Efficient quantization and fixed-point representation for MIMO turbo-detection and turbo-demapping

Mostafa Rizk, Amer Baghdadi, Michel Jezequel, Yasser Mohanna, Youssef Atat

Rizk *et al. EURASIP Journal on Embedded Systems* (2017) 2017:33
DOI 10.1186/s13639-017-0081-y

EURASIP Journal on Embedded Systems

RESEARCH Open Access

Efficient quantization and fixed-point representation for MIMO turbo-detection and turbo-demapping

Mostafa Rizk[1,2,3*], Amer Baghdadi[2], Michel Jézéquel[2], Yasser Mohanna[3] and Youssef Atat[3]

Abstract

In the domain of wireless digital communication, floating-point arithmetic is generally used to conduct performance evaluation studies of algorithms. This is typically limited to theoretical performance evaluation in terms of communication quality and error rates. For a practical implementation perspective, using fixed-point arithmetic instead of floating-point reduces significantly implementation costs in terms of area occupation and energy consumption. However, this implies a complex conversion process, particularly if the considered algorithm includes complex arithmetic operations with high accuracy requirements and if the target system presents many configuration parameters. In this context, the purpose of the paper is to present an efficient quantization and fixed-point representation for turbo-detection and turbo-demapping. The impact of floating-to-fixed-point conversion is illustrated upon the error-rate performance of the receiver for different system configurations. Only a slight degradation in the error-rate performance of the receiver is observed when implementing the detector and demapper modules which utilize the devised quantization and fixed-point arithmetic rather than floating-point arithmetic.

Keywords: Turbo-detection, Turbo-demapping, Fixed-point, Quantization

1 Introduction

Low power consumption and reduced implementation area are vital factors to fulfill the ever increasing requirements of embedded systems. In digital communication applications, the algorithms are typically specified in floating-point arithmetic in order to evaluate the application performance. However, hardware architectures implementing these applications are designed using fixed-point arithmetic to satisfy the tight constraints on implementation area and power consumption related to embedded systems. In fixed-point representation, memory and bus widths are smaller, leading to definitively lower cost and power consumption. Moreover, floating-point operators are more complex, having to deal with the exponent and the mantissa, and hence, their area and latency are significantly greater than those of fixed-point operators. In spite of that, fixed-point arithmetic introduces an unalterable quantization error which modifies the application functionalities and degrades the desired performance. Thus, the design flow requires a floating-to-fixed-point conversion stage which optimizes the implementation cost under execution time and accuracy constraints [1]. For digital communication applications, the most commonly used criterion for evaluating the precision of fixed-point implementation is the error-rate performance degradation. Hence, the accuracy constraint is linked to the specifications of the supported communication standards.

On the other hand, due to the rapid evolution of related standards, modern wireless digital communication systems are highly concerned about the flexibility feature. Circuits and systems adopted in this application domain must consider not only performance and implementation constraints, but also the requirement of flexibility. In this context, flexible application-specific hardware architectures implementing the functionalities of digital baseband components of the receiver are under design scope.

*Correspondence: mostafa.rizk@liu.edu.lb
[1]Lebanese International University, School of Engineering, Mousaytbeh, Beirut, Lebanon
[2]Institut Mines-Telecom, IMT Atlantique, CNRS UMR 6285 Lab-STICC, Brest, France
Full list of author information is available at the end of the article

Application-specific processors constitute a key trend in implementing definite blocks of wireless system since they provide a good solution in designing flexible architectures that can fulfill nowadays requirements in terms of low error-rate performance and high throughput and satisfy the tight constraints on implementation area and power consumption.

Recent emergent wireless communication standards, such as LTE/LTE-A for mobile phones, 802.11 (WiFi) and 802.16 (WiMAX) for wireless local and wide area networks, and DVB for digital video broadcasting, support various modes and configurations related to channel coding type, modulation type and mapping style, and antenna dimension for multiple-input multiple-output (MIMO) transmission techniques. On the other hand, iterative concept is also utilized at the receiver side to alleviate the destructive effects of the channel. Iterative processing concept (so-called turbo processing) was proposed firstly in the channel decoding [2] to achieve error-rate performance close to the theoretical limits.

The extension of turbo principle to the demapping and inter-symbol interference (ISI) equalization blocks gives rise to turbo-demapping [3] and turbo-equalization [4] concepts. These concepts are achieved when the extrinsic information at the output of the turbo decoder is fed back as a priori soft information to the input of the demapper and equalizer.

In previous work, presented in [5], flexible application-specific processor dedicated for turbo-demapping has been proposed. The demapper implements the Max-Log-MAP algorithm. It supports iterative demodulation and its flexibility is not restricted to certain types of modulation and/or mapping styles. Similarly, another flexible application-specific processor dedicated for minimum mean-squared error (MMSE) linear equalizer has been proposed in [6]. Its flexibility is extracted from the following requirements: (1) the capability to support different MIMO schemes reaching to 4×4 antenna dimension, (2) the ability to maintain efficient use of hardware resources for different time diversity channel types (fast fading, quasi-static, and block fading) and (3) the possibility to execute in an iterative or non-iterative modes. In fact, the techniques which were found to be effective in combating ISI are often extended to the context of MIMO detection [7, 8]. Therefore, the designed MMSE equalizer in [6] is used for iterative MIMO detection. In the remainder of this paper, iterative MIMO detection based on MMSE linear equalization is referred to as turbo-equalization.

