned} \tag{3-27}$$

and 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] &= \mathbb{E}\Big[\boldsymbol{\mathcal{D}}\widetilde{\mathbf{W}}_{i-1}\widetilde{\mathbf{W}}_{i-1}^*\boldsymbol{\mathcal{D}}^*\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{A}}^T\Big].\end{aligned} \tag{3-28}$$

We now use the commutative property of the expectation and vectorization operations, and the relationship between the vectorization operation and the Kronecker product, $\text{vec}(ABC) = (C^T \otimes A)\,\text{vec}(B)$, to write (3-28) as follows:

$$\text{vec}(\boldsymbol{\Omega}_i) = (\boldsymbol{\mathcal{D}} \otimes \boldsymbol{\mathcal{D}})\,\text{vec}(\boldsymbol{\Omega}_{i-1}) + \text{vec}(\boldsymbol{\Xi}_i), \tag{3-29}$$

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 (3-28). Note that $\text{vec}(\boldsymbol{\Omega}_i)$ is stable if and only if the spectral radius of $(\boldsymbol{\mathcal{D}} \otimes \boldsymbol{\mathcal{D}})$ is strictly smaller than 1. Therefore, we obtain

$$\lim_{i \to +\infty} \text{vec}(\boldsymbol{\Omega}_i) = \Big(\mathbf{I}_{M^2N^2} - (\boldsymbol{\mathcal{D}} \otimes \boldsymbol{\mathcal{D}})\Big)^{-1}\,\text{vec}(\boldsymbol{\Xi}_{+\infty}), \tag{3-30}$$

and $\text{vec}(\boldsymbol{\Xi}_{+\infty})$ is given by

$$\begin{aligned}\mathrm{vec}(\boldsymbol{\Xi}_{+\infty}) &= \lim_{i\to+\infty}\mathrm{vec}(\boldsymbol{\Xi}_i) = (\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\\ &\quad \mathrm{vec}\Big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\\ &\quad +\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\Big]\\ &\quad +\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\zeta}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]\Big)\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\Big)\\ &\approx (\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\\ &\quad \mathrm{vec}\Big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\Big)\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\Big),\end{aligned} \tag{3-31}$$

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

$$\begin{aligned}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big] &= \mathrm{diag}\Big\{\mathbf{R}_{x_1}\sigma_{p_1}^2,\ldots,\mathbf{R}_{x_N}\sigma_{p_N}^2\Big\}\\ \mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big] &= \mathrm{diag}\Big\{g_{d_1}^2\mathbf{R}_{q_x,1}\Big(\mathbf{w}_o^*\mathbf{R}_{x_1}\mathbf{w}_o\Big),\ldots,g_{d_N}^2\mathbf{R}_{q_x,N}\Big(\mathbf{w}_o^*\mathbf{R}_{x_N}\mathbf{w}_o\Big)\Big\}\\ \mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big] &= \mathrm{diag}\Big\{\mathbf{R}_{q_x,1}\sigma_{p_1}^2,\ldots,\mathbf{R}_{q_x,N}\sigma_{p_N}^2\Big\}.\end{aligned} \tag{3-32}$$

The analytical computation of $\mathbf{R}_{\overline{x}_k}$, $\mathbf{R}_{q_x,k}$, and $\sigma_{p_k}^2$ is detailed in Appendix A. Therefore, the steady-state MSD at node k is given by

$$\begin{aligned}\mathrm{MSD}_k &= \lim_{i\to+\infty}\mathbb{E}[\|\widetilde{\mathbf{w}}_k(i)\|^2] = \mathrm{vec}^T\Big(\mathbf{C}_k\otimes\mathbf{I}_M\Big)\,\mathrm{vec}(\boldsymbol{\Omega}_{+\infty})\\ &= \mathrm{vec}^T\Big(\mathbf{C}_k\otimes\mathbf{I}_M\Big)\Big(\mathbf{I}_{M^2N^2}-(\boldsymbol{\mathcal{D}}\otimes\boldsymbol{\mathcal{D}})\Big)^{-1}(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\\ &\quad \mathrm{vec}\Big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\Big)\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\Big),\end{aligned} \tag{3-33}$$

where $\mathbf{C}_k$ is an $N\times N$ matrix with zero entries and a single one 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 (3-29) that, for $0\ll\mu_k<1$ and $\boldsymbol{\mathcal{A}}$ with the entries a_{lk} subject to (2-6), the eigenvalues of $\boldsymbol{\mathcal{D}}\otimes\boldsymbol{\mathcal{D}}$ remain in the interval $(-1,1)$, and thus DQA-LMS is stable in the mean-square sense and $\Big(\mathbf{I}_{M^2N^2}-(\boldsymbol{\mathcal{D}}\otimes\boldsymbol{\mathcal{D}})\Big)$ in (3-33) is nonsingular. The global MSD over all the nodes is given by

$$\mathrm{MSD}_{\mathrm{global}} = \frac{1}{N}\sum_{k=1}^{N}\mathrm{MSD}_k = \frac{1}{N}\,\mathrm{Tr}(\boldsymbol{\Omega}_{+\infty}). \tag{3-34}$$

Remark 3: *(High precision signals, $b = \infty$).* Increasing the number of quantization bits, the diagonal entries of $\mathbf{G}_{x_k}$ approach unity where for high precision signals ($b = \infty$) with $\overline{\mathbf{x}}_k = \mathbf{x}_k$, we have $\mathbf{G}_{x_k} = \mathbf{I}_M$ according to (2-33) and $\mathbf{R}_{q_x,k} = \mathbf{0}$ according to (3-5a), and consequently $\boldsymbol{\beta}_k(i) = \mathbf{I}_M$ from (3-20). For $b = \infty$ we also have $q_{d_k} = 0$ and $g_{d_k} = 1$, and since $p_k(i) = g_{d_k}v_k(i) + q_{d_k}(i)$, thus $\sigma^2_{p_k} = \sigma^2_{v_k}$. Therefore, for high precision signals, the third and fourth definitions in (3-32) vanish, and with $\boldsymbol{\mathcal{R}} = \mathrm{blockdiag}\{\mathbf{R}_{x_1},\ldots,\mathbf{R}_{x_N}\}$, (3-33) reduces to

$$\begin{aligned}\mathrm{MSD}_k = \mathrm{vec}^T\Big(\mathbf{C}_k \otimes \mathbf{I}_M\Big) \\ \Big(\mathbf{I}_{M^2N^2} - \Big((\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}) \otimes (\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}})(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\Big)\Big)^{-1} \\ (\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\,\mathrm{vec}\Big(\,\mathrm{diag}\Big\{\mu_1^2\sigma_{v_1}^2\mathbf{R}_{x_1},\ldots,\mu_N^2\sigma_{v_N}^2\mathbf{R}_{x_N}\Big\}\Big),\end{aligned} \tag{3-35}$$

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

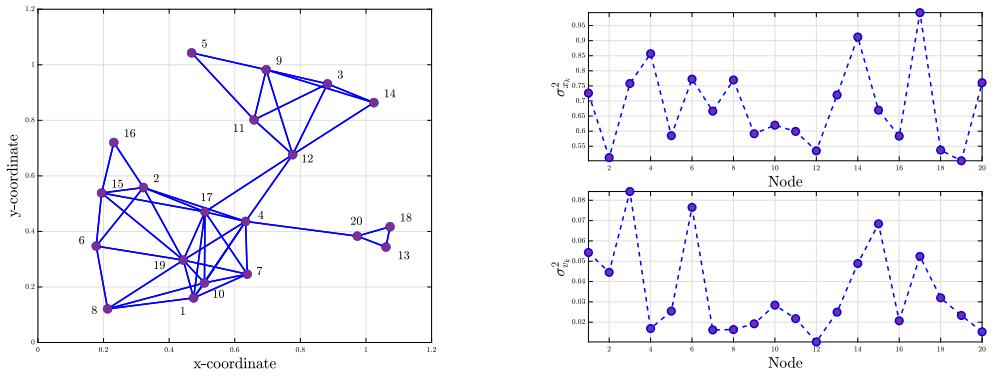
3.6 Computational Complexity

Table 3.1 shows the computational complexity of the DQA-LMS 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-LMS performs a few more operations ($\approx O(2^b)$) than DLMS. Note that we compute $g_k(i)$ online since this is more appropriate for non-stationary input data. However, one can compute $\mathbf{G}_k$ offline if an estimate of $\mathbf{R}_{x_k}$ in (2-33) is available. However, the extra complexity of DQA-LMS allows the system to work in a more energy-efficient way.

Table 3.1: Computational complexity of DQA-LMS algorithm per time instant at agent k

Task	$+-$	$\times$	$\div$	exp
$\widehat{\sigma}^2_{x_k}$	3	3	0	0
$g_k(i)$	2^b-1	$2^{b+1}+1$	2^b+1	2^b
$\beta_k(i)$	1	2	1	0
$\widehat{d}_k(i)$	$M-1$	$M+1$	0	0
$e_k(i)$	1	0	0	0
$\mathbf{h}_k(i+1)$	M	$M+2$	0	0
$\mathbf{w}_k(i+1)$	n_kM	n_kM	0	0
Total	$(2+n_k)M+2^b+3$	$(2+n_k)M+2^{b+1}+9$	2^b+2	2^b
Total (DLMS)	$(2+n_k)M$	$(2+n_k)M+1$	0	0

3.7 Numerical Results



(a) Distributed network (b) Input and noise variances

Figure 3.1: A wireless network with $N = 20$ nodes.

In this section, we assess the performance of the DLMS [14], DQA-LMS [59] and AdDQA-LMS [60] algorithms for a parameter estimation problem in an IoT network with $N = 20$ nodes. The impulse response of the unknown system has $M = 8$ taps, is generated randomly and normalized to one. The input signals $\mathbf{x}_k(i)$ at each node are generated by a white Gaussian noise process with variance $\sigma^2_{x_k}$ and quantized using the Lloyd-Max quantization scheme [46, 47] to generate $\overline{\mathbf{x}}_k(i)$. The noise samples of each node are drawn

from a zero mean white Gaussian process with variance $\sigma^2_{v_k}$. Fig. 3.1 plots the network details.

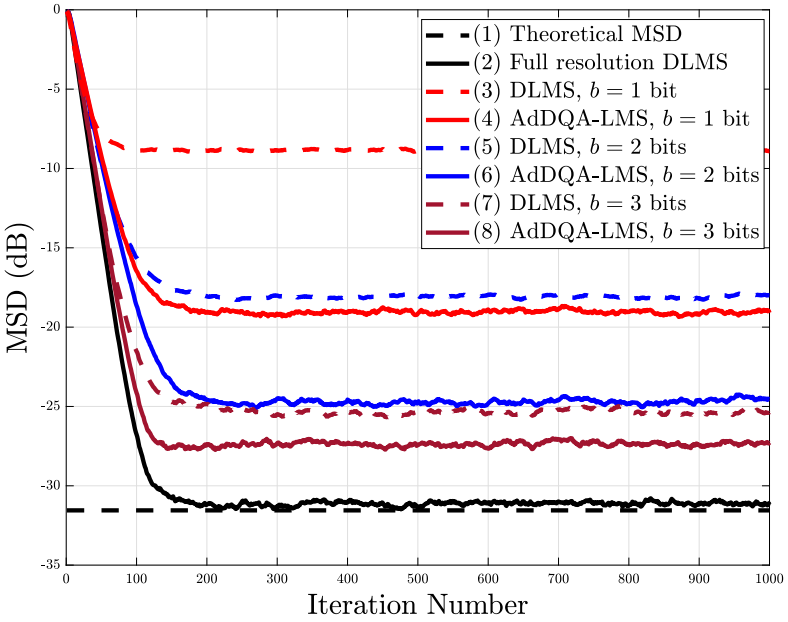


Figure 3.2: MSD curves for DLMS and AdDQA-LMS algorithms.

The simulated mean-square deviation (MSD) learning curves are obtained by ensemble averaging over 100 independent trials. We choose the same step sizes for all agents, i.e., $\mu_k = 0.05$. The combining coefficients a_{lk} are computed by the Metropolis rule (2-8).

The evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}[\|\widetilde{\mathbf{W}}(i)\|^2]$, for the ATC diffusion strategy with different numbers of bits is shown in Figure 3.2. The theoretical MSD of the DLMS with the same step size μ and the Metropolis rule applied to a_{lk} is approximated by $\frac{\mu M}{N^2}\sum_{k=1}^{N}\sigma^2_{v_k}$ [41] and shown by curve 1. Curve 2 shows the standard DLMS performance assuming full resolution ADCs to perform system identification. Curves 3, 5 and 7 show the MSD evolution of the standard DLMS with low resolution signals coarsely quantized with 1, 2 and 3 bits, respectively. Curves 4, 6 and 8 show the MSD performance of the proposed AdDQA-LMS algorithm that improves the MSD for coarsely quantized signals. The performance of the proposed AdDQA-LMS algorithm is closer to the DLMS while it reduces about 90% of the power consumption of

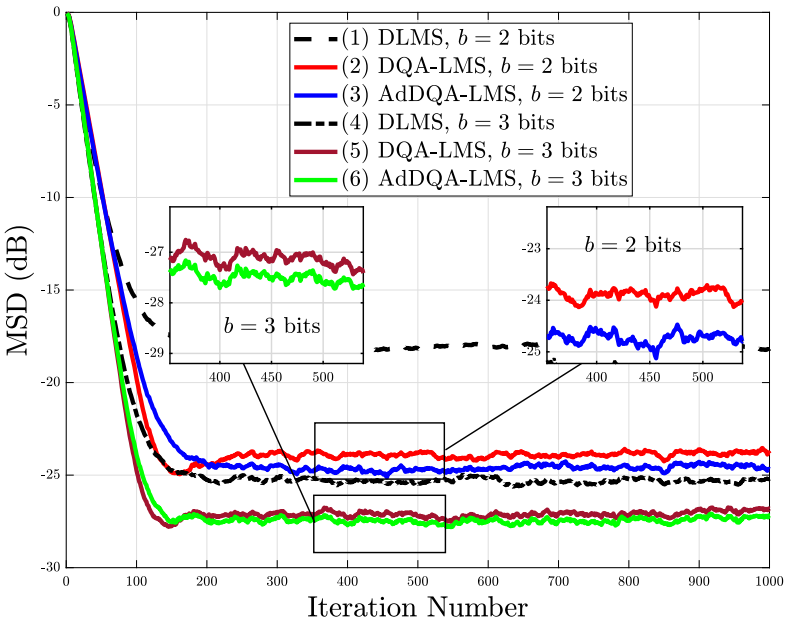


Figure 3.3: The MSD curves for the DLMS, DQA-LMS, and AdDQA-LMS algorithms.

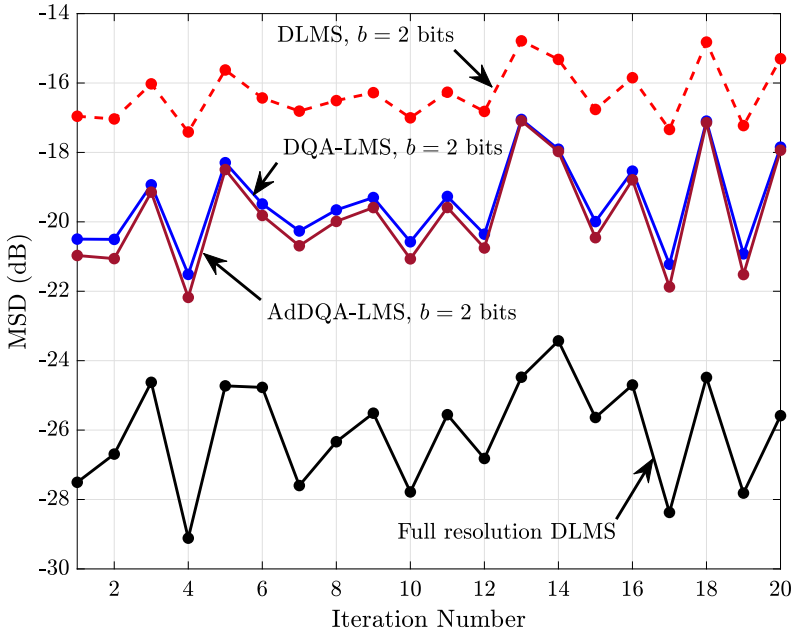


Figure 3.4: The steady state MSD curves for the DLMS, DQA-LMS, and AdDQA-LMS algorithms.

ADCs in the network (see Figure 2.3).

In the next experiment, the MSD learning curves of the proposed AdDQA-LMS and DQA-LMS are compared and the learning performance of the algorithms is shown in Figure 3.3. It can be seen that the AdDQA-LMS improves the estimation performance of the DQA-LMS while both outperform the standard DLMS with low resolution signals coarsely quantized. Figure 3.4 shows the node-wise steady state MSD values of the proposed AdDQA-LMS, DQA-LMS, and standard DLMS, by averaging the MSD values over the last 200 time samples.

To investigate the performance of the DQA-LMS algorithm for the system identification setup with medium colored input, we use the same setup as in Figure 3.1 and the input signals $\mathbf{x}_k(i)$ at each node are generated by passing a white Gaussian noise process with variance $\sigma_{x_k}^2$ through a first-order auto-regressive model with transfer function $\frac{1}{1-r_{x_k}z^{-1}}$ where $r_{x_k} \in (0.3, 0.5)$ are the correlation coefficients. As we can see in Figure 3.5, the DQA-LMS algorithm improves the MSD performance as compared to the DLMS algorithm for coarsely quantized signals with medium colored inputs too.

In the next examples, we generate a network with $N = 10$ nodes to evaluate the MSD performance of DLMS and DQA-LMS algorithms 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 $\pm 20\,\%$. The step-size is set to 0.2 for 1-bit quantization and 0.05 for other curves. Figure 3.6 plots the network details and the results are shown in Figures 3.7 and 3.8. In particular, Figure 3.8 compares the simulation results with those obtained by the analytical expression in (3-34). The results in Figure 3.8 indicate that the theoretical and simulated results agree well especially for $b = 3$ and $b = 4$ bits, and for low to moderate values of SNR, confirming the validity of the theoretical development. The node-wise theoretical and experimental MSD values for different nodes and a moderate SNR are

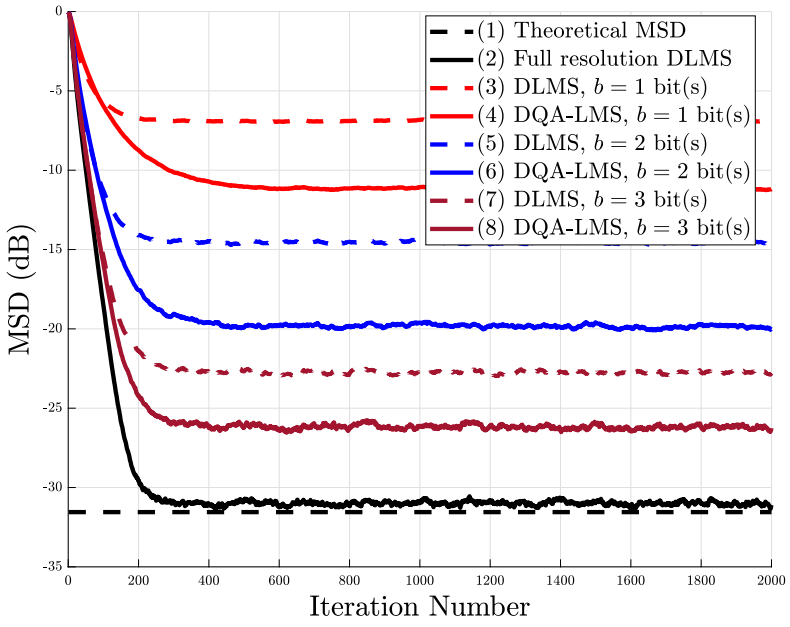
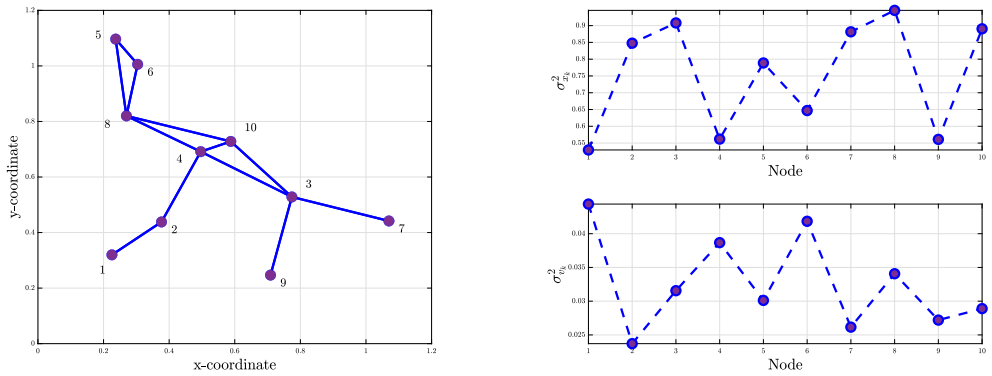


Figure 3.5: The MSD curves for the DLMS and DQA-LMS algorithms and colored inputs.

compared in Figure 3.9. The results shown in Figure 3.9 show that the MSD theoretical expression in (3-33) can accurately predict the MSD values.