In the designed architectures for Max-Log-MAP demapping [5] and MMSE equalization [6], fixed-point arithmetic has been adopted. In addition, the input data, the output data, and the intermediate computational values have been quantized according to defined precisions. Due to truncation and rounding processes, quantization errors occur. These errors propagate through the computational steps of the algorithms, and they are exacerbated in iterative schemes leading to a divergence at the output. In order to maintain the numerical stability of the algorithms and to ensure that quantization errors induce only small errors in the final result, a careful numerical study has been conducted. An accurate quantization and fixed-point representation of all parameters and computational values involved in both algorithms have been determined.

Despite quantization approach and its corresponding evaluation are considered mandatory tasks in the design flow and can take much time and effort, their presentation are rarely published in the literature. Only few number of works have illustrated the used quantization and fixed-point representation. In [9] and [10], the implementation of a low-complexity turbo-equalizer has been presented targeting a 16-b fixed-point DSP device with two's-complement arithmetic. The authors have focused on BPSK signal set and only presented its corresponding simulation results. In addition, the quantization of input and output values of the main modules constituting the equalizer has been given. The precise quantization of intermediate computation result values have not been shown. Moreover, the authors has focused on exploiting a given word length without illustrating the fixed-point representation. In [11], a fixed-point representation of MMSE-based turbo equalizer with soft cancelation (SC) has been presented. The authors have targeted specific constellation scheme (QPSK) and presented the performance comparison between a non-quantized system and quantized system for different system configuration. In [12], the previous work has been extended to 16-QAM constellation scheme. With the help of extrinsic information transfer (EXIT) charts, the authors have determined the sufficient number of fractional bits.

On the other side, in [13], a quantization study of log-likelihood ratios (LLR) in bit-interleaved coded modulation (BICM) systems has been provided. The performance of LLR quantization (1, 2, and 3 b) for MIMO-BICM systems has been investigated for BPSK and 16-QAM constellation schemes. In [14], the quantization and fixed-point representation of few parameters of SISO demapper algorithm have been presented without showing their effect on the demapper performance. In [15], the authors proposed an architecture that supports only 16-QAM modulation scheme. The quantization of input and output has been only provided without mentioning the fixed-point representation.

The purpose of this paper is to present the efficient data quantization and fixed-point representation that are devised for the architectures of MMSE turbo-equalizer [6] and Max-Log-Map turbo-demapper [5]. Moreover, their influence on the receiver error-rate performance is

evaluated for multitude configurations. In this regard, the contribution of this paper can be considered as an important reference. The rest of the paper is organized as follows. The system model is presented in the next section. Sections 3 and 4 describe, respectively, the adopted algorithms for turbo-equalization and turbo-demapping, discuss the required operations to implement the algorithms, present the required fixed-point arithmetic and data quantization, and finally show their influence on error-rate performance. At last, Section 5 concludes the paper.

2 System model

The digital wireless communication system is basically composed of three blocks: transmitter, wireless channel, and receiver. The structures of the transmitter and receiver blocks rely on the specifications of the applied wireless communication standard. In general, data processing before the transmission of source data bits into the wireless channel includes adding redundant information to the original data, and/or rearranging data stream, and/or adding diversity of data. At the receiver side, the input information is distorted by fading and other destructive channel effects. The constituent components of the receiver process the received corrupted data to retrieve the original source data by exploiting the added redundancy and/or diversity. Modern wireless communication systems adopt MIMO technology, which uses multiple antennas at both transmitter and receiver sides of the wireless system, to meet the requirement of high data rate, reliability, and bandwidth efficiency. Iterative concept is also utilized at the receiver side to alleviate the destructive effects of the channel. Passing soft information between different components in the receiver through both forward and feedback paths has shown a prominent improvement of the output over the iterations leading to error-rate performance close to theoretical limits. MIMO technology and iterative processing have been incorporated in many modern wireless communication systems. The overall system model considered in this work is presented in Fig. 1. In the following subsections, the considered models of transmitter, channel, and receiver are briefly explained.

2.1 Transmitter scheme

The transmitter chain is established by concatenating different components to provide immunity to channel effects. Initially, the source bits s, so-called systematic bits, are encoded by a turbo encoder, which concatenates in parallel two eight-state double binary circular recursive systematic convolutional (CRSC) encoders [16, 17]. The output codeword c, that is made up of the source data and parities, is then punctured to reach a desired coding rate R_c. Bit interleaved coded modulation (BICM) [18, 19] is used to disperse the obtained coded binary data sequence to assure that no single coded symbol is fully destroyed while passing through a fading channel. Punctured and interleaved bit stream v is passed to the mapper. Each m-bit combination is mapped to channel symbol x according to the chosen constellation (BPSK till 256-QAM) formed of 2^m symbols. After mapping, the symbols x are transmitted using either single antenna or MIMO techniques. Signal space diversity (SSD) technique [20] can be applied against the fading events in case of single-input-single-output transmission, whereas in case of MIMO transmission, spatial multiplexing (SM) is adopted to improve the transmission rate [21].

2.2 Channel

The considered channel has a flat Rayleigh fading nature with additive white Gaussian noise (AWGN). The channel flat in frequency is a realistic model for several terrestrial mobile radio channels [22], and most works in MIMO literature assume this channel model [23, 24]. For a single-antenna transmission system, the received discrete time baseband complex signal y_k can be expressed as follows [25]:

$$y_k = h_k x_k + w_k \tag{1}$$

where x_k is the complex signal transmitted at time k, h_k is a Rayleigh distributed fading coefficient, and w_k is a complex additive white Gaussian noise.

For MIMO systems with N_t transmit antennas and N_r receive antennas, the relation between channel, transmitted symbols and received symbols is given by the expression below:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{2}$$

where

$$\mathbf{y} = [\, y_1, \ldots, y_{N_r} \,]^T \in \mathbb{C}^{N_r \times 1}$$
$$\mathbf{x} = [\, x_1, \ldots, x_{N_t} \,]^T \in \mathbb{C}^{N_t \times 1}$$
$$\mathbf{w} = [\, \omega_1, \ldots, \omega_{n_R} \,]^T \in \mathbb{C}^{n_R \times 1}$$
$$\mathbf{H} = \begin{bmatrix} h_{11} & \cdots & h_{1N_t} \\ \vdots & \ddots & \vdots \\ h_{N_r 1} & \cdots & h_{N_r N_t} \end{bmatrix}$$

where $\mathbf{y}$ and $\mathbf{x}$ represent, respectively, the received and transmitted symbol vectors, $\mathbf{w}$ represents the AWGN vector, $\mathbf{H}$ is the channel matrix whose element h_{ij} represents the fading coefficient that characterizes the relation between the ith receive antenna and jth transmit antenna. On the other hand, the channel can be further categorized on the basis of time selectivity conditions. The time selectivity characteristic of a channel defines the variation of the channel with respect to time. It is related to the mobility of the transmitter, receiver, or the obstacles between the two depending on the nature of fading. This selectivity characterizes 3 types of channels: (1) fast fading, (2) quasi-static, and (3) block fading.