3.8 Chapter Summary

In this chapter, we have proposed an energy-efficient framework for distributed learning and developed the DQA-LMS algorithm along with bias compensation strategies for IoT networks. The DQA-LMS algorithm has comparable computational complexity to the standard DLMS algorithm while it substantially reduces the power consumption of the sensors in the network by using low-resolution ADCs. We also proposed the AdDQA-LMS that iteratively computes the bias compensation term and is a suitable choice for the systems with delay line inputs. Simulations have shown excellent performance of DQA-LMS and AdDQA-LMS as compared to DLMS for coarsely quantized signals.



(a) Distributed network structure (b) Network statistical settings

Figure 3.6: A wireless network with $N = 10$ nodes.

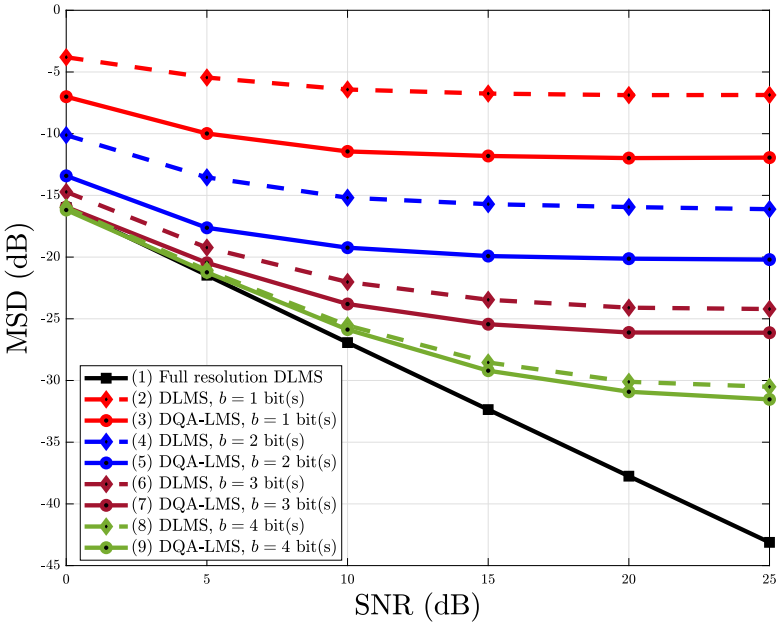


Figure 3.7: Steady-state MSD values for the DLMS and DQA-LMS algorithms for different SNR values.

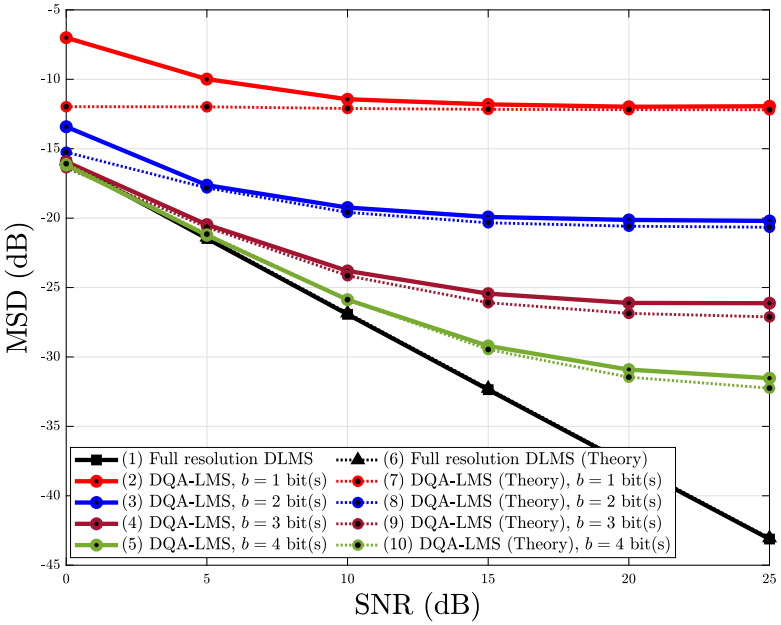


Figure 3.8: Steady-state and theoretical MSD values for the DLMS and DQA-LMS algorithms for different SNR values.

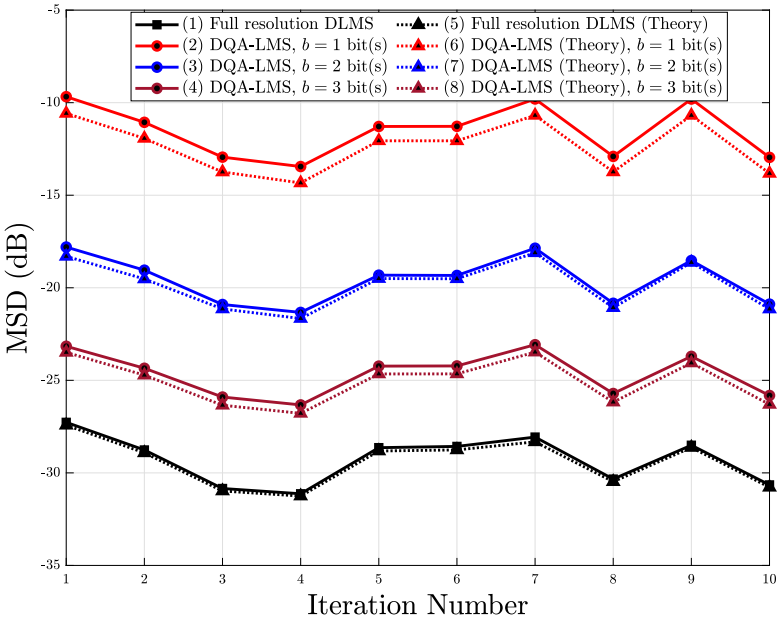


Figure 3.9: Node-wise Steady-state and theoretical MSD values for the DLMS and DQA-LMS algorithms.

4
Distributed Quantization-Aware RLS Algorithm

In this chapter, we propose a distributed quantization-aware recursive least-squares (DQA-RLS) algorithm. We start with the derivation of the proposed DQA-RLS algorithm and 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 of the proposed and existing algorithms. In the end, we assess the estimation performance of the DQA-RLS algorithm for a distributed system identification setup with numerical results.

4.1
Derivation of DQA-RLS

We consider $\mathbf{x}_k(t)$ and $d_k(t)$ as the 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 $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ denote the coarsely quantized versions of $\mathbf{x}_k(i)$ and $d_k(i)$, respectively.

We show next that a learning algorithm based directly on (2-10) is biased for estimating $\mathbf{w}_o$ and show how to correct for this bias. To this end, let $\boldsymbol{\beta}_k(i)$ be an $M \times M$ bias-compensation matrix to be computed and define $\widehat{d}_k(i) = \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)$. Thus, the error is given by

$$e_k(i) = \overline{d}_k(i) - \widehat{d}_k(i) = \overline{d}_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i). \tag{4-1}$$

As seen in (4-1), $e_k(i)$ is different from $e_k(i) = d_k(i) - \mathbf{w}_k^*(i-1)\mathbf{x}_k(i)$ in the diffusion RLS (DRLS) [15].

Let us consider a network of N nodes distributed over an area as in Fig. 2.1. At time i, we globally collect the quantized input regressors into a

matrix $\overline{\mathbf{X}}_i$, the quantized desired signal into vector $\overline{\mathbf{d}}_i$, the noise samples into vector $\mathbf{v}_i$, and the bias-compensation coefficients into the matrix $\boldsymbol{\beta}_i$ over all nodes as follows:

$$\begin{aligned}
\overline{\mathbf{X}}_i &= \text{blockdiag}\{\overline{\mathbf{x}}_1(i), \ldots, \overline{\mathbf{x}}_N(i)\} && (MN \times N) \\
\overline{\mathbf{d}}_i &= \text{col}\{\overline{d}_1(i), \ldots, \overline{d}_N(i)\} && (N \times 1) \\
\mathbf{v}_i &= \text{col}\{v_1(i), \ldots, v_N(i)\} && (N \times 1) \\
\boldsymbol{\beta}_i &= [\boldsymbol{\beta}_1(i), \ldots, \boldsymbol{\beta}_N(i)] && (M \times MN).
\end{aligned} \tag{4-2}$$

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

$$\mathbf{R}_v = \mathbb{E}[\mathbf{v}_i \mathbf{v}_i^*] = \text{diag}\{\sigma_{v_1}^2, \ldots, \sigma_{v_N}^2\} \qquad (N \times N). \tag{4-3}$$

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

$$\begin{aligned}
\overline{\boldsymbol{X}}_i &= \text{blockdiag}\{\overline{\mathbf{X}}_i, \ldots, \overline{\mathbf{X}}_0\} && (MN(i+1) \times N(i+1)), \\
\overline{\boldsymbol{d}}_i &= \text{col}\{\overline{\mathbf{d}}_i, \ldots, \overline{\mathbf{d}}_0\}^T && (1 \times N(i+1)), \\
\boldsymbol{v}_i &= \text{col}\{\mathbf{v}_i, \ldots, \mathbf{v}_0\}^T && (1 \times N(i+1)), \\
\boldsymbol{B}_i &= [\boldsymbol{\beta}_i, \ldots, \boldsymbol{\beta}_0] && (M \times MN(i+1)),
\end{aligned} \tag{4-4}$$

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

$$\min_{\mathbf{w}} \left[\left\| \mathbf{w} - \check{\mathbf{w}} \right\|_{\boldsymbol{\Pi}_i}^2 + \left\| \overline{\boldsymbol{d}}_i - \mathbf{w}^* \boldsymbol{B}_i \overline{\boldsymbol{X}}_i \right\|_{\boldsymbol{\Sigma}_i}^2 \right], \tag{4-5}$$

where $\check{\mathbf{w}}$ is a given column vector, usually $\check{\mathbf{w}} = \mathbf{0}$, $\boldsymbol{\Pi}_i > 0$ is an $M \times M$ positive-definite matrix that incorporates a regularization term $\|\mathbf{w} - \check{\mathbf{w}}\|_{\boldsymbol{\Pi}_i}^2$ 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

$$\boldsymbol{\Pi}_i = \lambda^{i+1}\boldsymbol{\Pi} \qquad \text{and} \qquad \boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i, \tag{4-6}$$

where $0 \ll \lambda < 1$, $\boldsymbol{\Pi} = \delta^{-1}\mathbf{I}_M$ with $\delta > 0$ as a positive constant, and $\boldsymbol{\Lambda}_i \triangleq \text{diag}\{\mathbf{I}_N, \lambda\mathbf{I}_N, \ldots, \lambda^i\mathbf{I}_N\}$. 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 (4-6).

We can rewrite (4-5) in the equivalent form

$$\min_{\mathbf{w}} \Big[(\mathbf{w}-\check{\mathbf{w}})^*\boldsymbol{\Pi}_i(\mathbf{w}-\check{\mathbf{w}}) + (\overline{\boldsymbol{d}}_i - \mathbf{w}^*\boldsymbol{B}_i\overline{\boldsymbol{X}}_i)\boldsymbol{\Sigma}_i(\overline{\boldsymbol{d}}_i - \mathbf{w}^*\boldsymbol{B}_i\overline{\boldsymbol{X}}_i)^*\Big] \tag{4-7}$$

To solve (4-7), we reduce it to the standard least-squares form by introducing the eigendecompositions of $\boldsymbol{\Pi}_i$ as follows

$$\boldsymbol{\Pi}_i = \boldsymbol{\Psi}_i\boldsymbol{\Delta}_i\boldsymbol{\Psi}_i^*, \tag{4-8}$$

where $\boldsymbol{\Delta}_i$ is diagonal with positive entries, and $\boldsymbol{\Psi}_i$ is 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

$$\mathbf{a} \triangleq \overline{\boldsymbol{d}}_i\boldsymbol{\Lambda}_i^{1/2}, \qquad \text{and} \qquad \mathbf{L} \triangleq \boldsymbol{B}_i\overline{\boldsymbol{X}}_i\boldsymbol{\Lambda}_i^{1/2}. \tag{4-9}$$

Using (4-9), we can rewrite (4-7) as follows

$$\min_{\mathbf{w}} \Big[(\mathbf{w}-\check{\mathbf{w}})^*\boldsymbol{\Pi}_i(\mathbf{w}-\check{\mathbf{w}}) + \|\mathbf{a}-\mathbf{w}^*\mathbf{L}\|^2\Big]. \tag{4-10}$$

Defining the change of variables

$$\mathbf{z} \triangleq \mathbf{w}-\check{\mathbf{w}} \qquad \text{and} \qquad \mathbf{f} \triangleq \mathbf{a}-\check{\mathbf{w}}^*\mathbf{L}, \tag{4-11}$$

we can rewrite (4-10) as follows

$$\min_{\mathbf{z}} \left[\mathbf{z}^*\mathbf{\Pi}_i\mathbf{z} + \|\mathbf{f} - \mathbf{z}^*\mathbf{L}\|^2\right]. \tag{4-12}$$

Let $\mathbf{\Delta}_i^{1/2}$ denote the diagonal matrix whose entries are equal to the positive square roots of the entries of $\mathbf{\Delta}_i$. Then using the eigendecomposition in (4-8), we can write the equivalent form of (4-12) as follows

$$\min_{\mathbf{z}} \left[\left\| \begin{bmatrix} \mathbf{0} & \mathbf{f} \end{bmatrix} - \mathbf{z}^* \begin{bmatrix} \mathbf{\Psi}_i^* \mathbf{\Delta}_i^{1/2} & \mathbf{L} \end{bmatrix} \right\|^2\right], \tag{4-13}$$

which is a form of the standard least-squares

$$\min_{\mathbf{w}} \left[\left\|\mathbf{y} - \mathbf{w}^*\mathbf{H}\right\|^2\right], \tag{4-14}$$

in which the roles of $\mathbf{y}$ and $\mathbf{H}$ are played by $\begin{bmatrix} \mathbf{0} & \mathbf{f} \end{bmatrix}$ and $\begin{bmatrix} \mathbf{\Psi}_i^* \mathbf{\Delta}_i^{1/2} & \mathbf{L} \end{bmatrix}$, respectively. All solutions $\widehat{\mathbf{w}}$ to the least-squares problem (4-14) are characterized as solutions to the linear system of equations

$$\widehat{\mathbf{w}}^*\mathbf{H}\mathbf{H}^* = \mathbf{y}\mathbf{H}^*. \tag{4-15}$$

We replace $\mathbf{y}$ and $\mathbf{H}$ with $\begin{bmatrix} \mathbf{0} & \mathbf{f} \end{bmatrix}$ and $\begin{bmatrix} \mathbf{\Psi}_i^* \mathbf{\Delta}_i^{1/2} & \mathbf{L} \end{bmatrix}$ in (4-15) to form the solutions $\widehat{\mathbf{z}}$ to (4-12) as follows

$$\left(\begin{bmatrix} \mathbf{0} & \mathbf{f} \end{bmatrix} - \widehat{\mathbf{z}}^* \begin{bmatrix} \mathbf{\Psi}_i^* \mathbf{\Delta}_i^{1/2} & \mathbf{L} \end{bmatrix} \right) \begin{bmatrix} \mathbf{\Psi}_i^* \mathbf{\Delta}_i^{1/2} & \mathbf{L} \end{bmatrix}^* = 0. \tag{4-16}$$

Using $\widehat{\mathbf{z}} = \widehat{\mathbf{w}} - \check{\mathbf{w}}$ and (4-11) in (4-16), we can write the solution $\widehat{\mathbf{w}}$ to (4-12) in a system of equations as follows

$$(\widehat{\mathbf{w}} - \check{\mathbf{w}})^*[\mathbf{\Pi}_i + \mathbf{L}\mathbf{L}^*] = (\mathbf{a} - \check{\mathbf{w}}^*\mathbf{L})\mathbf{L}^*, \tag{4-17}$$

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

$$\mathbf{w}(i) = \check{\mathbf{w}} + [\mathbf{\Pi}_i + \mathbf{L}\mathbf{L}^*]^{-1}\mathbf{L}(\mathbf{a} - \check{\mathbf{w}}^*\mathbf{L})^*. \tag{4-18}$$

Using (4-9), the solution $\mathbf{w}(i)$ to the weighted regularized least-squares (4-5) is given by

$$\mathbf{w}(i) = \check{\mathbf{w}} + \left(\boldsymbol{\Pi}_i + \boldsymbol{B}_i\overline{\boldsymbol{X}}_i\boldsymbol{\Sigma}_i\overline{\boldsymbol{X}}_i^*\boldsymbol{B}_i^*\right)^{-1}\boldsymbol{B}_i\overline{\boldsymbol{X}}_i\boldsymbol{\Sigma}_i\left(\overline{\boldsymbol{d}}_i - \check{\mathbf{w}}^*\boldsymbol{B}_i\overline{\boldsymbol{X}}_i\right)^*. \tag{4-19}$$

To simplify, we use (4-6), assume the given column vector $\check{\mathbf{w}} = \mathbf{0}$, and write an exponentially weighted version of (4-19) as follows

$$\mathbf{w}(i) = \left(\lambda^{i+1}\boldsymbol{\Pi} + \boldsymbol{B}_i\overline{\boldsymbol{X}}_i\boldsymbol{\Lambda}_i\overline{\boldsymbol{X}}_i * \boldsymbol{B}_i^*\right)^{-1}\boldsymbol{B}_i\overline{\boldsymbol{X}}_i\boldsymbol{\Lambda}_i\overline{\boldsymbol{d}}_i^*. \tag{4-20}$$

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

$$\boldsymbol{P}_i = \left(\lambda^{i+1}\boldsymbol{\Pi} + \boldsymbol{B}_i\overline{\boldsymbol{X}}_i\boldsymbol{\Lambda}_i\overline{\boldsymbol{X}}_i^*\boldsymbol{B}_i^*\right)^{-1}, \tag{4-21}$$

and write down (4-20) as follows

$$\mathbf{w}(i) = \boldsymbol{P}_i\boldsymbol{B}_i\overline{\boldsymbol{X}}_i\boldsymbol{\Lambda}_i\overline{\boldsymbol{d}}_i^*. \tag{4-22}$$

We then reformulate the global least-squares solution in (4-22) as a local least-squares solution where nodes have access to limited data from their neighbors in the diffusion scheme [15]. In this scheme, nodes update their local intermediate estimates $\mathbf{h}_k(i)$ following (4-22) which yields

$$\mathbf{h}_k(i) = \mathbf{P}_k(i)\boldsymbol{B}_k(i)\overline{\boldsymbol{X}}_k(i)\boldsymbol{\Lambda}_k(i)\overline{\boldsymbol{d}}_k^*(i), \tag{4-23}$$

where

$$\mathbf{P}_k(i) = \left(\lambda^{i+1}\boldsymbol{\Pi} + \boldsymbol{B}_k(i)\overline{\boldsymbol{X}}_k(i)\boldsymbol{\Lambda}_k(i)\overline{\boldsymbol{X}}_k^*(i)\boldsymbol{B}_k^*(i)\right)^{-1},$$

and $\overline{\boldsymbol{X}}_k(i)$, $\overline{\boldsymbol{d}}_k(i)$, and $\boldsymbol{B}_k(i)$ 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

$$\mathbf{w}_k(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\mathbf{h}_l(i), \tag{4-24}$$

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

$$a_{lk} = 0 \text{ if } l \notin \mathcal{N}_k, a_{lk} > 0 \text{ if } l \in \mathcal{N}_k, \text{ and } \sum_{l \in \mathcal{N}_k} a_{lk} = 1. \tag{4-25}$$