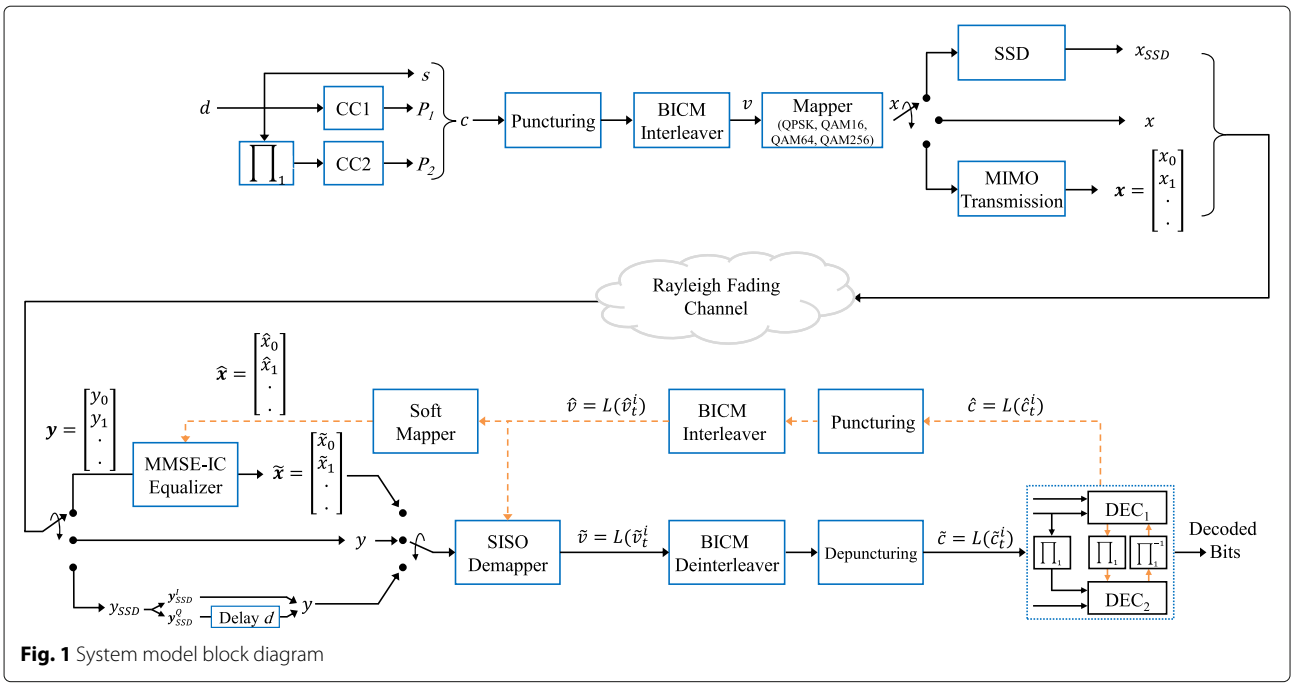

Fig. 1 System model block diagram

2.3 Receiver scheme

At the receiver side, the objective is to remove the channel effects to retrieve the original source data by exploiting the redundancy and diversity added to source information before transmitting data through the channel. Figure 1 shows the structure of the considered iterative receiver. It is characterized by the existence, in addition to forward paths, of feedback paths through which constituent blocks can send the information to previous blocks iteratively. On every new iteration, each block generates soft information depending on channel information and on received a priori soft information generated by other blocks in the previous iteration. The blocks constituting such receiver are referred to as soft-input soft-output (SISO) processing blocks. In case of MIMO, the symbol vector $\mathbf{y}$ is received at the input of the MIMO equalizer, whereas in case of single antenna transmission, y symbol is received directly at the input of the demapper. For cases where SSD is used at the transmitter side, an additional latency similar to the one applied in the transmitter is required at the receiver in order to match in-phase (I) and quadrature (Q) components of received symbols.

Benefiting from a priori information from the feedback path, the MMSE equalizer provides the estimated symbol vector $\tilde{\mathbf{x}}$ and the corresponding equivalent bias vector (fading coefficient) to the demapper. The SISO demapper produces the probabilities $\tilde{v}$ on transmit sequence in the form of log likelihood ratio (LLR), which construct after deinterleaving and depuncturing the input $\tilde{c}$ to the decoder. The turbo decoder uses the Bahl-Cock-Jelinek-Raviv (BCJR) [26] decoding algorithm with Max-log-MAP approximation [27] and outputs the a posteriori information both on systematic and parity bits. This information is punctured and interleaved and then fed back to both SISO demapper and soft mapper. The latter provides a priori information to the equalizer as decoded symbol vector $\hat{\mathbf{x}}$. This iterative process is stopped if a maximum number of iterations is reached. Then, the turbo decoder outputs the decoded bits.

3 MMSE linear equalizer

Turbo-equalization concept was first introduced in [4] to mitigate the detrimental effects of ISI for digital transmission protected by convolution codes. In the emerging wireless standards where MIMO techniques have been inducted, co-channel interference occurs at the receiver side. Co-channel interference is a cause of signal distortion when multiple signals are transmitted on the same frequency slots concurrently [7]. The concept of turbo-equalization can be used to cancel iteratively this interference caused by MIMO. One of the best-known low-complexity approaches to achieve equalization in iterative MIMO systems is referred to as MMSE linear equalization (LE) [28, 29]. This approach is able to significantly lower the computational complexity compared to optimal maximum-likelihood (ML) algorithm. The use of MMSE in iterative scheme reduces the performance loss leading to error-rate results close to ML. At least, 3-dB gain can be obtained for bit error rate (BER) performance, compared to a non-iterative MMSE [28, 30].

3.1 Algorithmic overview

The inputs to the MMSE equalizer are the received symbol vector $\mathbf{y}$ of size N_r, channel matrix $\mathbf{H}$ of size $N_r \times N_t$, and the variance of the AWGN vector σ_w^2. Using this information, the equalizer generates the estimated symbol vector $\tilde{\mathbf{x}}$. The equalizer considers that a symbol of the vector $\mathbf{x}$ is distorted by the N_t-1 other symbols of the vector and by the noise channel and it tries to combat both. Equation (2) can be written in the following form:

$$\mathbf{y} = \mathbf{h}_j.x_j + \sum_{i \neq j} \mathbf{h}_i.x_i + \mathbf{w} \tag{3}$$

where $j \in \{0, N_t - 1\}$, $\mathbf{h}_i$, $\mathbf{h}_j$ are the ith and jth columns of $\mathbf{H}$ matrix and $\mathbf{w}$ is the AWGN noise vector of size N_r. One of the low-complexity techniques to achieve the equalization function is the use of filter-based symbol equalization [29]. An estimation of the symbol x_j can be carried out through a linear filter which minimizes the mean square error (MSE) between the transmitted symbol x_j and the output of the equalizer $\tilde{x}_j$. Using the Wiener filter $\mathbf{a}_j^H = \lambda_j.\mathbf{P}_j^H$, the estimation of $\mathbf{x}$ is given by [28]:

$$\tilde{x}_j = \lambda_j.\mathbf{P}_j^H \left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}} + \mathbf{h}_j\hat{x}_j\right) \tag{4}$$

where $j \in \{0, N_t - 1\}$, $\hat{x}_j$ is the jth element of vector $\hat{\mathbf{x}}$, $\mathbf{h}_j$ is the jth column of $\mathbf{H}$ matrix, and $(.)^H$ is the Hermitian operator. $\mathbf{P}_j$ and λ_j are defined as follows:

$$\mathbf{P}_j = \mathbf{E}^{-1}\mathbf{h}_j \tag{5}$$

where

$$\mathbf{E} = \left(\sigma_x^2 - \sigma_{\hat{x}}^2\right)\mathbf{H}.\mathbf{H}^H + \sigma_w^2.\mathbf{I} \tag{6}$$

and σ_x^2, $\sigma_{\hat{x}}^2$, and σ_w^2 are variances of transmitted symbols, decoded symbols, and noise, respectively. $\mathbf{I}$ is the identity matrix of size $N_r \times N_r$.

$$\lambda_j = \frac{\sigma_x^2}{1 + \sigma_{\hat{x}}^2\beta_j} \tag{7}$$

where

$$\beta_j = \mathbf{P}_j^H\mathbf{h}_j \tag{8}$$

Equation (4) can be written as:

$$\tilde{x}_j = \lambda_j.\mathbf{P}_j^H(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) + g_i\hat{x}_i \tag{9}$$

where

$$g_j = \lambda_j.\beta_j \tag{10}$$

During the first iteration of turbo-equalization process, no a priori information is presented ($\hat{\mathbf{x}}$ is a null vector and $\sigma_{\hat{x}}^2 = 0$) and the symbols are equiprobable. The estimated symbol becomes as following:

$$\tilde{x}_j = \lambda_j.\mathbf{P}_j^H\mathbf{y} \tag{11}$$

where

$$\lambda_j = \sigma_x^2 \tag{12}$$

and

$$\mathbf{P}_j = \mathbf{E}^{-1}\mathbf{h}_j = \left(\sigma_x^2\mathbf{H}.\mathbf{H}^H + \sigma_w^2.\mathbf{I}\right)^{-1}\mathbf{h}_j \tag{13}$$

From the second iteration, the a priori information, which is provided by channel decoder about transmitted symbols, improves gradually over the iterations and approaches to asymptotic performance. Asymptotic performance is achieved when the a priori data is perfect, i.e., becomes equal to the transmitted data ($\hat{x}_j = x_j$).

3.2 MMSE algorithm towards implementation

The above-listed expressions exhibit three main computations steps:

1. Detection vector computation referred by $\mathbf{P}$ in (5)
2. Equalization coefficients computation referred by β, λ, and $\mathbf{g}$ in (8), (7), and (10)
3. Estimated symbols computation referred by $\tilde{\mathbf{x}}$ (9)

A closer look at the expressions required in MMSE algorithm ((4) to (10)) reveals the serial nature of the implied elementary computations. Firstly, one need to compute serially the detection vector ($\mathbf{P}$) and the equalization coefficients (β, λ, and $\mathbf{g}$) due to their related dependency, and then symbols are estimated using these coefficients. Furthermore, the expressions performed to fulfill the equalization tasks of computing the detection vector and coefficients and estimating symbols have similar arithmetic operations. But since the computed coefficients are involved in symbol estimation process, the two tasks are executed at different times.

Furthermore, at each iteration, new value of decoded symbols variance $\sigma_{\hat{x}}^2$ (6) is delivered to the equalizer imposing the re-computation of β, λ, $\mathbf{g}$, and $\mathbf{P}$ for all channel selectivity types. In fact, these values also depend on the channel matrix $\mathbf{H}$ (6), which entries change according to the time selectivity of the channel. Hence, the time diversity of the channel decides how frequent the computations of detection vector and equalization coefficients are required. These computations are recomputed repeatedly for each received vector in case of fast fading channel, once for a set of received vectors for which channel matrix is considered as constant in case of block fading channels and once for all received vectors of the frame in case of quasi-static channel. Thus, the channel type (fast fading, quasi-static, or block fading) specifies the computation overhead per iteration. To ensure efficiency and flexibility related to time selectivity of the channel, hardware operators are shared among all required computations in order to take into account the required treatment of data flow for each channel type.

Another flexibility requirement is related to antenna dimension. To cope with diverse configurations which are

imposed by the emerging communication standards, different MIMO schemes are supported. In order to maintain efficiency and to meet the requested flexibility requirement, the hardware implementation considers the lowest complex configuration (2×2) and applies a hardware resource sharing technique to support the other high-order configurations. To manage variable size complex matrix operations that are involved in the MMSE equalization algorithm, complex matrix operations are decomposed into basic real arithmetic operations. The required operations to perform coefficient computations and symbol estimation can be categorized into complex number operations and complex matrix operations.

Complex number addition, subtraction, and negation are performed using real operators as shown in Fig. 2. Complex number multiplication is reformulated to reduce the number of required multiplication operations. Figure 3 shows the real operators used in complex number multiplication operation.

Complex matrix operations involved in MMSE such as matrix addition, subtraction, conjugation, and multiplication are broken down into basic complex number operations. The Hermitian of a complex matrix can be viewed as matrix conjugation followed by a transposition (swapping columns for rows in the matrix). As an example of complex matrix operations decomposition, Fig. 4 shows the required operators to perform 2×2 complex matrix multiplication operation. In the figure, each complex multiplier and each complex adder integrates the real operators presented in Figs. 3 and 2a respectively.