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

$$\begin{aligned}\mathbf{P}_k^{-1}(i) &= \lambda\Big(\lambda^i \boldsymbol{\Pi} + \boldsymbol{B}_k(i-1)\overline{\boldsymbol{X}}_k(i-1)\boldsymbol{\Lambda}_k(i-1)\overline{\boldsymbol{X}}_k^*(i-1)\boldsymbol{B}_k^*(i-1)\Big) \\ &\quad + \boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i) \\ &= \lambda\mathbf{P}_k^{-1}(i-1) + \boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i).\end{aligned} \tag{4-26}$$

Using the matrix inversion lemma

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

and the assignments

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

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

$$\mathbf{P}_k(i) = \lambda^{-1}\left(\mathbf{P}_k(i-1) - \frac{\lambda^{-1}\mathbf{P}_k(i-1)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i)\mathbf{P}_k(i-1)}{1 + \lambda^{-1}\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i)\mathbf{P}_k(i-1)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)}\right). \tag{4-27}$$

To form the recursion for $\mathbf{h}_k(i)$, we rewrite (4-23) as follows

$$\begin{aligned}\mathbf{h}_k(i) &= \mathbf{P}_k(i)\Big(\boldsymbol{B}_k(i-1)\overline{\boldsymbol{X}}_k(i-1)\lambda\boldsymbol{\Lambda}_k(i-1)\overline{\boldsymbol{d}}_k^*(i-1) + \boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{d}_k(i)\Big) \\ &= \mathbf{P}_k(i)\Big(\mathbf{P}_k^{-1}(i-1)\mathbf{P}_k(i-1)\boldsymbol{B}_k(i-1)\overline{\boldsymbol{X}}_k(i-1)\lambda\boldsymbol{\Lambda}_k(i-1)\overline{\boldsymbol{d}}_k^*(i-1) \\ &\qquad + \boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{d}_k^*(i)\Big)\end{aligned} \tag{4-28}$$

From (4-26), we have $\mathbf{P}_k^{-1}(i-1) = \lambda^{-1}\Big(\mathbf{P}_k^{-1}(i) - \boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i)\Big)$. Then considering $\mathbf{h}_k(i-1) = \mathbf{P}_k(i-1)\boldsymbol{B}_k(i-1)\overline{\boldsymbol{X}}_k(i-1)\boldsymbol{\Lambda}_k(i-1)\overline{\boldsymbol{d}}_k^*(i-1)$ from (4-23) and the fact that $\mathbf{P}_k$ is a symmetric matrix, we can write (4-28) as follows

$$\begin{aligned}\mathbf{h}_k(i) &= \mathbf{P}_k(i)\mathbf{P}_k^{-1}(i-1)\lambda\mathbf{h}_k(i-1) + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{d}_k^*(i)\\ &= \mathbf{h}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i)\mathbf{h}_k(i-1) + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{d}_k^*(i)\\ &= \mathbf{h}_k(i-1) + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\Big(\overline{d}_k^*(i) - \overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i)\mathbf{h}_k(i-1)\Big).\end{aligned} \tag{4-29}$$

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

$$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)e_k^*(i), \tag{4-30}$$

where

$$e_k(i) = \overline{d}_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i) \tag{4-31}$$

The next section shows how the bias compensation $\boldsymbol{\beta}_k(i)$ should be chosen such that (4-24) is asymptotically unbiased in the mean.

4.2 Essentials of Analysis

In this section, we present some assumptions and definitions for the statistical analysis of the proposed DQA-RLS algorithm that focuses on the mean and mean-square performances.

Combining (4-6) with (4-26), we can write $\mathbf{P}_k^{-1}(i)$ as follows:

$$\begin{aligned}\mathbf{P}_k^{-1}(i) &= \lambda\mathbf{P}_k^{-1}(i-1) + \boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^*(i)\\ &= \lambda^{i+1}\boldsymbol{\Pi} + \sum_{j=0}^{i}\lambda^{i-j}\boldsymbol{\beta}_k(j)\overline{\mathbf{x}}_k(j)\overline{\mathbf{x}}_k^*(j)\boldsymbol{\beta}_k^*(j).\end{aligned} \tag{4-32}$$

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

$$\lim_{i\to+\infty}\mathbb{E}[\mathbf{P}_k^{-1}(i)] = \frac{\mathbb{E}\Big[\boldsymbol{\beta}_k(j)\overline{\mathbf{x}}_k(j)\overline{\mathbf{x}}_k^*(j)\boldsymbol{\beta}_k^*(j)\Big]}{1-\lambda} \triangleq \mathbf{P}_k^{-1}. \tag{4-33}$$

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 $\sigma_{v_k}^2$ and the quantized regressors $\overline{\mathbf{x}}_k(i)$ with covariance matrices $\mathbf{R}_{\overline{x}_k} = \mathbb{E}[\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)]$. 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 [45, 61]. Note that this a common assumption for the performance analysis of RLS-type algorithms for λ close to unity. This also applies to $\mathbf{G}_{x_k}$ since (2-33) 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 $\mathbf{P}_k^{-1}(i)$ can be replaced by their expected values, $\mathbb{E}[\mathbf{P}_k(i)]$ and $\mathbb{E}[\mathbf{P}_k^{-1}(i)]$. Note that such replacements are commonly used in the performance analysis of RLS-type algorithms, see [15, 21, 28, 58], i.e.,

$$\mathbb{E}[\mathbf{P}_k(i)] \approx \mathbf{P}_k \text{ and } \mathbb{E}[\mathbf{P}_k^{-1}(i)] \approx \mathbf{P}_k^{-1}. \tag{4-34}$$

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

$$\widetilde{\mathbf{w}}_k(i) = \mathbf{w}_o - \mathbf{w}_k(i), \qquad \text{and} \qquad \widetilde{\mathbf{h}}_k(i) = \mathbf{w}_o - \mathbf{h}_k(i). \tag{4-35}$$

The performance of bias-compensated-type adaptive filters is usually compared in terms of the mean-square deviation (MSD) [22], defined by

$$\text{MSD}_k \triangleq \lim_{i \to +\infty} \mathbb{E}[\|\widetilde{\mathbf{w}}_k(i)\|^2] = \mathbb{E}\Big[\,\text{Tr}\left(\widetilde{\mathbf{w}}_k(i)\widetilde{\mathbf{w}}_k^*(i)\right)\Big]. \tag{4-36}$$

We also define the following matrices:

$$\begin{aligned} \boldsymbol{\mathcal{A}} &= \mathbf{A} \otimes \mathbf{I}_M \qquad (MN \times MN) \\ \widetilde{\mathbf{W}}_i &= \operatorname{col}\left\{\widetilde{\mathbf{w}}_1(i), \ldots, \widetilde{\mathbf{w}}_N(i)\right\} \quad (MN \times 1). \end{aligned} \tag{4-37}$$

4.3 Mean Performance Analysis

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 $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ as

$$\overline{\mathbf{x}}_k(i) = \mathbf{G}_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i) \tag{4-38a}$$

$$\overline{d}_k(i) = g_{d_k}d_k(i) + q_{d_k}(i) = g_{d_k}\mathbf{w}_o^*\mathbf{x}_k(i) + p_k(i), \tag{4-38b}$$

where $p_k(i) = g_{d_k}v_k(i) + q_{d_k}(i)$ is uncorrelated with $\mathbf{x}_k(i)$. Note that $\mathbf{G}_{x_k}$ and g_{d_k} are built based on $\mathbf{x}_k(i)$ and $d_k(i)$, respectively. Using the decomposition model in (4-38a), we can rewrite the error in (4-1) as follows

$$\begin{aligned} e_k(i) &= \overline{d}_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_k(i)\overline{\mathbf{x}}_k(i) \\ &= g_{d_k}\mathbf{w}_o^*\mathbf{x}_k(i) + p_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_k(i)\Big(\mathbf{G}_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i)\Big) \\ &= g_{d_k}\mathbf{w}_o^*\mathbf{x}_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i) - \mathbf{w}_k^*(i-1)\boldsymbol{\beta}_k(i)\mathbf{q}_{x_k}(i) + p_k(i). \end{aligned} \tag{4-39}$$

Combining (4-39) with (4-30) and subtracting from $\mathbf{w}_o$ yields

$$
\begin{aligned}
\widetilde{\mathbf{h}}_k(i) &= \widetilde{\mathbf{w}}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\Big(\mathbf{G}_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i)\Big) \\
&\quad \Big(\mathbf{x}_k^*(i)g_{d_k}\mathbf{w}_o - \mathbf{x}_k^*(i)\mathbf{G}_{x_k}\boldsymbol{\beta}_k^*(i)\mathbf{w}_k(i-1) - \mathbf{q}_{x_k}^*(i)\boldsymbol{\beta}_k^*(i)\mathbf{w}_k(i-1) + p_k^*(i)\Big) \\
&= \widetilde{\mathbf{w}}_k(i-1) + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{G}_{x_k}\boldsymbol{\beta}_k^*(i)\mathbf{w}_k(i-1) \\
&\quad + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\boldsymbol{\beta}_k^*(i)\mathbf{w}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{d_k}\mathbf{w}_o \\
&\quad + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{q}_{x_k}^*(i)\boldsymbol{\beta}_k^*(i)\mathbf{w}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i)p_k^*(i) \\
&\quad + \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\mathbf{G}_{x_k}\boldsymbol{\beta}_k^*(i)\mathbf{w}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)g_{d_k}\mathbf{w}_o \\
&\quad - \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{q}_{x_k}(i)p_k^*(i).
\end{aligned}
\tag{4-40}
$$

The errors p_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 (4-40), we obtain

$$
\begin{aligned}
\mathbb{E}\Big[\widetilde{\mathbf{h}}_k(i)\Big] &= \mathbb{E}\Big[\widetilde{\mathbf{w}}_k(i-1)\Big] - \mathbb{E}\Big[\mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{d_k}\Big]\mathbf{w}_o \\
&\quad + \mathbb{E}\Big[\mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\Big(\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{G}_{x_k} + \mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big)\boldsymbol{\beta}_k^*(i)\Big]\mathbb{E}\Big[\mathbf{w}_k(i-1)\Big].
\end{aligned}
\tag{4-41}
$$

Let us define two $M \times M$ matrices that include terms multiplied by $\mathbf{w}_k(i-1)$ and $\mathbf{w}_o$ in (4-41) as follows:

$$
\begin{aligned}
\boldsymbol{\Theta}_k(i) &\triangleq \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\Big(\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{G}_{x_k} + \mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big)\boldsymbol{\beta}_k^*(i) \\
\boldsymbol{\Gamma}_k(i) &\triangleq \mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{d_k}.
\end{aligned}
\tag{4-42}
$$

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

$$
\mathbb{E}\Big[\boldsymbol{\Theta}_k(i)\Big] = \mathbb{E}\Big[\boldsymbol{\Gamma}_k(i)\Big],
\tag{4-43}
$$

and we show in the next section that this condition is possible by appropriately choosing $\boldsymbol{\beta}_k(i)$. Assuming (4-43), we can write (4-41) as follows

$$\mathbb{E}\Big[\widetilde{\mathbf{h}}_k(i)\Big] = \mathbb{E}\Big[\widetilde{\mathbf{w}}_k(i-1)\Big] - \mathbb{E}\Big[\boldsymbol{\Theta}_k(i)\Big]\mathbb{E}\Big[\widetilde{\mathbf{w}}_k(i-1)\Big]. \tag{4-44}$$

From (4-32) and (4-33), we can verify that for sufficiently large i under Assumption 3, we have

$$\mathbb{E}\Big[\boldsymbol{\Theta}_k(i)\Big] = 1 - \lambda. \tag{4-45}$$

We now apply the weight-error vectors to the combined estimates (4-24) and obtain

$$\widetilde{\mathbf{w}}_k(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\widetilde{\mathbf{h}}_l(i), \tag{4-46}$$

and taking the expectation from both sides of it and using (4-44) and (4-45), we arrive at

$$\mathbb{E}\Big[\widetilde{\mathbf{w}}_k(i)\Big] = \sum_{l\in\mathcal{N}_k} a_{lk}(1-\lambda)\mathbb{E}\Big[\widetilde{\mathbf{w}}_k(i-1)\Big]. \tag{4-47}$$

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

$$\mathbb{E}[\widetilde{\mathbf{W}}_i] = (1-\lambda)\boldsymbol{\mathcal{A}}\mathbb{E}[\widetilde{\mathbf{W}}_{i-1}] = (1-\lambda)^{i-i_0}\boldsymbol{\mathcal{A}}^{i-i_0}\mathbb{E}[\widetilde{\mathbf{W}}_{i_0}]. \tag{4-48}$$

Observing (4-48) and recalling that the spectral radius of $\boldsymbol{\mathcal{A}}$ (i.e., the largest eigenvalue in modulus) is equal to one [41], 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 (4-25), for $0 \ll \lambda < 1$, the term on the right-hand side of (4-48) converges to zero and DQA-RLS is asymptotically unbiased in the mean.

4.4 Bias Compensation

In this section, we derive an expression for the bias compensation term $\boldsymbol{\beta}_k(i)$ such that (4-43) is true and (4-24) is asymptotically unbiased in the mean. From (4-42) and (4-43), we must have

$$\begin{aligned}\mathbb{E}\Big[\mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\Big(\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{G}_{x_k} + \mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big)\boldsymbol{\beta}_k^*(i)\Big] \\ = \mathbb{E}\Big[\mathbf{P}_k(i)\boldsymbol{\beta}_k(i)\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{d_k}\Big].\end{aligned} \tag{4-49}$$

Under Assumptions 2 and 3, we can write (4-49) as follows

$$\Big(\mathbf{G}_{x_k}\mathbb{E}\Big[\mathbf{x}_k(i)\mathbf{x}_k^*(i)\Big]\mathbf{G}_{x_k}+\mathbb{E}\Big[\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big]\Big)\boldsymbol{\beta}_k^*(i)=\mathbf{G}_{x_k}\mathbb{E}\Big[\mathbf{x}_k(i)\mathbf{x}_k^*(i)\Big]g_{d_k}. \tag{4-50}$$

Therefore, the bias compensation term is expressed by

$$\boldsymbol{\beta}_k(i)=\boldsymbol{\beta}_k=g_{d_k}\mathbf{R}_{x_k}\mathbf{G}_{x_k}\Big(\mathbf{G}_{x_k}\mathbf{R}_{x_k}\mathbf{G}_{x_k}+\mathbf{R}_{q_x,k}\Big)^{-1}, \tag{4-51}$$

which needs an $M\times M$ matrix inversion at each time instant i if $\mathbf{x}_k(i)$ are nonstationary and, moreover, we realize $\boldsymbol{\beta}_k\in\mathbb{R}^{M\times M}$ 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 $\mathbf{G}_{x_k}$ reduces to $g_{x_k}\mathbf{I}_M$ with g_{x_k} estimated as follows

$$\widehat{g}_{x_k}(i)=\frac{1}{\sqrt{\widehat{\sigma}_{x_k}^2}}\sum_{j=0}^{2^b-1}\frac{l_j}{\sqrt{\pi}}\Bigg(e^{-\frac{\tau_j^2}{\widehat{\sigma}_{x_k}^2}}-e^{-\frac{\tau_{j+1}^2}{\widehat{\sigma}_{x_k}^2}}\Bigg), \tag{4-52}$$

where $\widehat{\sigma}_{x_k}^2$ is an instantaneous approximation to $\sigma_{x_k}^2$ given by:

$$\widehat{\sigma}_{x_k}^2=\widehat{\sigma}_{\overline{x}_k}^2+\widehat{\sigma}_{q_k}^2. \tag{4-53}$$

where $\widehat{\sigma}_{q_k}^2$ is given by (2-35). To compute $\widehat{\sigma}_{\overline{x}_k}^2$ which is given by (3-18), we use the iterative recursion (3-19) to reduce the computational complexity of our algorithm. Therefore, the bias compensation term is given by

$$\boldsymbol{\beta}_k(i)=\beta_k(i)\mathbf{I}_M=\frac{\widehat{g}_{x_k}(i)g_{d_k}(i)\widehat{\sigma}_{x_k}^2}{\widehat{g}_{x_k}^2(i)\widehat{\sigma}_{x_k}^2+\widehat{\sigma}_{q_k}^2}\mathbf{I}_M. \tag{4-54}$$

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} and g_{d_k} can be considered equal and the bias compensation term is given by

Algorithm 4: DQA-RLS algorithm at agent k

input : Initial values $\mathbf{w}_k(-1) \in \mathbb{R}^M$, combination matrix $\mathbf{A}$, $\widehat{\sigma}^2_{\overline{x}_k}(-1) = 0$, $0 \ll \gamma < 1$, $0 \ll \lambda < 1$ and $\mathbf{P}_k(-1) = \mathbf{\Pi}^{-1}$
Generate $\mathcal{T}_b$ and $\mathcal{L}_b$, and Compute $\widehat{\sigma}^2_{q_k}$ from (2-35)

output : $\mathbf{w}_k(i)$

for $i = 0, 1, 2, \ldots$ **do**

data: $\overline{d}_k(i)$ and $\overline{\mathbf{x}}_k(i) = [\overline{x}_k(i), \overline{x}_k(i-1), \ldots, \overline{x}_k(i-M+1)]^T$

$\widehat{\sigma}^2_{\overline{x}_k}(i) = \gamma \widehat{\sigma}^2_{\overline{x}_k}(i-1) + (1-\gamma)|\overline{x}_k(i)|^2$

$\widehat{\sigma}^2_{x_k} = \widehat{\sigma}^2_{\overline{x}_k}(i) + \widehat{\sigma}^2_{q_k}$

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

$\beta_k(i) = \frac{\widehat{g}^2_{x_k}(i)\widehat{\sigma}^2_{x_k}}{\widehat{g}^2_{x_k}(i)\widehat{\sigma}^2_{x_k} + \widehat{\sigma}^2_{q_k}}$

$e_k(i) = \overline{d}_k(i) - \mathbf{w}_k^*(i-1)\beta_k(i)\overline{\mathbf{x}}_k(i)$

$\mathbf{P}_k(i) = \frac{1}{\lambda}\left(\mathbf{P}_k(i-1) - \frac{\mathbf{P}_k(i-1)\beta_k(i)\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\beta_k(i)\mathbf{P}_k(i-1)}{\lambda + \overline{\mathbf{x}}_k^*(i)\beta_k(i)\mathbf{P}_k(i-1)\beta_k(i)\overline{\mathbf{x}}_k(i)} \right)$

$\mathbf{h}_k(i) = \mathbf{w}_k(i-1) + \mathbf{P}_k(i)\beta_k(i)\overline{\mathbf{x}}_k(i)e_k^*(i)$

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

end for

$$\boldsymbol{\beta}_k(i) = \frac{\widehat{g}^2_{x_k}(i)\widehat{\sigma}^2_{x_k}}{\widehat{g}^2_{x_k}(i)\widehat{\sigma}^2_{x_k} + \widehat{\sigma}^2_{q_k}}\mathbf{I}_M. \tag{4-55}$$

This is the simple version of the bias compensation term that we use in the proposed DQA-RLS algorithm which is detailed in Algorithm 4, 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 ($\sigma^2_{x_k} \neq 1$).