For matrix inversion, the analytical method is used for 2×2 matrix inversion which is given by:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{14}$$

In case of 4×4 matrix, the matrix is first divided into four sub-matrices and then inverted in a block-wise manner by using the following formula:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{W} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} \end{bmatrix} \tag{15}$$

where

$$\begin{aligned} \mathbf{W} &= \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{A}^{-1} \\ \mathbf{X} &= -\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1} \\ \mathbf{Y} &= -\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{A}^{-1} \\ \mathbf{Z} &= \left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1} \end{aligned} \tag{16}$$

and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{W}$, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are 2×2 matrices. In case of 3×3 matrix, the matrix can be extended firstly to a 4×4 matrix and then inverted by applying the same formula derived above for 4×4 matrix inversion. The extending is done by copying all three rows of 3×3 matrix into first three rows of 4×4 matrix and then putting zeros in all elements of fourth row and fourth column except in their intersection where one should be placed. The final result lies in the first three elements of first three rows and first three columns.

Based on the positive-definite property of the matrix resulting from the multiplication of the MIMO channel matrix $\mathbf{H}$ by its Hermitian $\mathbf{H}^H$, β_j values and the matrices determinants ($\Delta_{\mathbf{E}}$), ($\Delta_{\mathbf{A}}$), and ($\Delta_{\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B}}$) are proved to be positive real numbers. Hence, there is no need to implement the computationally demanding complex number inversion operations required in the computations of determinants and λ_j values (7). Real inversion operations can be applied instead.

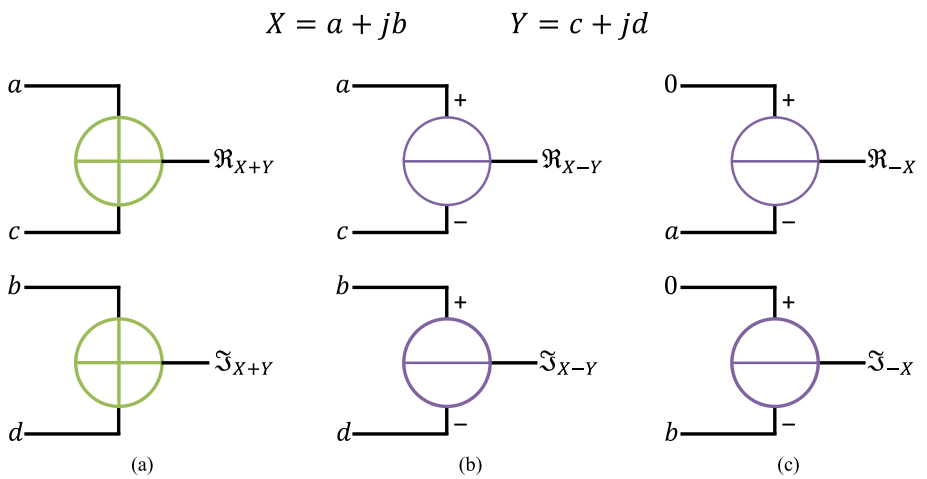

Fig. 2 Operators used in **a** complex number addition, **b** complex number subtraction, and **c** complex number negation

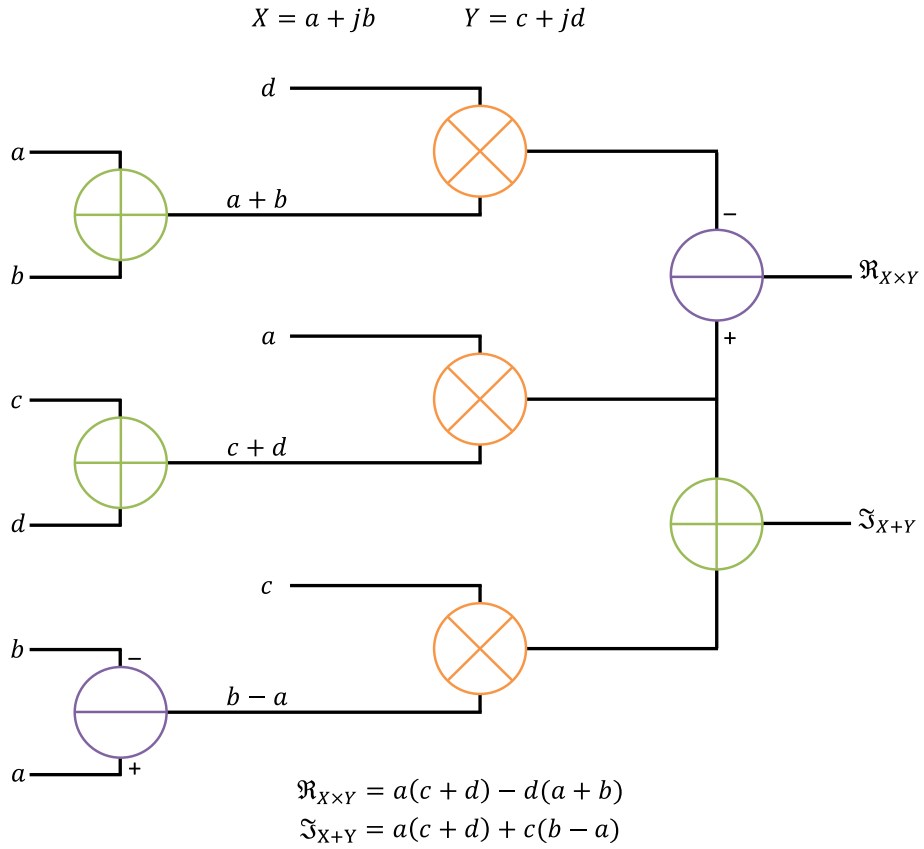

Fig. 3 Operators used in complex number multiplication

Inversion process is preferably replaced by look-up table (LUT). LUT is appraised as an efficient implementation of inversion process by using memory instead of large numbers of logical elements. Both resource utilization and propagation delay are reduced at the cost of accuracy. The utilized LUT should contain all possible inverse values. The value x intended to be inverted is used directly as the LUT index (address) to retrieve the inverse value $\frac{1}{x}$.

3.3 Quantization and fixed-point arithmetic

The aim of quantization and fixed-point arithmetic is to minimize the implementation cost. However, a minimum computational accuracy must be guaranteed to maintain the application performance. A careful numerical study has been conducted to determine the accurate quantization and fixed-point representation of all parameters and computational values involved in the algorithm. The implementation cost is minimized as long as the equalizer performance is fulfilled.

A fixed-point data is made up of integer part and fractional part. The number of bits required for integer part is defined from the dynamic range of the data in order to avoid the occurrence of an overflow [31]. The re-usability and sharing of resources implies that the allocated registers and operators deal with multiple computational values which have different dynamic ranges and variable precisions. So, the bit-width for all components have to be fixed and their precisions vary according to the data requirements by choosing different bits for integer and fractional parts. Long numerical simulations have been conducted for different configurations to find the required data width and accurate precisions for fixed-point representation of all parameters involved in MMSE LE algorithm. Utilizing 16-b two's complement representation with different bits for integer and fractional part in different computation steps shows low performance degradation. Using fixed-point representation demands establishing a virtual decimal point placed in between two bit locations for a given length of data. Figure 5 and Table 1 illustrate the devised quantization and fixed-point representation of different parameters for MMSE algorithm and matrix inversion.

For different modulation types, the quantization values are shown in Table 1 in signed two's complement representation using the notation $Q[I].[F]$ where $[I]$