4.5 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 (4-40) as

$$\begin{aligned}\widetilde{\mathbf{h}}_k(i) = \mathbf{P}_k(i)\mathbf{P}_k^{-1}(i)\widetilde{\mathbf{w}}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k\overline{\mathbf{x}}_k(i)\overline{d}_k^*(i) \\ + \mathbf{P}_k(i)\boldsymbol{\beta}_k\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^T\mathbf{w}_k(i-1).\end{aligned} \tag{4-56}$$

Using (4-32), we obtain

$$\begin{aligned}\widetilde{\mathbf{h}}_k(i) &= \mathbf{P}_k(i)\lambda\mathbf{P}_k^{-1}(i-1)\widetilde{\mathbf{w}}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k\overline{\mathbf{x}}_k(i)\overline{d}_k^*(i) \\ &\quad + \mathbf{P}_k(i)\boldsymbol{\beta}_k\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^T\Big(\widetilde{\mathbf{w}}_k(i-1) + \mathbf{w}_k(i-1)\Big) \\ &= \lambda\widetilde{\mathbf{w}}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k\overline{\mathbf{x}}_k(i)\overline{d}_k^*(i) + \mathbf{P}_k(i)\boldsymbol{\beta}_k\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^T\mathbf{w}_o.\end{aligned} \tag{4-57}$$

We now use (4-38a) and (4-38b) to write (4-57) as follows

$$\begin{aligned}\widetilde{\mathbf{h}}_k(i) &= \lambda\widetilde{\mathbf{w}}_k(i-1) - \mathbf{P}_k(i)\boldsymbol{\beta}_k\mathbf{G}_{x_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)g_{d_k}\mathbf{w}_o \\ &\quad - \mathbf{P}_k(i)\boldsymbol{\beta}_k\mathbf{G}_{x_k}\mathbf{x}_k(i)p_k^*(i) - \mathbf{P}_k(i)\boldsymbol{\beta}_k\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)g_{d_k}\mathbf{w}_o \\ &\quad - \mathbf{P}_k(i)\boldsymbol{\beta}_k\mathbf{q}_{x_k}(i)p_k^*(i) + \mathbf{P}_k(i)\boldsymbol{\beta}_k\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\boldsymbol{\beta}_k^T\mathbf{w}_o.\end{aligned} \tag{4-58}$$

In order to simplify this expression, we assume that the choice of $\boldsymbol{\beta}_k$ in (4-55) makes (4-51) approximately true, so that we can approximate the instantaneous values in the second and sixth terms on the RHS of (4-58) to be equal with different signs and to vanish for sufficiently large i (our simulations show that this approximation is reasonable). Therefore, the weight-error vectors of the combined estimates in (4-46) are given by

$$\begin{aligned}\widetilde{\mathbf{w}}_k(i) = \lambda\sum_{l\in\mathcal{N}_k} a_{lk}\widetilde{\mathbf{w}}_l(i-1) - \sum_{l\in\mathcal{N}_k} a_{lk}\mathbf{P}_l\boldsymbol{\beta}_l\mathbf{G}_{x_l}\mathbf{x}_l(i)p_l^*(i) \\ - \sum_{l\in\mathcal{N}_k} a_{lk}\mathbf{P}_l\boldsymbol{\beta}_l\mathbf{q}_{x,l}(i)\mathbf{x}_l^*(i)g_{d_l}\mathbf{w}_o - \sum_{l\in\mathcal{N}_k} a_{lk}\mathbf{P}_l\boldsymbol{\beta}_l\mathbf{q}_{x,l}(i)p_l^*(i).\end{aligned} \tag{4-59}$$

Let us now define

$$
\begin{aligned}
\boldsymbol{\mathcal{P}} &= \text{blockdiag}\Big\{\mathbf{P}_1, \ldots, \mathbf{P}_N\Big\} && (MN \times MN)\\
\boldsymbol{\mathcal{B}} &= \text{blockdiag}\Big\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N\Big\} && (MN \times MN)\\
\boldsymbol{\mathcal{G}} &= \text{blockdiag}\Big\{\mathbf{G}_{x_1}, \ldots, \mathbf{G}_{x_N}\Big\} && (MN \times MN)\\
\boldsymbol{\Upsilon}_i &= \text{blockdiag}\Big\{\mathbf{q}_{x,1}(i)\mathbf{x}_1^*(i), \ldots, \mathbf{q}_{x,N}(i)\mathbf{x}_N^*(i)\Big\} && (MN \times MN)\\
\boldsymbol{\xi}_i &= \text{col}\Big\{\mathbf{x}_1(i)p_1^*(i), \ldots, \mathbf{x}_N(i)p_N^*(i)\Big\} && (MN \times 1)\\
\boldsymbol{\zeta}_i &= \text{col}\Big\{\mathbf{q}_{x,1}(i)p_1^*(i), \ldots, \mathbf{q}_{x,N}(i)p_N^*(i)\Big\} && (MN \times 1)\\
\boldsymbol{\eta} &= \text{col}\Big\{g_{d_1}\mathbf{w}_o, \ldots, g_{d_N}\mathbf{w}_o\Big\} && (MN \times 1),
\end{aligned}
$$

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

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

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}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big]\\
&\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big]\\
&\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big]\\
&\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big]\\
&\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{P}}^*\boldsymbol{\mathcal{A}}^T\Big].
\end{aligned}
\tag{4-60}
$$

We now use the commutative property of the expectation and vectorization operations, and the relationship between the vectorization operation and the Kronecker product, $\text{vec}(ABC) = (C^T \otimes A)\,\text{vec}(B)$, to write (4-60) as follows

$$
\text{vec}(\boldsymbol{\Omega}_i) = \lambda^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\,\text{vec}(\boldsymbol{\Omega}_{i-1}) + \text{vec}(\boldsymbol{\Xi}_i), \tag{4-61}
$$

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 (4-60). Note that $\text{vec}(\boldsymbol{\Omega}_i)$ 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

$$\lim_{i\to+\infty} \operatorname{vec}(\boldsymbol{\Omega}_i) = \Big(\mathbf{I}_{M^2N^2} - \lambda^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}})\Big)^{-1} \operatorname{vec}(\boldsymbol{\Xi}_{+\infty}). \tag{4-62}$$

For sufficiently large i, taking into account Assumption 3 and (4-33), we have

$$\mathbf{P}_k = (1-\lambda)\Big[\boldsymbol{\beta}_k \mathbf{R}_{\overline{x}_k} \boldsymbol{\beta}_k\Big]^{-1},$$

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

$$\begin{aligned}
\operatorname{vec}(\boldsymbol{\Xi}_{+\infty}) = \lim_{i\to+\infty} \operatorname{vec}(\boldsymbol{\Xi}_i) &= (1-\lambda)^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}}) \\
&\operatorname{vec}\Bigg(\boldsymbol{\mathcal{B}}^{-1}\boldsymbol{\mathcal{R}}_{\overline{x}}^{-1}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T + \mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big] + \mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big] \\
&+ \boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big] + \mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T + \boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\Big] \\
&+ \mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T + \mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\zeta}_i^*\Big] + \mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]\Big)\boldsymbol{\mathcal{R}}_{\overline{x}}^{-1}\boldsymbol{\mathcal{B}}^{-1}\Bigg) \\
&\approx (1-\lambda)^2(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{A}}) \operatorname{vec}\Bigg(\boldsymbol{\mathcal{B}}^{-1}\boldsymbol{\mathcal{R}}_{\overline{x}}^{-1}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T \\
&+ \mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big] + \mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\Big)\boldsymbol{\mathcal{R}}_{\overline{x}}^{-1}\boldsymbol{\mathcal{B}}^{-1}\Bigg),
\end{aligned} \tag{4-63}$$

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

$$\begin{aligned}
\boldsymbol{\mathcal{R}}_{\overline{x}} &= \operatorname{blockdiag}\Big\{\mathbf{R}_{\overline{x}_1}, \ldots, \mathbf{R}_{\overline{x}_N}\Big\} \\
\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big] &= \operatorname{blockdiag}\Big\{\mathbf{R}_{x_1}\sigma_{p_1}^2, \ldots, \mathbf{R}_{x_N}\sigma_{p_N}^2\Big\} \\
\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big] &= \operatorname{blockdiag}\Big\{g_{d_1}^2\mathbf{R}_{q_x,1}\Big(\mathbf{w}_o^*\mathbf{R}_{x_1}\mathbf{w}_o\Big), \ldots, g_{d_N}^2\mathbf{R}_{q_x,N}\Big(\mathbf{w}_o^*\mathbf{R}_{x_N}\mathbf{w}_o\Big)\Big\} \\
\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big] &= \operatorname{blockdiag}\Big\{\mathbf{R}_{q_x,1}\sigma_{p_1}^2, \ldots, \mathbf{R}_{q_x,N}\sigma_{p_N}^2\Big\}.
\end{aligned} \tag{4-64}$$

The analytical computation of $\mathbf{R}_{\overline{x}_k}$, $\mathbf{R}_{q_x,k}$, and $\sigma_{p_k}^2$ is detailed in Appendix A. Therefore, the steady-state MSD at node k is given by

$$\begin{aligned}\mathrm{MSD}_k &= \lim_{i\to+\infty}\mathbb{E}[\|\widetilde{\mathbf{w}}_k(i)\|^2] = \mathrm{vec}^T\Big(\mathbf{C}_k\otimes\mathbf{I}_M\Big)\,\mathrm{vec}(\boldsymbol{\Omega}_{+\infty})\\ &= (1-\lambda)^2\,\mathrm{vec}^T\Big(\mathbf{C}_k\otimes\mathbf{I}_M\Big)\Big(\mathbf{I}_{M^2N^2}-\lambda^2(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\Big)^{-1}(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\\ &\qquad \mathrm{vec}\,\Big(\boldsymbol{\mathcal{B}}^{-1}\boldsymbol{\mathcal{R}}_{\overline{x}}^{-1}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\Big)\boldsymbol{\mathcal{R}}_{\overline{x}}^{-1}\boldsymbol{\mathcal{B}}^{-1}\Big),\end{aligned} \tag{4-65}$$

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 (4-61) that, for $0\ll\lambda<1$ and $\boldsymbol{\mathcal{A}}$ with the entries a_{lk} subject to (4-25), 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 $\Big(\mathbf{I}_{M^2N^2}-\lambda^2(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\Big)$ in (4-65) is nonsingular. The global MSD over all the nodes is given by

$$\mathrm{MSD}_{\mathrm{global}} = \frac{1}{N}\sum_{k=1}^{N}\mathrm{MSD}_k = \frac{1}{N}\,\mathrm{Tr}(\boldsymbol{\Omega}_{+\infty}). \tag{4-66}$$

Remark 3: *(High precision signals, $b=\infty$).* Increasing the number of quantization bits, the diagonal entries of $\mathbf{G}_{x_k}$ approach unity where for high precision signals ($b=\infty$) with $\overline{\mathbf{x}}_k=\mathbf{x}_k$, we have $\mathbf{G}_{x_k}=\mathbf{I}_M$ according to (2-33) and $\mathbf{R}_{q_x,k}=\mathbf{0}$ according to (4-38a), and consequently $\boldsymbol{\beta}_k(i)=\mathbf{I}_M$ from (4-55). For $b=\infty$ we also have $q_{d_k}=0$ and $g_{d_k}=1$, and since $p_k(i)=g_{d_k}v_k(i)+q_{d_k}(i)$, thus $\sigma_{p_k}^2=\sigma_{v_k}^2$. Therefore, for high precision signals, the third and fourth definitions in (4-64) vanish, and (4-65) reduces to

$$\begin{aligned}\mathrm{MSD}_k = (1-\lambda)^2\,\mathrm{vec}^T\Big(\mathbf{C}_k\otimes\mathbf{I}_M\Big)\Big(\mathbf{I}_{M^2N^2}-\lambda^2(\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\Big)^{-1}\\ (\boldsymbol{\mathcal{A}}\otimes\boldsymbol{\mathcal{A}})\,\mathrm{vec}\,\Big(\,\mathrm{diag}\Big\{\mathbf{R}_{x_1}^{-1}\sigma_{v,1}^2,\ldots,\mathbf{R}_{x_N}^{-1}\sigma_{v,N}^2\Big\}\Big),\end{aligned} \tag{4-67}$$

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.

4.6 Computational Complexity

Table 4.1 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 $O(M^2)$. Figure 4.1 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 Figure 4.1, 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_{d_k} online since this is more appropriate for non-stationary input data. However, one can compute $\mathbf{G}_{x_k}$ offline if an estimate of $\mathbf{R}_{x_k}$ in (2-33) is available.

4.7 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 $\mathcal{T}_b = \{\tau_0, \ldots, \tau_{2^b}\}$ and labels $\mathcal{L}_b = \{l_0, \ldots, l_{2^b-1}\}$ using the Lloyd-Max algorithm [46, 47]. The input signals $\mathbf{x}_k(i)$ at each node are chosen as Gaussian i.i.d. with a covariance matrix $\mathbf{R}_{x_k} = \mathbf{U}_k \operatorname{diag}\{\mathbf{s}_k\}\mathbf{U}_k^*$ 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

Table 4.1: Computational complexity of DQA-RLS algorithm per time instant at node k

Task	$+-$	$\times$	$\div$	exp
$\widehat{\sigma}^2_{\overline{x}_k}(i)$	2	3	0	0
$\widehat{\sigma}^2_{x_k}(i)$	1	0	0	0
$g_{x_k}(i)$	2^b-1	$2^{b+1}+1$	2^b+1	2^b
$\beta_k(i)$	1	2	1	0
$e_k(i)$	$4M$	$4M+2$	0	0
$\mathbf{P}_k(i)$	$8M^2+2M-1$	$10M^2+6M+2$	1	0
$\mathbf{h}_k(i)$	$4M$	$4M+2$	0	0
$\mathbf{w}_k(i)$	$2n_kM$	$4n_kM$	0	0
Total	$8M^2+2^b+2+$ $(2n_k+10)M$	$10M^2+2^{b+1}+$ $(4n_k+14)M+12$	2^b+3	2^b
Total (DRLS [15])	$8M^2-1+$ $(2n_k+10)M$	$10M^2+$ $(4n_k+14)M$	1	0

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 $\mathcal{T}_b$ and $\mathcal{L}_b$ to generate $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$. Figure 4.2 plots the network details.

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_{lk} are computed by the Metropolis rule [41], $\gamma = 0.9$ and $\lambda = 0.99$. We have compared the proposed DQA-RLS [62] in Algorithm 4 with DRLS [15], DLMS [14], and DQA-LMS [59]. 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 $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ as the coarsely quantized data.

Figures 4.3 and 4.4 show the global MSD learning curve (average MSD

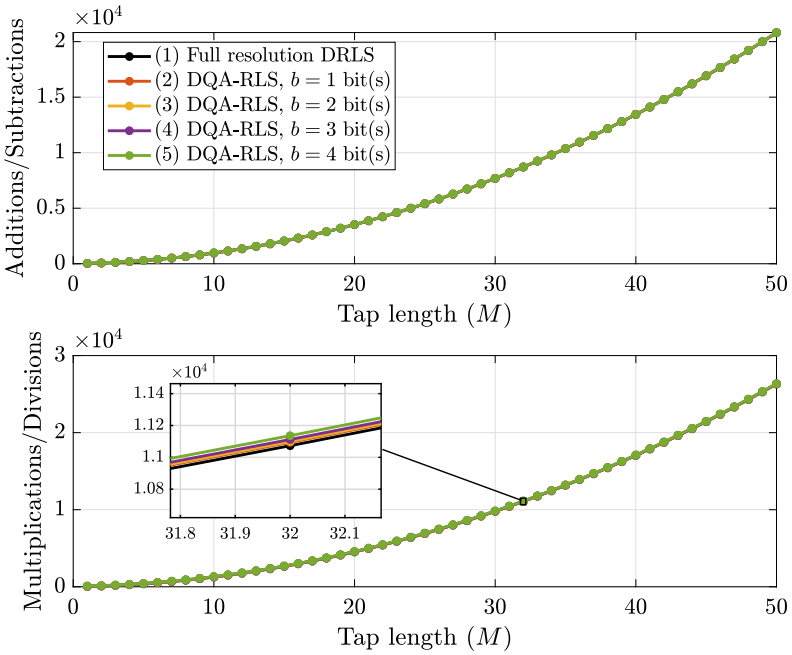
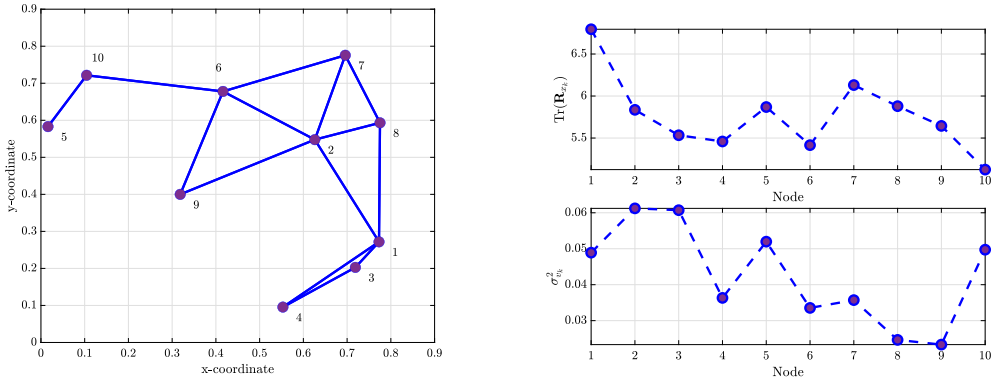


Figure 4.1: Number of operations per node versus the filter length for $n_k = 3$.

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 Figure 2.3).

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 Figures 4.5 and 4.6. In particular, Figure 4.6 compares the simulation results with those obtained by the analytical expression in (4-66). The results in Figure 4.6 indicate that the theoretical and simulated results agree well especially for $b = 3$ and $b = 4$ bits, and low to



(a) Distributed network structure (b) Network statistical settings

Figure 4.2: A wireless network with $N = 10$ nodes.

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 Figure 4.7 and authenticate the validation of the MSD theoretical expression (4-65).

In Figures 4.8 and 4.9, the MSD learning curves of the proposed DQA-RLS are compared with those of DQA-LMS [59], standard DRLS [15] and DLMS algorithms [14]. We choose the same step sizes for all agents, i.e., $\mu_k = 0.05$ for DLMS and DQA-LMS 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 (3) and (4) in Figure 4.9, 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.

4.8 Chapter Summary

In this chapter, 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

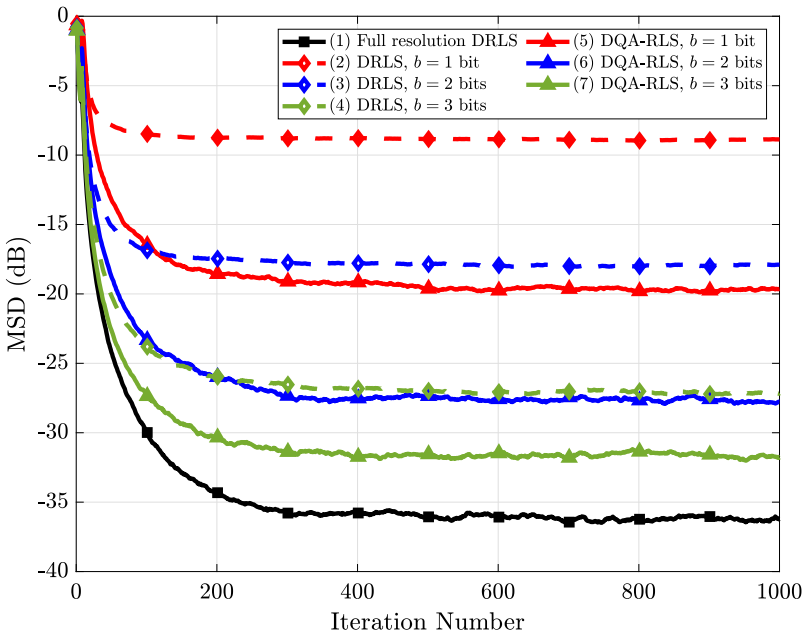


Figure 4.3: MSD curves for the DRLS and DQA-RLS algorithms.

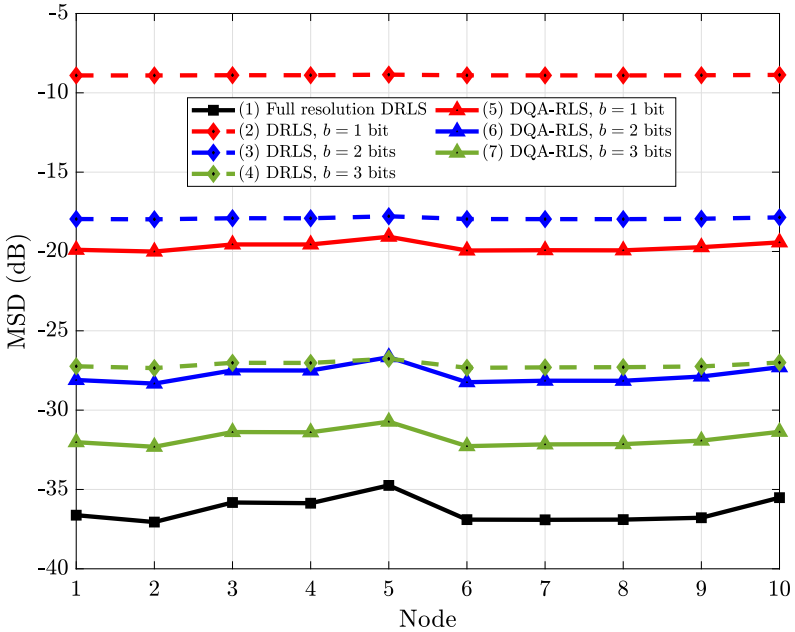


Figure 4.4: Steady-state MSD values for the DRLS and DQA-RLS algorithms.

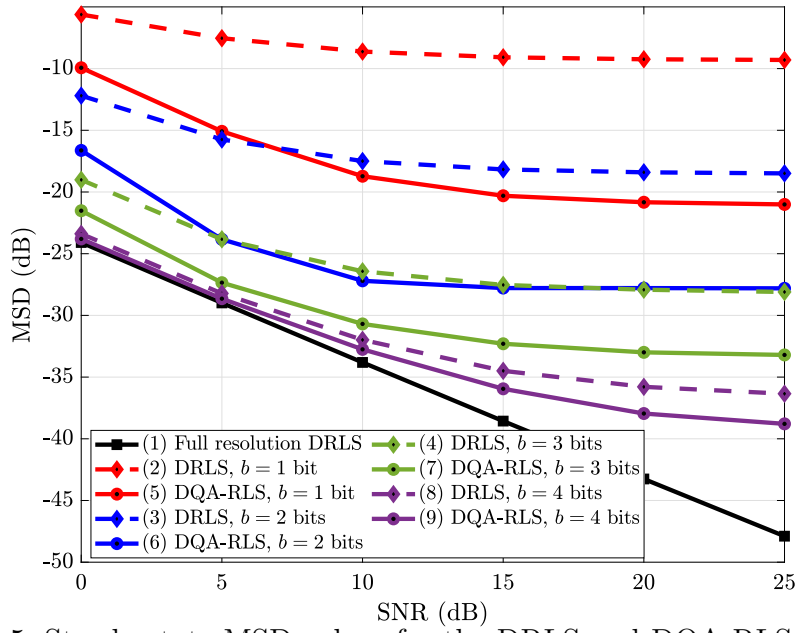


Figure 4.5: Steady-state MSD values for the DRLS and DQA-RLS algorithms for different SNR values.

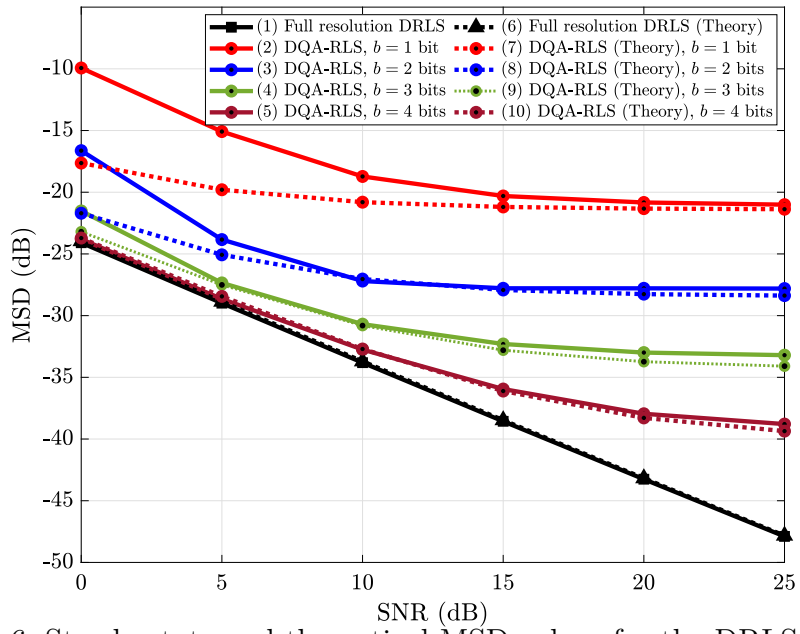


Figure 4.6: Steady-state and theoretical MSD values for the DRLS and DQA-RLS algorithms for different SNR values.

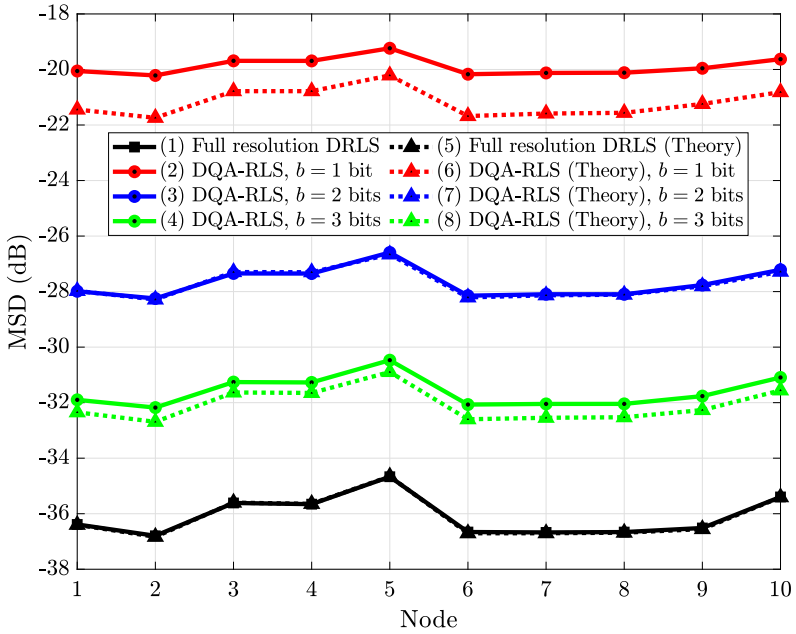


Figure 4.7: Node-wise Steady-state and theoretical MSD values for the DRLS and DQA-RLS algorithms.

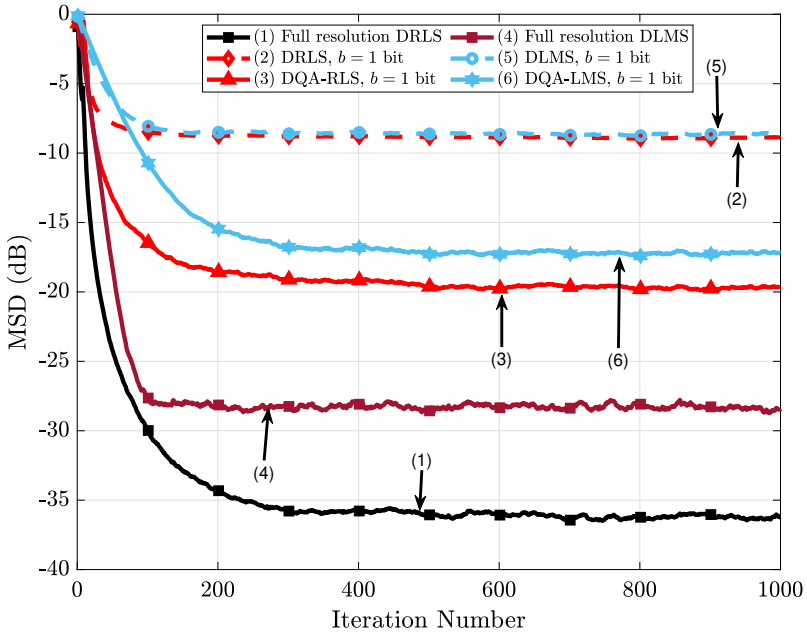


Figure 4.8: MSD curves for the DRLS, DLMS, DQA-RLS and DQA-LMS algorithms.

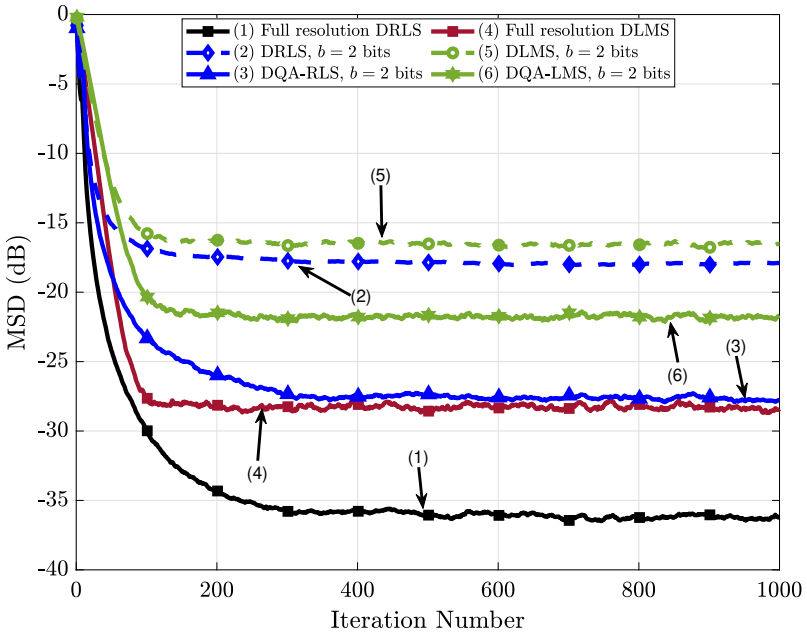


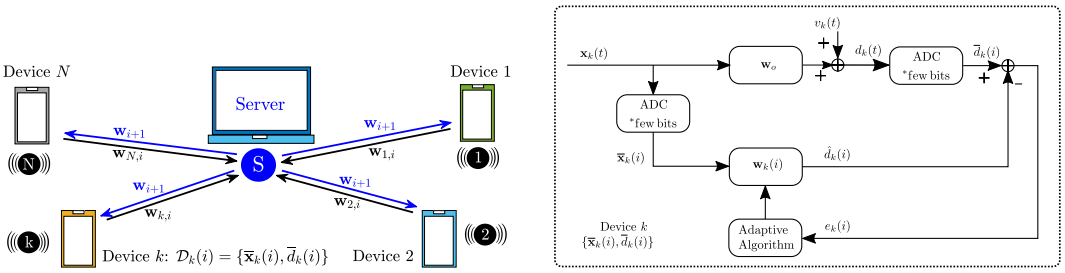
Figure 4.9: MSD curves for the DRLS, DLMS, DQA-RLS and DQA-LMS algorithms.

comparable computational complexity to the standard DRLS algorithm while it reduces the power consumption of the sensors in the network by using low-resolution ADCs. We have also carried out a statistical analysis of the DQA-RLS algorithm deducing a stability condition and an expression for the mean square deviation (MSD). 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.

5
Quantization-Aware Federated Averaging LMS Algorithm

In this chapter, we present the derivation of the proposed QA-FedAvg-LMS algorithm and present a statistical analysis of the QA-FedAvg-LMS algorithm. In addition, we devise a bias compensation strategy and investigate the computational complexity of the proposed and existing algorithms. We assess the estimation performance of the QA-FedAvg-LMS algorithm for a parameter estimation setup with numerical results.

5.1
Derivation of QA-FedAvg-LMS



(a) Federated learning setup (b) IoT device k

Figure 5.1: A federated IoT network

We consider an IoT network that is working under the federated learning setup, as illustrated in Figure 5.1a, where N IoT devices are working under the supervision of a server to complete a task. Let us consider $\mathbf{x}_k(t)$ and $d_k(t)$ as the analog input and output of the unknown system $\mathbf{w}_o$ at IoT device k. Let $\mathbf{x}_k(i)$ and $d_k(i)$ denote the high-precision sampled versions of $\mathbf{x}_k(t)$ and $d_k(t)$, and $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ denote the coarsely quantized versions of $\mathbf{x}_k(i)$ and $d_k(i)$, respectively, as shown in Figure 5.1b.

Let $\beta_k(i)$ be a bias compensation coefficient to be chosen, define $\widehat{d}_k(i) = \beta_k(i)\mathbf{w}^*(i-1)\overline{\mathbf{x}}_k(i)$ and construct an MSE cost function as described by

$$\begin{aligned} j_k(\mathbf{w}(i-1)) &= \mathbb{E}[|e_k(i)|^2] = \mathbb{E}[\|\overline{d}_k(i) - \widehat{d}_k(i)\|^2] \\ &= \mathbb{E}[\|\overline{d}_k(i) - \beta_k(i)\mathbf{w}^*(i-1)\overline{\mathbf{x}}_k(i)\|^2], \end{aligned} \tag{5-1}$$

which is defined based on the quantized data samples $\overline{d}_k(i)$ and $\overline{\mathbf{x}}_k(i)$, and a bias correction term $\beta_k(i)$. The gradient of (5-1) with respect to $\mathbf{w}^*(i-1)$ is given by:

$$\nabla j_k(\mathbf{w}(i-1)) = -\beta_k(i)\overline{\mathbf{x}}_k(i)(\overline{d}_k(i) - \beta_k(i)\mathbf{w}^*(i-1)\overline{\mathbf{x}}_k(i))^*. \tag{5-2}$$

Replacing (5-2) into (2-20) and using (2-21), we obtain the QA-FedAvg-LMS algorithm as follows:

$$\mathbf{w}_k(i) = \mathbf{w}(i-1) + \mu_k\beta_k(i)\overline{\mathbf{x}}_k(i)e_k^*(i) \tag{5-3a}$$

$$\mathbf{w}(i) = \frac{1}{N}\sum_{k=1}^{N}\mathbf{w}_k(i), \tag{5-3b}$$

where $e_k(i) = \overline{d}_k(i) - \beta_k(i)\mathbf{w}^*(i-1)\overline{\mathbf{x}}_k(i)$ is the estimation error. We call (5-3a) and (5-3b) the adaptation and averaging steps which are performed on the devices and the server, respectively. The next section shows how the bias compensation $\beta_k(i)$ should be chosen such that (5-3) is asymptotically unbiased in the mean.

5.2 Mean Performance Analysis

To start the analysis, we use the following assumption that is very common in parameter estimation [55, 56, 57] and adaptive signal processing [58].

Assumption 1: The input data 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 $\sigma_{v_k}^2$ and the quantized regressors $\overline{\mathbf{x}}_k(i)$ with covariance matrices $\mathbf{R}_{\overline{x}_k} = \mathbb{E}[\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)]$. Moreover, covariance matrices are time-invariant and all data are assumed spatially independent.

To analyze the performance of QA-FedAvg-LMS, we use the weight-error vectors defined as:

$$\widetilde{\mathbf{w}}_k(i) = \mathbf{w}_o - \mathbf{w}_k(i), \qquad \text{and} \qquad \widetilde{\mathbf{w}}_i = \mathbf{w}_o - \mathbf{w}(i). \tag{5-4}$$

Under Assumption 1, if the entries in the input regressors $\mathbf{x}_k(i)$ are uncorrelated and with equal variance, we have $\mathbf{R}_{x_k} = \mathbb{E}[\mathbf{x}_k(i)\mathbf{x}_k^*(i)] \approx \sigma_{x_k}^2 \mathbf{I}_M$ and the matrix $\mathbf{G}_{x_k}$ reduces to $g_{x_k}\mathbf{I}_M$. Let us denote $\mathbf{R}_{q_k} = E[\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)] \approx \widehat{\sigma}_{q_k}^2 \mathbf{I}_M$ where $\widehat{\sigma}_{q_k}^2$ is given by (2-35).

Using (2-32), we can decompose $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$ as follows:

$$\overline{\mathbf{x}}_k(i) = g_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i), \tag{5-5}$$

$$\overline{d}_k(i) = g_{d_k}d_k(i) + q_{d_k}(i) = g_{d_k}\mathbf{w}_o^*\mathbf{x}_k(i) + p_k(i), \tag{5-6}$$

where $p_k(i) = g_{d_k}v_k(i) + q_{d_k}(i)$. Using this decomposition, we write the error $e_k(i)$ as follows:

$$\begin{aligned} e_k(i) &= \overline{d}_k(i) - \beta_k(i)\mathbf{w}^*(i-1)\overline{\mathbf{x}}_k(i) \\ &= g_{d_k}\mathbf{w}_o^*\mathbf{x}_k(i) + p_k(i) - \beta_k(i)\mathbf{w}^*(i-1)\Big(g_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i)\Big). \end{aligned} \tag{5-7}$$

Replacing (5-7) into (5-3a) and subtracting from $\mathbf{w}_o$ yields

$$
\begin{aligned}
\widetilde{\mathbf{w}}_k(i) &= \widetilde{\mathbf{w}}_{i-1} - \mu_k\beta_k(i)\overline{\mathbf{x}}_k(i)e_k^*(i) = \widetilde{\mathbf{w}}_{i-1} - \mu_k\beta_k(i)(g_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i))e_k^*(i) \\
&= \widetilde{\mathbf{w}}_{i-1} - \mu_k\beta_k(i)\Big(g_{x_k}g_{d_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{w}_o + g_{x_k}\mathbf{x}_k(i)p_k(i) \\
&\quad - g_{x_k}^2\beta_k(i)\mathbf{x}_k(i)\mathbf{x}_k^*(i)\mathbf{w}(i-1) - g_{x_k}\beta_k(i)\mathbf{x}_k(i)\mathbf{q}_{x_k}^*(i)\mathbf{w}(i-1) \\
&\quad + g_{d_k}\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\mathbf{w}_o + \mathbf{q}_{x_k}(i)p_k(i) \\
&\quad - g_{x_k}\beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\mathbf{w}(i-1) - \beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\mathbf{w}(i-1)\Big) \\
&= \widetilde{\mathbf{w}}_{i-1} - \mu_k\beta_k(i)\Big(\Big(g_{x_k}g_{d_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i) + g_{d_k}\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\Big)\mathbf{w}_o \\
&\quad - \Big(g_{x_k}^2\beta_k(i)\mathbf{x}_k(i)\mathbf{x}_k^*(i) + g_{x_k}\beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i) \\
&\qquad + g_{x_k}\beta_k(i)\mathbf{x}_k(i)\mathbf{q}_{x_k}^*(i) + \beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big)\mathbf{w}(i-1) \\
&\quad + g_{x_k}\mathbf{x}_k(i)p_k(i) + \mathbf{q}_{x_k}(i)p_k(i)\Big).
\end{aligned}
\tag{5-8}
$$

We take the expectation from both sides of (5-8). Since $\mathbf{x}_k(i)$, $\mathbf{q}_{x_k}(i)$, and $p_k(i)$ are uncorrelated pairwise, the expectations of these cross terms vanish. Considering this, we obtain

$$
\begin{aligned}
\mathbb{E}[\widetilde{\mathbf{w}}_k(i)] &= \mathbb{E}[\widetilde{\mathbf{w}}_{i-1}] - \mu_k\beta_k(i)\Big(\mathbb{E}\Big[g_{x_k}g_{d_k}\mathbf{x}_k(i)\mathbf{x}_k^*(i)\Big]\mathbf{w}_o \\
&\quad - \mathbb{E}\Big[\Big(g_{x_k}^2\beta_k(i)\mathbf{x}_k(i)\mathbf{x}_k^*(i) + \beta_k(i)\mathbf{q}_{x_k}(i)\mathbf{q}_{x_k}^*(i)\Big)\mathbf{w}(i-1)\Big]\Big) \\
&= \mathbb{E}[\widetilde{\mathbf{w}}_{i-1}] - \mu_k\Big(g_{x_k}g_{d_k}\beta_k(i)\mathbf{R}_{x_k}\mathbf{w}_o \\
&\quad - \Big(g_{x_k}^2\beta_k^2(i)\mathbf{R}_{x_k} + \beta_k^2(i)\mathbf{R}_{q_k}\Big)\mathbb{E}\Big[\mathbf{w}(i-1)\Big]\Big).
\end{aligned}
\tag{5-9}
$$

In the last line of (5-9), we use a common assumption that states that $\mathbf{x}_k(i)$ varies slowly in relation to $\widetilde{\mathbf{w}}_{i-1}$ [58]. Thus, when they appear inside the expectations we decouple their expected values. This also applies to $\mathbf{q}_{x_k}(i)$ in relation to $\widetilde{\mathbf{w}}_{i-1}$.

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

$$
g_{x_k}g_{d_k}\beta_k(i)\mathbf{R}_{x_k} = g_{x_k}^2\beta_k^2(i)\mathbf{R}_{x_k} + \beta_k^2(i)\mathbf{R}_{q_k},
\tag{5-10}
$$

and we show in the next section that this condition is possible by appropriately

choosing $\beta_k(i)$. Assuming (5-10) and using (5-4), we can write (5-9) as follows:

$$\mathbb{E}[\widetilde{\mathbf{w}}_k(i)] = \Big(\mathbf{I}_M - \mu_k g_{x_k} g_{d_k} \beta_k(i) \mathbf{R}_{x_k}\Big)\mathbb{E}[\widetilde{\mathbf{w}}_{i-1}]. \tag{5-11}$$

Subtracting $\mathbf{w}_o$ from both sides of (5-3b) we observe that adding (5-11) results in the recursion

$$\mathbb{E}[\widetilde{\mathbf{w}}_i] = \Big(\mathbf{I}_M - \frac{1}{N}\sum_{l=1}^{N} \mu_l g_{x_l} g_{d_l} \beta_l(i) \mathbf{R}_{x_l}\Big)\mathbb{E}[\widetilde{\mathbf{w}}_{i-1}]. \tag{5-12}$$

We now define the following matrices:

$$\begin{aligned}
\boldsymbol{\mathcal{H}} &= \Big(\frac{1}{N}\mathbb{1}_N \otimes \mathbf{I}_M\Big)^T && (M \times MN)\\
\boldsymbol{\mathcal{M}} &= \text{blockdiag}\{\mu_1 \mathbf{I}_M, \ldots, \mu_N \mathbf{I}_M\} && (MN \times MN)\\
\boldsymbol{\mathcal{R}}_x &= \text{blockdiag}\{g_{x_1} g_{d_1} \beta_1(i) \mathbf{R}_{x_1}, \ldots, g_{x_N} g_{d_N} \beta_N(i) \mathbf{R}_{x_N}\} && (MN \times MN)\\
\widetilde{\mathbf{W}}_{i-1} &= \text{col}\Big\{\widetilde{\mathbf{w}}(i-1), \ldots, \widetilde{\mathbf{w}}(i-1)\Big\} && (MN \times 1),
\end{aligned}$$

and write the global form of (3-11) as follows:

$$\mathbb{E}[\widetilde{\mathbf{w}}_i] = \boldsymbol{\mathcal{H}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_x\Big)\mathbb{E}[\widetilde{\mathbf{W}}_{i-1}]. \tag{5-13}$$

The necessary and sufficient condition to ensure the mean stability of the network, i.e., $\mathbb{E}[\widetilde{\mathbf{w}}_i] \to 0$ as $i \to \infty$ is to have $\rho\Big(\boldsymbol{\mathcal{H}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_x\Big)\Big) < 1$. Since the spectral radius of $\boldsymbol{\mathcal{H}}$ is less than one, we must have $\rho\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_x\Big) < 1$. Therefore, the stability condition for QA-FedAvg-LMS is given by

$$0 < \mu_k < \frac{2}{\lambda_{\max}(\boldsymbol{\mathcal{R}}_x)}, \tag{5-14}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{\mathcal{R}}_x$.

5.3 Bias Compensation

From (5-10), we must have

$$\beta_k(i)\mathbf{I}_M = g_{x_k} g_{d_k} \mathbf{R}_{x_k}\Big(g_{x_k}^2 \mathbf{R}_{x_k} + \mathbf{R}_{q_k}\Big)^{-1}. \tag{5-15}$$

Therefore, the bias compensation term is expressed by

$$\beta_k(i) = \beta_k = \frac{g_{x_k} g_{d_k} \sigma_{x_k}^2}{g_{x_k}^2 \sigma_{x_k}^2 + \widehat{\sigma}_{q_k}^2}. \tag{5-16}$$

Remark 1: *(One ADC for each sensor).* To reduce the cost and energy consumption of sensors, we consider one ADC to quantize the measurement data $\{\mathbf{x}_k(i), d_k(i)\}$. Then g_{x_k} and g_{d_k} can be considered equal and this reduces the complexity of our algorithm as well.

Remark 2: *(Approximation of data variance).* Since the devices receive quantized data and have access to the covariance of the quantized data, $\mathbf{R}_{\overline{x}_k} = \mathbb{E}[\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)] \approx \sigma_{\overline{x}_k}^2 \mathbf{I}_M$, we approximate the variance of high precision data as follows:

$$\widehat{\sigma}_{x_k}^2 = \widehat{\sigma}_{\overline{x}_k}^2 + \widehat{\sigma}_{q_k}^2, \tag{5-17}$$

where $\widehat{\sigma}_{q_k}^2$ is given by (2-35). To compute $\widehat{\sigma}_{\overline{x}_k}^2$ which is given by (3-18), we use the iterative recursion (3-19) to reduce the computational complexity of our algorithm. Therefore, at each data sample i, the bias correction term is given by:

$$\beta_k = \frac{g_{x_k}^2 \widehat{\sigma}_{x_k}^2}{g_{x_k}^2 \widehat{\sigma}_{x_k}^2 + \widehat{\sigma}_{q_k}^2}. \tag{5-18}$$

The QA-FedAvg-LMS algorithm is summarized in Algorithm 5.

5.4 Mean Square Performance Analysis

In this section, we carry out a mean-square performance analysis and discuss the steady-state behavior of the QA-FedAvg-LMS algorithm. We first write (5-8) as follows:

Algorithm 5: QA-FedAvg-LMS algorithm

input : Initial value $\mathbf{w}(-1) \in \mathbb{R}^M$, step-size μ_k, $\widehat{\sigma}^2_{\overline{x}_k}(-1) = 0$ and $0 \ll \gamma < 1$
Generate $\mathcal{T}_b$ and $\mathcal{L}_b$, and Compute $\widehat{\sigma}^2_{q_k}$ from (2-35)

output : $\mathbf{w}(i)$

for $i = 0, 1, 2, \ldots$ **do**

  for device $k = 1, 2, \ldots, N$ in parallel **do**

    data: $\overline{d}_k(i)$ and $\overline{\mathbf{x}}_k(i) = [\overline{x}_k(i), \overline{x}_k(i-1), \ldots, \overline{x}_k(i-M+1)]^T$

    $\widehat{\sigma}^2_{\overline{x}_k}(i) = \gamma \widehat{\sigma}^2_{\overline{x}_k}(i-1) + (1-\gamma)|\overline{x}_k(i)|^2$

    $\widehat{\sigma}^2_{x_k} = \widehat{\sigma}^2_{\overline{x}_k}(i) + \widehat{\sigma}^2_{q_k}$

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

    $\beta_k(i) = \frac{g^2_{x_k} \widehat{\sigma}^2_{x_k}}{g^2_{x_k} \widehat{\sigma}^2_{x_k} + \widehat{\sigma}^2_{q_k}}$

    $e_k(i) = \overline{d}_k(i) - \beta_k(i) \mathbf{w}^*(i-1) \overline{\mathbf{x}}_k(i)$

    $\mathbf{w}_k(i) = \mathbf{w}(i-1) + \mu_k \beta_k(i) \mathbf{x}_k(i) e_k^*(i)$

  end for

  Server sends $\mathbf{w}(i) = \frac{1}{N} \sum_{k=1}^{N} \mathbf{w}_k(i)$ to all devices

end for

$$
\begin{aligned}
\widetilde{\mathbf{w}}_k(i) &= \widetilde{\mathbf{w}}(i-1) - \mu_k \beta_k \overline{\mathbf{x}}_k(i) \Big(\overline{d}_k(i) - \beta_k \mathbf{w}^*(i-1) \overline{\mathbf{x}}_k(i) \Big)^* \\
&= \widetilde{\mathbf{w}}(i-1) - \mu_k \beta_k \overline{\mathbf{x}}_k(i) \overline{d}_k^*(i) + \mu_k \beta_k^2 \overline{\mathbf{x}}_k(i) \overline{\mathbf{x}}_k^*(i) \mathbf{w}(i-1) \\
&= \Big(\mathbf{I}_M - \mu_k \beta_k^2 \overline{\mathbf{x}}_k(i) \overline{\mathbf{x}}_k^*(i) \Big) \widetilde{\mathbf{w}}(i-1) - \mu_k \beta_k \overline{\mathbf{x}}_k(i) \Big(g_{d_k} \mathbf{x}_k^*(i) \mathbf{w}_o + p_k^*(i) \Big) \\
&\quad + \mu_k \beta_k^2 \overline{\mathbf{x}}_k(i) \overline{\mathbf{x}}_k^*(i) \mathbf{w}_o \\
&= \Big(\mathbf{I}_M - \mu_k \beta_k^2 \overline{\mathbf{x}}_k(i) \overline{\mathbf{x}}_k^*(i) \Big) \widetilde{\mathbf{w}}(i-1) - \underbrace{\mu_k \beta_k g_{x_k} g_{d_k} \mathbf{x}_k(i) \mathbf{x}_k^*(i) \mathbf{w}_o}_{I} \\
&\quad - \mu_k \beta_k g_{d_k} \mathbf{q}_{x_k}(i) \mathbf{x}_k^*(i) \mathbf{w}_o - \mu_k \beta_k g_{x_k} \mathbf{x}_k(i) p_k^*(i) - \mu_k \beta_k \mathbf{q}_{x_k}(i) p_k^*(i) \\
&\quad + \underbrace{\mu_k \beta_k^2 g_{x_k}^2 \mathbf{x}_k(i) \mathbf{x}_k^*(i) \mathbf{w}_o + \mu_k \beta_k^2 \mathbf{q}_{x_k}(i) \mathbf{q}_{x_k}^*(i) \mathbf{w}_o}_{II} .
\end{aligned}
\tag{5-19}
$$

The choice of β_k in (5-18) makes (5-10) approximately true. In order to reduce the complexity of the model, we assume in the sequel that (5-10) is exactly true, so the instantaneous values of terms I and II in (5-19) are equal with different signs and vanish for sufficiently large i (our simulations in Sec. 5.6 show that this approximation is reasonable). Then (5-19) simplifies to

$$\begin{aligned}\widetilde{\mathbf{w}}_k(i) = \Big(\mathbf{I}_M - \mu_k\beta_k^2\overline{\mathbf{x}}_k(i)\overline{\mathbf{x}}_k^*(i)\Big)\widetilde{\mathbf{w}}(i-1) - \mu_k\beta_k g_{d_k}\mathbf{q}_{x_k}(i)\mathbf{x}_k^*(i)\mathbf{w}_o \\ - \mu_k\beta_k g_{x_k}\mathbf{x}_k(i)p_k^*(i) - \mu_k\beta_k\mathbf{q}_{x_k}(i)p_k^*(i).\end{aligned} \tag{5-20}$$

Therefore, the weight-error vectors of the averaged estimates at the server are given by

$$\begin{aligned}\widetilde{\mathbf{w}}(i) = \frac{1}{N}\sum_{k=1}^{N}\widetilde{\mathbf{w}}_l(i) = \Big(\mathbf{I}_M - \frac{1}{N}\sum_{k=1}^{N}\mu_l\beta_l^2\overline{\mathbf{x}}_l(i)\overline{\mathbf{x}}_l^*(i)\Big)\widetilde{\mathbf{w}}_l(i-1) \\ - \frac{1}{N}\sum_{k=1}^{N}\mu_l\beta_l g_{d_l}\mathbf{q}_{x_l}(i)\mathbf{x}_l^*(i)\mathbf{w}_o - \frac{1}{N}\sum_{k=1}^{N}\mu_l\beta_l g_{x_l}\mathbf{x}_l(i)p_l^*(i) \\ - \frac{1}{N}\sum_{k=1}^{N}\mu_l\beta_l\mathbf{q}_{x_l}(i)p_l^*(i)\end{aligned} \tag{5-21}$$

Let us now define

$$\begin{aligned}
\boldsymbol{\mathcal{R}}_{\overline{x}} &= \text{blockdiag}\Big\{\mathbf{R}_{\overline{x}_1},\ldots,\mathbf{R}_{\overline{x}_N}\Big\} && (MN\times MN)\\
\boldsymbol{\mathcal{B}} &= \text{blockdiag}\Big\{\beta_1\mathbf{I}_M,\ldots,\beta_N\mathbf{I}_M\Big\} && (MN\times MN)\\
\mathfrak{B} &= \text{blockdiag}\Big\{\beta_1^2\mathbf{I}_M,\ldots,\beta_N^2\mathbf{I}_M\Big\} && (MN\times MN)\\
\boldsymbol{\mathcal{G}} &= \text{blockdiag}\Big\{g_{x_1}\mathbf{I}_M,\ldots,g_{x_N}\mathbf{I}_M\Big\} && (MN\times MN)\\
\boldsymbol{\Upsilon}_i &= \text{blockdiag}\Big\{\mathbf{q}_{x_1}(i)\mathbf{x}_1^*(i),\ldots,\mathbf{q}_{x_N}(i)\mathbf{x}_N^*(i)\Big\} && (MN\times MN)\\
\boldsymbol{\xi}_i &= \text{col}\Big\{\mathbf{x}_1(i)p_1^*(i),\ldots,\mathbf{x}_N(i)p_N^*(i)\Big\} && (MN\times 1)\\
\boldsymbol{\zeta}_i &= \text{col}\Big\{\mathbf{q}_{x_1}(i)p_1^*(i),\ldots,\mathbf{q}_{x_N}(i)p_N^*(i)\Big\} && (MN\times 1)\\
\boldsymbol{\eta} &= \text{col}\Big\{g_{d_1}\mathbf{w}_o,\ldots,g_{d_N}\mathbf{w}_o\Big\} && (MN\times 1),
\end{aligned}$$

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

$$\widetilde{\mathbf{w}}_i = \boldsymbol{\mathcal{H}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\mathfrak{B}\boldsymbol{\mathcal{R}}_{\overline{x}}\Big)\widetilde{\mathbf{W}}_{i-1} - \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\boldsymbol{\eta} - \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i - \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i.$$

Defining an $M \times MN$ matrix $\boldsymbol{\mathcal{D}}$ as follows:

$$\begin{aligned}\boldsymbol{\mathcal{D}} &= \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\Big(\mathbf{I}_{MN} - \text{blockdiag}\{\mu_1\beta_1^2\overline{\mathbf{x}}_1(i)\overline{\mathbf{x}}_1^*(i), \ldots, \mu_N\beta_N^2\overline{\mathbf{x}}_N(i)\overline{\mathbf{x}}_N^*(i)\}\Big)\Big] \\ &= \boldsymbol{\mathcal{H}}\Big(\mathbf{I}_{MN} - \boldsymbol{\mathcal{M}}\mathfrak{B}\boldsymbol{\mathcal{R}}_{\overline{x}}\Big)\end{aligned} \tag{5-22}$$

and 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] &= \mathbb{E}\Big[\boldsymbol{\mathcal{D}}\widetilde{\mathbf{W}}_{i-1}\widetilde{\mathbf{W}}_{i-1}^*\boldsymbol{\mathcal{D}}^*\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\eta^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\eta^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\eta\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\eta\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] \\ &\quad + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big] + \mathbb{E}\Big[\boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\boldsymbol{\Upsilon}_i\eta\eta^*\boldsymbol{\Upsilon}_i^*\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\boldsymbol{\mathcal{H}}^T\Big].\end{aligned} \tag{5-23}$$

We now use the commutative property of the expectation and vectorization operations, and the relationship between the vectorization operation and the Kronecker product, $\text{vec}(ABC) = (C^T \otimes A)\,\text{vec}(B)$, to write (5-23) as follows:

$$\text{vec}(\boldsymbol{\Omega}_i) = (\boldsymbol{\mathcal{D}} \otimes \boldsymbol{\mathcal{D}})\,\text{vec}(\boldsymbol{\Omega}_{i-1}) + \text{vec}(\boldsymbol{\Xi}_i), \tag{5-24}$$

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 (5-23). Note that $\text{vec}(\boldsymbol{\Omega}_i)$ is stable if and only if the spectral radius of $(\boldsymbol{\mathcal{D}} \otimes \boldsymbol{\mathcal{D}})$ is strictly smaller than 1. Therefore, we obtain

$$\lim_{i \to +\infty} \text{vec}(\boldsymbol{\Omega}_i) = \Big(\mathbf{I}_{M^2N^2} - (\boldsymbol{\mathcal{D}} \otimes \boldsymbol{\mathcal{D}})\Big)^{-1}\,\text{vec}(\boldsymbol{\Xi}_{+\infty}), \tag{5-25}$$

and $\text{vec}(\boldsymbol{\Xi}_{+\infty})$ is given by

$$\begin{aligned}\mathrm{vec}(\boldsymbol{\Xi}_{+\infty}) &= \lim_{i\to+\infty}\mathrm{vec}(\boldsymbol{\Xi}_i) = (\boldsymbol{\mathcal{H}}\otimes\boldsymbol{\mathcal{H}})\\ &\quad \mathrm{vec}\Big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\\ &\quad +\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\zeta}_i^*\Big]\\ &\quad +\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\zeta}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]\Big)\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\Big)\\ &\approx (\boldsymbol{\mathcal{H}}\otimes\boldsymbol{\mathcal{H}})\\ &\quad \mathrm{vec}\Big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\Big)\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\Big),\end{aligned} \tag{5-26}$$

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

$$\begin{aligned}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big] &= \mathrm{diag}\Big\{\mathbf{R}_{x_1}\sigma_{p_1}^2,\ldots,\mathbf{R}_{x_N}\sigma_{p_N}^2\Big\}\\ \mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big] &= \mathrm{diag}\Big\{g_{d_1}^2\mathbf{R}_{q_x,1}\Big(\mathbf{w}_o^*\mathbf{R}_{x_1}\mathbf{w}_o\Big),\ldots,g_{d_N}^2\mathbf{R}_{q_x,N}\Big(\mathbf{w}_o^*\mathbf{R}_{x_N}\mathbf{w}_o\Big)\Big\}\\ \mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big] &= \mathrm{diag}\Big\{\mathbf{R}_{q_x,1}\sigma_{p_1}^2,\ldots,\mathbf{R}_{q_x,N}\sigma_{p_N}^2\Big\}.\end{aligned} \tag{5-27}$$

The analytical computation of $\mathbf{R}_{\overline{x}_k}$, $\mathbf{R}_{q_x,k}$, and $\sigma_{p_k}^2$ is detailed in Appendix A. Therefore, the steady-state MSD of the network is given by

$$\begin{aligned}\mathrm{MSD} &= \lim_{i\to+\infty}\mathbb{E}[\|\widetilde{\mathbf{w}}(i)\|^2] = \mathrm{vec}(\boldsymbol{\Omega}_{+\infty})\\ &= \mathrm{vec}^T\Big(\mathbf{I}_{M^2N^2}-(\boldsymbol{\mathcal{D}}\otimes\boldsymbol{\mathcal{D}})\Big)^{-1}(\boldsymbol{\mathcal{H}}\otimes\boldsymbol{\mathcal{H}})\\ &\quad \mathrm{vec}\Big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{B}}\Big(\boldsymbol{\mathcal{G}}\mathbb{E}\Big[\boldsymbol{\xi}_i\boldsymbol{\xi}_i^*\Big]\boldsymbol{\mathcal{G}}^T+\mathbb{E}\Big[\boldsymbol{\Upsilon}_i\boldsymbol{\eta}\boldsymbol{\eta}^*\boldsymbol{\Upsilon}_i^*\Big]+\mathbb{E}\Big[\boldsymbol{\zeta}_i\boldsymbol{\zeta}_i^*\Big]\Big)\boldsymbol{\mathcal{B}}^T\boldsymbol{\mathcal{M}}^T\Big).\end{aligned} \tag{5-28}$$

It can be seen from (5-24) that, for $0 \ll \mu_k < 1$ the eigenvalues of $\boldsymbol{\mathcal{D}}\otimes\boldsymbol{\mathcal{D}}$ remain in the interval $(-1,1)$, and thus QA-FedAvg-LMS is stable in the mean-square sense and $\Big(\mathbf{I}_{M^2N^2}-(\boldsymbol{\mathcal{D}}\otimes\boldsymbol{\mathcal{D}})\Big)$ in (5-28) is nonsingular.

Remark 3: *(High precision signals, $b=\infty$).* Increasing the number of quantization bits, the diagonal entries of $\mathbf{G}_{x_k}$ approach unity where for high precision signals ($b=\infty$) with $\overline{\mathbf{x}}_k = \mathbf{x}_k$, we have $\mathbf{G}_{x_k} = \mathbf{I}_M$ according to (2-33)

and $\mathbf{R}_{q_x,k} = \mathbf{0}$ according to (5-5), and consequently $\boldsymbol{\beta}_k(i) = \mathbf{I}_M$ from (5-18). For $b = \infty$ we also have $q_{d_k} = 0$ and $g_{d_k} = 1$, and since $p_k(i) = g_{d_k} v_k(i) + q_{d_k}(i)$, thus $\sigma^2_{p_k} = \sigma^2_{v_k}$. Therefore, for high precision signals, the third and fourth definitions in (5-27) vanish, and with $\boldsymbol{\mathcal{R}} = \text{blockdiag}\{\mathbf{R}_{x_1}, \ldots, \mathbf{R}_{x_N}\}$, (5-28) reduces to

$$\begin{aligned} \text{MSD} = \text{vec}^T \Big(\mathbf{I}_{M^2N^2} - \Big((\mathbf{I}_{MN} - \boldsymbol{\mathcal{R}}) \otimes (\mathbf{I}_{MN} - \boldsymbol{\mathcal{R}})(\boldsymbol{\mathcal{H}} \otimes \boldsymbol{\mathcal{H}}) \Big) \Big)^{-1} \\ (\boldsymbol{\mathcal{H}} \otimes \boldsymbol{\mathcal{H}})\, \text{vec} \Big(\text{diag} \Big\{ \mu_k^2 \sigma^2_{v_1} \mathbf{R}_{x_1}, \ldots, \mu_k^2 \sigma^2_{v_N} \mathbf{R}_{x_N} \Big\} \Big), \end{aligned} \tag{5-29}$$

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

5.5 Computational Complexity

Table 5.1 shows the computational complexity of the QA-FedAvg-LMS algorithm in terms of the number of multiplications and additions at device k per time instant. At each time instant, QA-FedAvg-LMS performs a few more operations ($\approx O(2^b)$) than FedAvg-LMS. However, the extra complexity of QA-FedAvg-LMS allows the system to work in a more energy-efficient way. The computational cost at the server (5-3b) is the same for QA-FedAvg-LMS and FedAvg-LMS algorithms. Table 5.2 compares the total computational complexity of the quantization-aware algorithms [59, 60, 62, 63, 64] in this thesis with the total computational complexity of the standard distributed adaptive algorithms [14, 15] at node k.

5.6 Simulation Results

In this section, we assess the performance of the QA-FedAvg-LMS algorithm for a parameter estimation problem in an IoT network with $N = 100$ devices. The unknown parameter vector has a length of $M = 32$, is generated randomly and normalized to unit norm. We generated 10^5 $M \times 1$ vectors with multivariate Gaussian distribution as the input data samples $\mathbf{x}_k(i)$ for

Table 5.1: Computational complexity of QA-FedAvg-LMS algorithm per time instant at device k

Task	$+-$	$\times$	$\div$	exp
$\widehat{\sigma}^2_{x_k}$	3	3	0	0
$g_k(i)$	2^b-1	$2^{b+1}+1$	2^b+1	2^b
$\beta_k(i)$	1	2	1	0
$\widehat{d}_{k,Q}(i)$	$M-1$	$M+1$	0	0
$e_k(i)$	1	0	0	0
$\mathbf{w}_k(i+1)$	M	$M+2$	0	0
Total	$2M+2^b+3$	$2M+2^{b+1}+9$	2^b+2	2^b
Total (FedAvg-LMS)	$2M$	$2M+1$	0	0

Table 5.2: Total computational complexity of the DLMS, DQA-LMS, DRLS, DQA-RLS, FedAvg-LMS, and QA-FedAvg-LMS algorithms per time instant at device k

Task	$+-$	$\times$	$\div$	exp
DLMS	$(2+n_k)M$	$(2+n_k)M+1$	0	0
DQA-LMS	$(2+n_k)M+2^b+3$	$(2+n_k)M+2^{b+1}+9$	2^b+2	2^b
DRLS	$8M^2-1+$ $(2n_k+10)M$	$10M^2+$ $(4n_k+14)M$	1	0
DQA-RLS	$8M^2+2^b+2+$ $(2n_k+10)M$	$10M^2+2^{b+1}+$ $(4n_k+14)M+12$	2^b+3	2^b
FedAvg-LMS	$2M$	$2M+1$	0	0
QA-FedAvg-LMS	$2M+2^b+3$	$2M+2^{b+1}+9$	2^b+2	2^b

100 devices (1000 data samples for each device) with the covariance matrix $\mathbf{R}_{x_k}=\sigma^2_{x_k}\mathbf{I}_M$ where $\sigma^2_{x_k}\in(0.5,1)$. The noise samples of each device are drawn from a zero mean white Gaussian process with variance $\sigma^2_{v_k}\in(0.01,0.05)$. The data samples are quantized with $\mathcal{T}_b$ and $\mathcal{L}_b$ to generate $\overline{\mathbf{x}}_k(i)$ and $\overline{d}_k(i)$. We choose $\mu_k=0.05$ as the step size of QA-FedAvg-LMS and FedAvg-LMS.

We use the mean-square deviation (MSD) to investigate the performance of the network and use the excess mean square error (EMSE) to compare the performance of each device k as given by:

$$\begin{aligned} \text{MSD} &\triangleq \lim_{i\to+\infty} \mathbb{E}[\|\mathbf{w}_o - \mathbf{w}(i)\|^2], \\ \text{EMSE}_k &\triangleq \lim_{i\to+\infty} \mathbb{E}[\|(\mathbf{w}_o - \mathbf{w}_k(i))^* \mathbf{x}_k(i)\|^2]. \end{aligned} \tag{5-30}$$

The simulated learning curves are obtained by ensemble averaging over 200 independent trials and the steady-state values are averaged over the last 10% data samples. We have compared QA-FedAvg-LMS (5-3) with FedAvg-LMS (2-24) with generated data quantized with different numbers of bits. Full resolution FedAvg-LMS refers to the case where the data $\{\mathbf{x}_k(i), d_k(i)\}$ is not quantized.

Figure 5.2 shows the evaluation of the global MSD (5-30) for 1000 communication rounds between server and devices. Figure 5.3 compares the steady-state MSD values for the different signal-to-noise ratios (SNR) (keeping $\sigma_{x_k}^2 \in (0.5, 1)$ and changing $\sigma_{v_k}^2$) where the SNR value is averaged over devices. Figure 5.4 compares the steady-state EMSE (5-30) performance of 10 randomly chosen devices. As it can be seen in the numerical results, the network MSD and device-wise EMSE performance of the proposed QA-FedAvg-LMS algorithm are closer to the full resolution FedAvg-LMS while it substantially reduces the power consumption related to the ADCs in the input sensors.

In the last example, we evaluate the MSD performance of FedAvg-LMS and QA-FedAvg-LMS algorithms for different SNRs and we set $M = 8$ and $N = 10$ to calculate the theoretical expressions in (5-28). The step-size is set to 0.2 for 1-bit quantization and 0.05 for other curves. Figure 5.5 compares the simulation results with those obtained by the analytical expression in (5-28). The results 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.

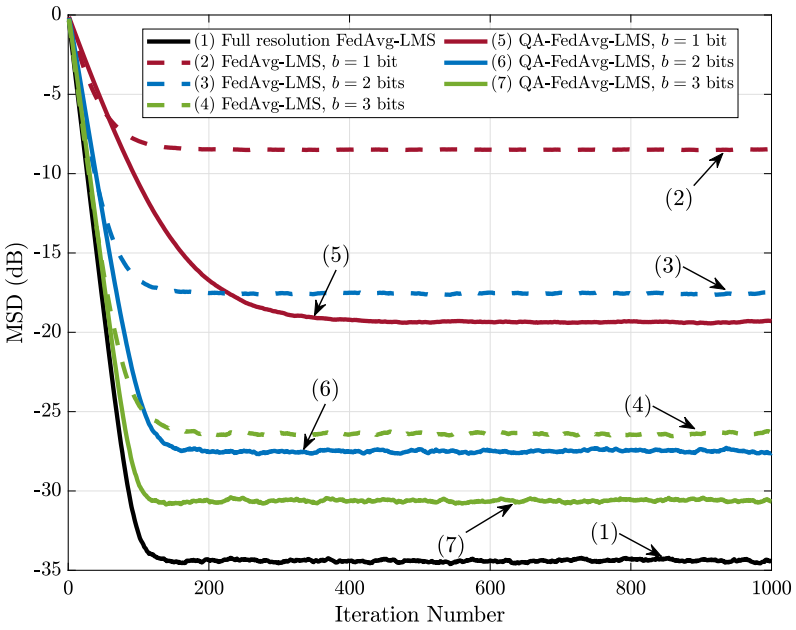


Figure 5.2: MSD curves for the FedAvg-LMS (2-24) and QA-FedAvg-LMS (5-3) algorithms.

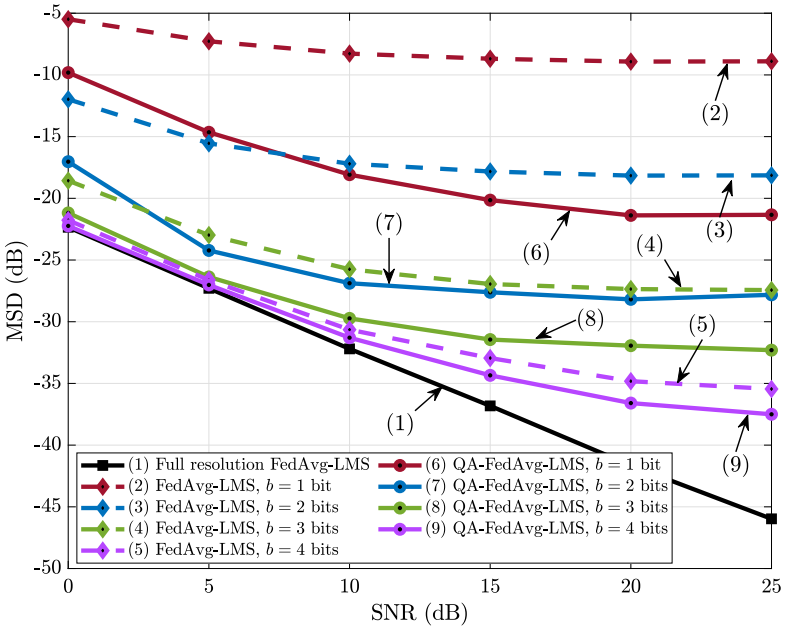


Figure 5.3: Steady-state MSD versus SNR for the FedAvg-LMS (2-24) and QA-FedAvg-LMS (5-3) algorithms.

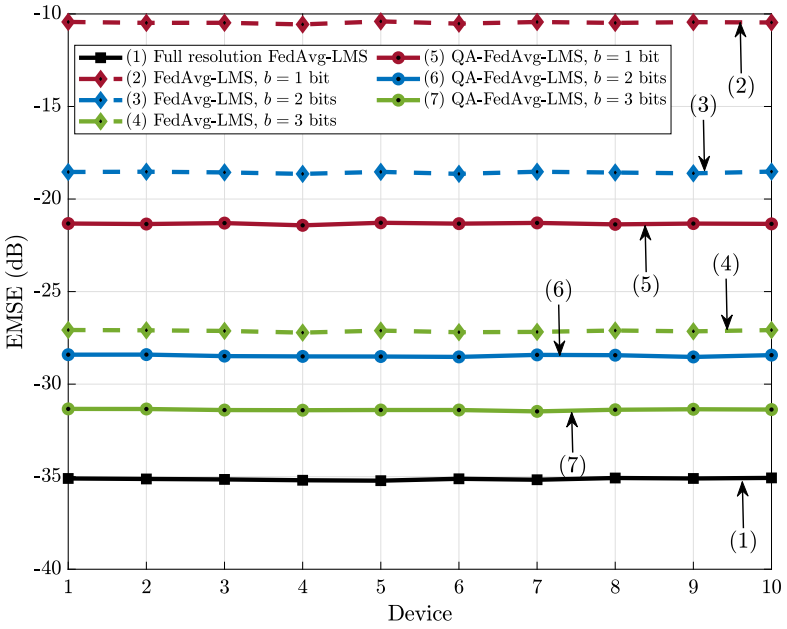


Figure 5.4: Steady-state EMSE curves for the FedAvg-LMS (2-24) and QA-FedAvg-LMS (5-3) algorithms.

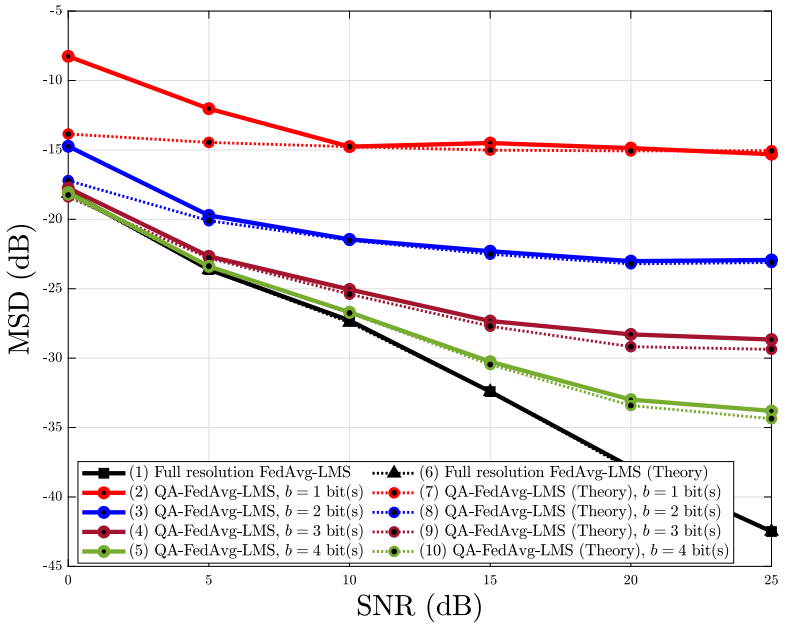


Figure 5.5: Steady-state and theoretical MSD values for the FedAvg-LMS and QA-FedAvg-LMS algorithms for different SNR values.

5.7 Chapter Summary

In this chapter, we have proposed an energy-efficient framework for federated learning and developed the QA-FedAvg-LMS algorithm along with bias compensation strategies for IoT networks. The QA-FedAvg-LMS algorithm has comparable computational complexity to the standard FedAvg-LMS algorithm while it substantially reduces the power consumption of the ADCs in the network. Simulations have shown excellent performance of QA-FedAvg-LMS as compared to FedAvg-LMS for coarsely quantized signals.

6
Conclusions and Future Works

This thesis has developed novel energy-efficient signal processing techniques for IoT networks to fill the existing gaps in distributed signal processing techniques working with coarsely quantized measurements. Two distributed adaptive algorithms, i.e., distributed least-mean square (LMS) and distributed recursive least-squares (RLS), have been developed and equipped with a quantization-aware framework to work with coarsely quantized signals which allows a substantial reduction of the energy consumption associated with ADCs using few quantization bits in two different scenarios: adaptive IoT networks where devices are operated in peer-to-peer mode and federated learning setup where devices are operated under the supervision of a central server.

Chapter 3 has introduced the distributed quantization-aware LMS (DQA-LMS) algorithm. The derivation of DQA-LMS was presented and consisted of the implementation of a signal decomposition based on Bussgang's theorem along with the development of a diffusion LMS algorithm with adaptive bias compensation to further improve the performance of DQA-LMS when the ADCs work with few bits. The complexity of DQA-LMS was investigated and it was shown that DQA-LMS has comparable computational cost to the standard DLMS algorithm while it enormously reduces the power consumption of the ADCs in the network. A statistical analysis of the proposed DQA-LMS algorithm was carried out including the mean and the mean square performance analysis, deducing a stability condition and an expression for the mean square deviation. Simulations have evaluated the DQA-LMS algorithm against existing techniques for a distributed parameter estimation task and demonstrated the

effectiveness of the DQA-LMS algorithm.

Chapter 4 has introduced the distributed quantization-aware RLS (DQA-RLS) algorithm. The derivation of DQA-RLS was presented with the bias compensation strategy to improve the performance of DQA-RLS when the ADCs are working with few bits. A statistical analysis of the proposed DQA-RLS algorithm was carried out including the mean and the mean square performance analysis, deducing a stability condition and an expression for the mean square deviation. The complexity of DQA-RLS was investigated and it was shown that DQA-RLS has comparable computational cost to the standard DRLS algorithm. Simulations have assessed the DQA-RLS algorithm against existing techniques for a distributed parameter estimation task and demonstrated the effectiveness of the DQA-RLS algorithm and a good match between theory and experiments.

Chapter 5 has presented the quantization-aware LMS algorithm for the federated learning setup, and derived the QA-FedAvg-LMS algorithm and acquired a bias compensation strategy to improve the performance of QA-FedAvg-LMS for coarsely quantized measurements. A statistical analysis of the proposed QA-FedAvg-LMS algorithm was carried out including the mean and the mean square performance analysis, deducing a stability condition and an expression for the mean square deviation. The computational complexity of QA-FedAvg-LMS was investigated and it was shown that QA-FedAvg-LMS has comparable computational cost to the standard FedAvg-LMS algorithm, while it enables IoT devices to work with low resolution ADCs to reduce their energy consumption. Simulations have evaluated the QA-FedAvg-LMS algorithm against existing techniques for a distributed parameter estimation task and demonstrated the effectiveness of the DQA-LMS algorithm.

Some suggestions for possible future works also include:

1. The DQA-LMS has the simplicity of LMS-type algorithms whereas the DQA-RLS is more robust against correlated inputs and faster in

convergence as other RLS-type algorithms. Therefore, 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 [65, 66], the partial-diffusion recursive least-squares (PDRLS) algorithm [28] have been successfully used for significant reduction in the complexity and communication cost of RLS algorithms, respectively, and are potential candidates to apply the quantization-aware framework from DQA-RLS to reduce the complexity, the communication cost and the energy consumption simultaneously.

2. Investigating the quantization-aware algorithms and the bias compensation strategies for the cases where i) the inputs have other probability distribution (not Gaussian) or ii) the distribution of input data is not known. Perhaps extending the Bussgang's theorem for different types of distribution could be helpful.

Bibliography

[1] PREDD, J. B.; KULKARNI, S. B. ; POOR, H. V., **Distributed learning in wireless sensor networks**, IEEE Signal Processing Magazine, vol. 23, no. 4, pp. 56–69, 2006.

[2] ROSSI, L. A.; KRISHNAMACHARI, B. ; KUO, C. C., **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., pp. 460–469. IEEE, 2004.

[3] RANA, M. M.; XIANG, W. ; WANG, E., **IoT-based state estimation for microgrids**, IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1345–1346, 2018.

[4] ZOU, Y.; XU, M.; SHENG, H.; XING, X.; XU, Y. ; ZHANG, Y., **Crowd density computation and diffusion via internet of things**, IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8111–8121, 2020.

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

[6] CHEN, J.; TOWFIC, Z. J. ; SAYED, A. H., **Dictionary learning over distributed models**, IEEE Transactions on Signal Processing, vol. 63, no. 4, pp. 1001–1016, 2014.

[7] CHOUVARDAS, S.; SLAVAKIS, K.; KOPSINIS, Y. ; THEODORIDIS, S., **A sparsity promoting adaptive algorithm for distributed learning**, IEEE Transactions on Signal Processing, vol. 60, no. 10, pp. 5412–5425, 2012.

[8] IBARS, C.; NAVARRO, M. ; GIUPPONI, L., **Distributed demand management in smart grid with a congestion game**, In: 2010 FIRST IEEE INTERNATIONAL CONFERENCE ON SMART GRID COMMUNICATIONS, pp. 495–500. IEEE, 2010.

[9] GIANNAKIS, G. B.; KEKATOS, V.; GATSIS, N.; KIM, S. J.; ZHU, H. ; WOLLENBERG, B. F., **Monitoring and optimization for power grids: A signal processing perspective**, IEEE Signal Processing Magazine, vol. 30, no. 5, pp. 107–128, 2013.

[10] MCMAHAN, B.; MOORE, E.; RAMAGE, D.; HAMPSON, S. ; ARCAS, B. A. Y., **Communication-efficient learning of deep networks from decentralized data**, In: ARTIFICIAL INTELLIGENCE AND STATISTICS, pp. 1273–1282. PMLR, 2017.

[11] KONEČNỲ, J.; MCMAHAN, H. B.; YU, F. X.; RICHTÁRIK, P.; SURESH, A. T. ; BACON, D., **Federated learning: Strategies for improving communication efficiency**, arXiv preprint arXiv:1610.05492, 2016.

[12] LI, T.; SAHU, A. K.; TALWALKAR, A. ; SMITH, V., **Federated learning: Challenges, methods, and future directions**, IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50–60, 2020.

[13] OLFATI-SABER, R.; FAX, J. A. ; MURRAY, R. M., **Consensus and cooperation in networked multi-agent systems**, Proceedings of the IEEE, vol. 95, no. 1, pp. 215–233, 2007.

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

[15] CATTIVELLI, F. S.; LOPES, C. G. ; SAYED, A. H., **Diffusion recursive least-squares for distributed estimation over adaptive networks**, IEEE Transactions on Signal Processing, vol. 56, no. 5, pp. 1865–1877, 2008.

[16] XU, S.; DE LAMARE, R. C. ; POOR, H. V., **Distributed estimation over sensor networks based on distributed conjugate gradient strategies**, IET Signal Processing, vol. 10, no. 3, pp. 291–301, 2016.

[17] MILLER, T. G.; XU, S.; DE LAMARE, R. C.; NASCIMENTO, V. H. ; ZAKHAROV, Y., **Sparsity-aware distributed conjugate gradient algorithms for parameter estimation over sensor networks**, In: 2015 49TH ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, pp. 1556–1560. IEEE, 2015.

[18] QIN, Z.; TAO, J. ; XIA, Y., **A proportionate recursive least squares algorithm and its performance analysis**, IEEE Transactions on Circuits and Systems II: Express Briefs, 2020.

[19] XU, S.; DE LAMARE, R. C. ; POOR, H. V., **Adaptive link selection algorithms for distributed estimation**, EURASIP Journal on Advances in Signal Processing, vol. 2015, no. 1, pp. 1–22, 2015.

[20] CHEN, F.; HU, L.; LIU, P. ; FENG, M., **A robust diffusion estimation algorithm for asynchronous networks in IoT**, IEEE Internet of Things Journal, vol. 7, no. 9, pp. 9103–9115, 2020.

[21] YU, Y.; ZHAO, H.; DE LAMARE, R. C.; ZAKHAROV, Y. ; LU, L., **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.

[22] BERTRAND, A.; MOONEN, M. ; SAYED, A. H., **Diffusion bias-compensated RLS estimation over adaptive networks**, IEEE Transactions on Signal Processing, vol. 59, no. 11, pp. 5212–5224, 2011.

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

[24] JACOBSSON, S.; DURISI, G.; COLDREY, M.; GUSTAVSSON, U. ; STUDER, C., **Throughput analysis of massive MIMO uplink with low-resolution ADCs**, IEEE Transactions on Wireless Communications, vol. 16, no. 6, pp. 4038–4051, 2017.

[25] JACOBSSON, S.; DURISI, G.; COLDREY, M. ; STUDER, C., **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.

[26] MEZGHANI, A.; KHOUFI, M.-S. ; NOSSEK, J. A., **A modified MMSE receiver for quantized MIMO systems**, Proc. ITG/IEEE WSA, Vienna, Austria, pp. 1–5, 2007.

[27] SHAO, Z.; LANDAU, L. ; DE LAMARE, R. C., **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, pp. 1–4. VDE, 2018.

[28] ARABLOUEI, R.; DOĞANÇAY, K.; WERNER, S. ; HUANG, Y. F., **Adaptive distributed estimation based on recursive least-squares and partial diffusion**, IEEE transactions on signal processing, vol. 62, no. 14, pp. 3510–3522, 2014.

[29] ARABLOUEI, R.; WERNER, S.; HUANG, Y. ; DOĞANÇAY, K., **Distributed least mean-square estimation with partial diffusion**, IEEE Transactions on Signal Processing, vol. 62, no. 2, pp. 472–484, 2013.

[30] LIN, W.; CAO, J. ; LIU, X., **E^3: Towards energy-efficient distributed least squares estimation in sensor networks**, In: 2014 IEEE 22ND

INTERNATIONAL SYMPOSIUM OF QUALITY OF SERVICE (IWQOS), pp. 21–30. IEEE, 2014.

[31] SATTLER, F.; WIEDEMANN, S.; MÜLLER, K.-R. ; SAMEK, W., **Robust and communication-efficient federated learning from non-iid data**, IEEE transactions on neural networks and learning systems, vol. 31, no. 9, pp. 3400–3413, 2019.

[32] KHAN, L. U.; SAAD, W.; HAN, Z.; HOSSAIN, E. ; HONG, C. S., **Federated learning for internet of things: Recent advances, taxonomy, and open challenges**, IEEE Communications Surveys & Tutorials, 2021.

[33] NGUYEN, D. C.; DING, M.; PATHIRANA, P. N.; SENEVIRATNE, A.; LI, J. ; POOR, H. V., **Federated learning for internet of things: A comprehensive survey**, IEEE Communications Surveys & Tutorials, 2021.

[34] ALIOTO, M.; SHAHGHASEMI, M., **The internet of things on its edge: Trends toward its tipping point**, IEEE Consumer Electronics Magazine, vol. 7, no. 1, pp. 77–87, 2017.

[35] HAYKIN, S. S., *Adaptive filter theory*, Pearson Education India, 2008.

[36] DE LAMARE, R. C.; SAMPAIO-NETO, R., **Reduced-rank adaptive filtering based on joint iterative optimization of adaptive filters**, IEEE Signal Processing Letters, vol. 14, no. 12, pp. 980–983, 2007.

[37] DE LAMARE, R. C.; SAMPAIO-NETO, R., **Adaptive reduced-rank processing based on joint and iterative interpolation, decimation, and filtering**, IEEE Transactions on Signal Processing, vol. 57, no. 7, pp. 2503–2514, 2009.

[38] FA, R.; DE LAMARE, R. C. ; WANG, L., **Reduced-rank stap schemes for airborne radar based on switched joint interpolation, decimation and filtering algorithm**, IEEE Transactions on Signal Processing, vol. 58, no. 8, pp. 4182–4194, 2010.

[39] XIAO, L.; BOYD, S., **Fast linear iterations for distributed averaging**, Systems & Control Letters, vol. 53, no. 1, pp. 65–78, 2004.

[40] XIAO, L.; BOYD, S. ; LALL, S., **A scheme for robust distributed sensor fusion based on average consensus**, In: IPSN 2005. FOURTH INTERNATIONAL SYMPOSIUM ON INFORMATION PROCESSING IN SENSOR NETWORKS, 2005., pp. 63–70. IEEE, 2005.

[41] SAYED, A. H.; TU, S.; CHEN, J.; ZHAO, X. ; TOWFIC, Z. J., **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.

[42] SAYED, A. H., **Adaptation, learning, and optimization over networks**, Foundations and Trends in Machine Learning, vol. 7, no. ARTICLE, pp. 311–801, 2014.

[43] SAYED, A. H., **Adaptive networks**, Proceedings of the IEEE, vol. 102, no. 4, pp. 460–497, 2014.

[44] TU, S.; SAYED, A. H., **Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks**, IEEE Transactions on Signal Processing, vol. 60, no. 12, pp. 6217–6234, 2012.

[45] DINIZ, P. S.; OTHERS, *Adaptive filtering*, vol. 4, Springer, 1997.

[46] LLOYD, S., **Least squares quantization in PCM**, IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129–137, 1982.

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

[48] ORHAN, O.; ERKIP, E. ; RANGAN, S., **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), pp. 191–198. IEEE, 2015.

[49] RATASUK, R.; VEJLGAARD, B.; MANGALVEDHE, N. ; GHOSH, A., **NB-IoT system for M2M communication**, In: 2016 IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE, pp. 1–5. IEEE, 2016.

[50] CHUNG, H.; RYLYAKOV, A.; DENIZ, Z. T.; BULZACCHELLI, J.; WEI, G. ; FRIEDMAN, D., **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, pp. 268–269. IEEE, 2009.

[51] DI LORENZO, P.; BATTILORO, C.; MERLUZZI, M. ; BARBAROSSA, S., **Dynamic resource optimization for adaptive federated learning at the wireless network edge**, In: ICASSP 2021-2021 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), pp. 4910–4914. IEEE, 2021.

[52] GOGINENI, V. C.; WERNER, S.; HUANG, Y. ; KUH, A., **Communication-efficient online federated learning framework for nonlinear regression**, arXiv preprint arXiv:2110.06556, 2021.

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

[54] BUSSGANG, J. J., **Crosscorrelation functions of amplitude-distorted gaussian signals**, Tech. Rep. 216, Research Laboratory of Electronics, Massachusetts Institute of Technology, 1952.

[55] KOUKOULAS, P.; KALOUPTSIDIS, N., **Nonlinear system identification using gaussian inputs**, IEEE Transactions on Signal Processing, vol. 43, no. 8, pp. 1831–1841, 1995.

[56] KOH, T.; POWERS, E., **Second-order volterra filtering and its application to nonlinear system identification**, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 33, no. 6, pp. 1445–1455, 1985.

[57] VAN DER MERWE, R.; WAN, E. A., **The square-root unscented kalman filter for state and parameter-estimation**, In: 2001 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING. PROCEEDINGS (CAT. NO. 01CH37221), vol. 6, pp. 3461–3464. IEEE, 2001.

[58] SAYED, A. H., *Fundamentals of adaptive filtering*, John Wiley & Sons, 2003.

[59] DANAEE, A.; DE LAMARE, R. C. ; NASCIMENTO, V. H., **Energy-efficient distributed learning with coarsely quantized signals**, IEEE Signal Processing Letters, vol. 28, pp. 329–333, 2021.

[60] DANAEE, A.; DE LAMARE, R. C. ; NASCIMENTO, V. H., **Energy-efficient distributed learning with adaptive bias compensation for coarsely quantized signals**, In: 2021 IEEE STATISTICAL SIGNAL PROCESSING WORKSHOP (SSP), pp. 61–65. IEEE, 2021.

[61] NASCIMENTO, V. H.; SILVA, M. T. M., **Adaptive filters**, In: ACADEMIC PRESS LIBRARY IN SIGNAL PROCESSING, vol. 1, pp. 619–761. Elsevier, 2014.

[62] DANAEE, A.; DE LAMARE, R. C. ; NASCIMENTO, V. H., **Distributed quantization-aware RLS learning with bias compensation and coarsely quantized signals**, IEEE Transactions on Signal Processing, vol. 70, pp. 3441–3455, 2022.

[63] DANAEE, A.; DE LAMARE, R. C. ; NASCIMENTO, V. H., **Energy-efficient distributed recursive least squares learning with coarsely**

quantized signals, In: 2020 54TH ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS, AND COMPUTERS, pp. 1533–1537. IEEE, 2020.

[64] DANAEE, A.; DE LAMARE, R. C. ; NASCIMENTO, V. H., **Quantization-aware federated learning with coarsely quantized measurements**, In: 2022 30TH EUROPEAN SIGNAL PROCESSING CONFERENCE (EUSIPCO), pp. 1691–1695. IEEE, 2022.

[65] ZAKHAROV, Y. V.; NASCIMENTO, V. H.; DE LAMARE, R. C. ; NETO, F. G. D. A., **Low-complexity DCD-based sparse recovery algorithms**, IEEE Access, vol. 5, pp. 12737–12750, 2017.

[66] YU, Y.; LU, L.; ZHENG, Z.; WANG, W.; ZAKHAROV, Y. ; DE LAMARE, R. C., **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.

[67] GERSHO, A.; GRAY, R. M., *Vector quantization and signal compression*, vol. 159, Springer Science & Business Media, 2012.

A
Theoretical expressions for MSD values

In this section, we obtain the theoretical expressions for $\mathbf{R}_{\overline{x}_k}$, $\mathbf{R}_{q_x,k}$, and $\sigma^2_{p_k}$ 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 (4-38a) and (2-31), $\mathbf{R}_{q_x,k}$ can be obtained as follows:

$$\mathbf{R}_{q_x,k} = \mathbf{R}_{\overline{x}_k} - \mathbf{G}_{x_k}\mathbf{R}_{x_k}\mathbf{G}_{x_k}, \tag{A-1}$$

where $\mathbf{G}_{x_k}$ is computed as in (2-33) with $\mathbf{R}_{x_k}$. We evaluate the covariance matrix $\mathbf{R}_{\overline{x}_k}$ whose diagonal entries are given by

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

where $\sigma^2_{x_k}(m) = \mathbb{E}[|\mathbf{x}_k(m)|^2] = [\mathbf{R}_{x_k}]_{m,m}$, $\Phi(.)$ 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 $\mathbf{R}_{\overline{x}_k}$ for $1 < m, n < M$ and $m \neq n$ are given by

$$\begin{aligned}
[\mathbf{R}_{\overline{x}_k}]_{m,n} &= \mathbb{E}[\overline{\mathbf{x}}_k(m)\overline{\mathbf{x}}_k^*(n)] = \sum_{j=0}^{2^b-1}\sum_{p=0}^{2^b-1} l_j l_p \mathbb{P}[\overline{\mathbf{x}}_k(m) = l_j, \overline{\mathbf{x}}_k(n) = l_p] \\
&= \sum_{j=0}^{2^b-1}\sum_{p=0}^{2^b-1} l_j l_p \mathbb{P}[\tau_j \le \mathbf{x}_k(m) < \tau_{j+1}, \tau_p \le \mathbf{x}_k(n) < \tau_{p+1}].
\end{aligned} \tag{A-3}$$

Unfortunately, (A-3) does not have a known closed-form expression and hence, has to be evaluated using numerical methods [25]. However, in what follows, we

shall present a closed-form approximation for the off-diagonal entries of $\mathbf{R}_{\overline{x}_k}$, following [26]. Let us rewrite (4-38a) as follows:

$$\overline{\mathbf{x}}_k(i) = \mathbf{G}_{x_k}\mathbf{x}_k(i) + \mathbf{q}_{x_k}(i) = \mathbf{x}_k(i) + \boldsymbol{\epsilon}_k(i), \tag{A-4}$$

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 $\rho_q(i)$ to indicate the relative amount of quantization error generated, which is defined as follows:

$$\rho_q(i) = \frac{\sigma^2_{\epsilon_k}(i)}{\sigma^2_{x_k}(i)}, \tag{A-5}$$

where $\sigma^2_{x_k}(i)$ is the variance of the input and the distortion factor $\rho_q(i)$ depends on the number of quantization bits b, the quantizer type (uniform or non-uniform) and the probability density function of $\mathbf{x}_k(i)$ [26]. For Gaussian inputs and a scalar non-uniform quantizer, e.g., Lloyd-Max quantizer, the distortion factor $\rho_q(i) = \rho_q$ can be obtained from Table A.1 in [47] for different b and asymptotically approximated by $\rho_q = \frac{\pi\sqrt{3}}{2}2^{-2b}$ for $b > 5$ [67]. Based on this, we obtain an approximation of $\mathbf{R}_{\overline{x}_k}$ as follows [26]:

$$\widehat{\mathbf{R}}_{\overline{x}_k} \approx (1-\rho_q)\Big(\mathbf{R}_{x_k} - \rho_q\,\mathrm{nondiag}\{\mathbf{R}_{x_k}\}\Big), \tag{A-6}$$

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

$$\mathrm{nondiag}\{\mathbf{R}_{\overline{x}_k}\} = \widehat{\mathbf{R}}_{\overline{x}_k} - \mathrm{diag}\{\widehat{\mathbf{R}}_{\overline{x}_k}\}. \tag{A-7}$$

Note that we compute the diagonal elements of $\mathbf{R}_{\overline{x}_k}$ directly using (A-2) instead of using the approximation in (A-6), to get an accurate expression for $[\mathbf{R}_{\overline{x}_k}]_{m,m}$, and approximate the off-diagonal entries $[\mathbf{R}_{\overline{x}_k}]_{m,n}$ from (A-7).

We now find a closed-form expression for $\sigma^2_{p_k}$, where $p_k(i) = g_{d_k}v_k(i) + q_{d_k}(i)$ and

$$\sigma^2_{p_k} = \mathbb{E}[p_k(i)p^*_k(i)] = g^2_{d_k}\sigma^2_{v_k} + \mathbb{E}[q_{d_k}(i)q^*_{d_k}(i)]. \tag{A-8}$$

From (4-38b), we have

$$\sigma^2_{q_{d_k}} = \mathbb{E}[q_{d_k}(i)q^*_{d_k}(i)] = \sigma^2_{\overline{d}_k} - g^2_{d_k}\sigma^2_{d_k}, \tag{A-9}$$

where considering the data model in (2-3), $\sigma^2_{d_k}$ is given by

$$\sigma^2_{d_k} = \mathbb{E}[d_k(i)d^*_k(i)] = \mathbf{w}^*_o\mathbf{R}_{x_k}\mathbf{w}_o + \sigma^2_{v_k}. \tag{A-10}$$

Finally, using the same evaluation as in (A-2), the variance of $\overline{d}_k$ is given by

$$\begin{aligned}\sigma^2_{\overline{d}_k} &= \mathbb{E}[\overline{d}_k(i)\overline{d}^*_k(i)] = h\sum_{j=0}^{2^b-1} l_j^2\mathbb{P}[\overline{\mathbf{x}}_k(m) = l_j] \\ &= h\sum_{j=0}^{2^b-1} l_j^2\mathbb{P}[\tau_j \le d_k = l_j < \tau_{j+1}] \\ &= h\sum_{j=0}^{2^b-1} l_j^2\bigg(\Phi\Big(\frac{\sqrt{h}\tau^2_{j+1}}{\sigma_{d_k}}\Big) - \Phi\Big(\frac{\sqrt{h}\tau^2_{j}}{\sigma_{d_k}}\Big)\bigg).\end{aligned} \tag{A-11}$$

Table A.1: Distortion factor ρ_q for different ADC resolutions b [47]

b	1	2	3	4	5
ρ_q	0.3634	0.1175	0.03454	0.009497	0.002499

B
Computational complexity of DQA-RLS algorithm per time instant at in Detail

Table B.1: Computational complexity of DQA-RLS per time instant at node k

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	3	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
$\beta_{k,b}(i)\mathbf{w}_k^H(i-1)\mathbf{x}_{k,Q}(i)$	$4M-2$	$4M+2$	0	0
$e_{k,Q}(i) = d_{k,Q}(i) - \mathbf{w}_k^H(i-1)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)$	2	0	0	0
$\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)$	$4M^2-2M$	$4M^2$	0	0
$\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)$	$4M-2$	$4M$	0	0
$\lambda + \beta^2_{k,b}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)$	1	2	0	0
$\frac{\beta^2_{k,b}(i)}{\lambda+\beta^2_{k,b}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)}$	0	0	1	0
$\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)\frac{\beta^2_{k,b}(i)}{\lambda+\beta^2_{k,b}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)}$	0	$2M$	0	0
$\frac{\beta^2_{k,b}(i)\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)}{\lambda+\beta^2_{k,b}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i)\mathbf{x}_{k,Q}(i)}\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i)$	$2M^2$	$4M^2$	0	0
$\mathbf{P}_k(i-1) - \frac{\beta^2_{k,b}(i)\mathbf{P}_k(i-1)\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i-1)}{\lambda+\beta^2_{k,b}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i-1)\mathbf{x}_{k,Q}(i)}$	$2M^2$	0	0	0
$\mathbf{P}_k(i) = \frac{1}{\lambda}\left(\mathbf{P}_k(i-1) - \frac{\beta^2_{k,b}(i)\mathbf{P}_k(i-1)\mathbf{x}_{k,Q}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i-1)}{\lambda+\beta^2_{k,b}(i)\mathbf{x}^*_{k,Q}(i)\mathbf{P}_k(i-1)\mathbf{x}_{k,Q}(i)}\right)$	0	$2M^2$	0	0
$\mathbf{P}_k(i)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)e_{k,Q}(i)$	$2M$	$4M+2$	0	0
$\mathbf{h}_k(i) = \mathbf{h}_k(i-1) + \mathbf{P}_k(i)\beta_{k,b}(i)\mathbf{x}_{k,Q}(i)e_{k,Q}(i)$	$2M$	0	0	0
$\mathbf{w}_k(i) = \sum_{l\in\mathcal{N}_k} a_{lk}\mathbf{h}_l(i)$	$2n_kM$	$4n_kM$	0	0
Total	$8M^2 + 2^b + 2 + (2n_k+10)M$	$10M^2 + 2^{b+1} + (4n_k+14)M + 12$	2^b+3	2^b