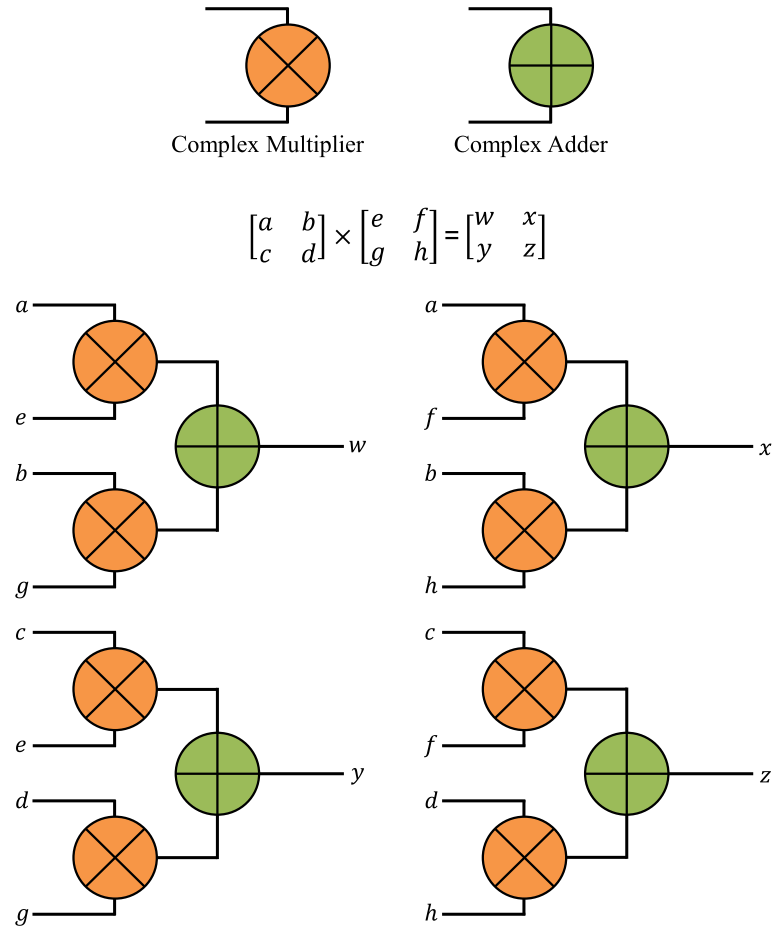

Fig. 4 Operators used in 2 × 2 complex matrix multiplication

and $[F]$ designate the number of bits for integer part and fractional part, respectively. In all algorithmic steps, fixed-point arithmetic is used. First of all, input data that is presented in less than 16-b is extended to 16-b by adding zeros in the added lower bits. Secondly, for all addition/subtraction operations, the operands (addends/subtrahend and minuend) should have the same precision. In case of overflow/underflow, the total/difference is directly set to the most positive/most negative value. Finally, after each multiplication, the double-precision product is converted to 16-b by eliminating the m least significant bits (LSB) and $16 - m$ most significant bits (MSB) where $m = F_a + F_b - F_c$ and F_a, F_b, and F_c represent the number of fractional part of the multiplicand, multiplier, and product, respectively. Figure 6 illustrates an example for the adopted technique to quantize the product value. The multiplicand and multiplies are represented in $Q[10].[6]$ and $Q[5].[11]$, respectively, whereas, the product is represented in $Q[6].[10]$. In fact, the multiplication of these two 16-b values results in 32-b product value that can be represented in $Q[15].[17]$. From the expression above, we have $m = 7$, and thus to accommodate the product in the target quantization representation ($Q[6].[10]$), the 7 LSB are truncated as well as the 9 MSB.

An overflow/underflow is detected if the multiplicand and multiplier have same/opposite signs, and the product is greater/smaller than the most positive/most negative value. In such case, the product is fixed to the most positive/most negative value. Last of all, the inversion operation of real numbers is achieved by the assist of a single $\frac{1}{x}$LUT instead of undergoing expensive computations. The LUT contains 16-b positive values. At each index, the stored value represents the quantized inverse of the index value.

3.4 Performance evaluation

One of the most critical parts of the quantization process is the evaluation of the degradation of the application performance. A software model of the MMSE equalizer

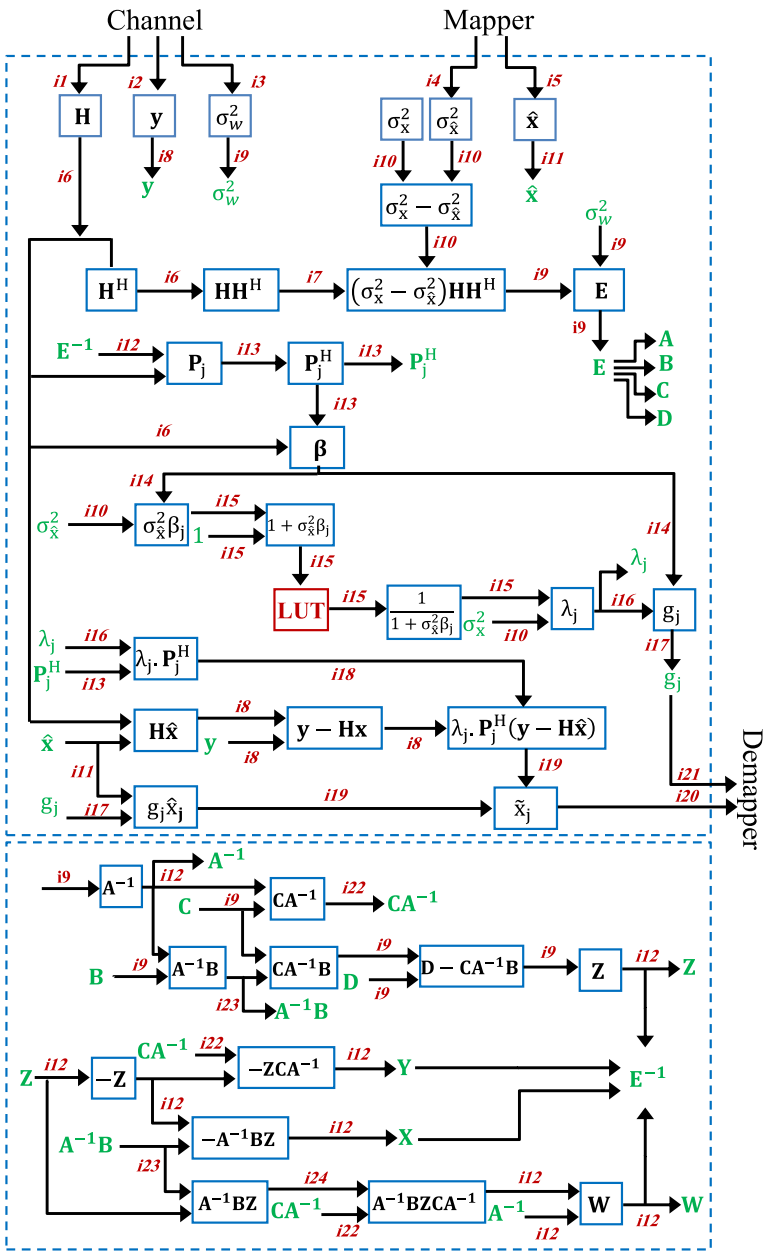

Fig. 5 Parameter quantization for MMSE algorithm and matrix inversion

has been developed to examine the impact of the devised quantization and the adopted fixed-point arithmetic on the error-rate performance. The model with quantization and fixed-point specifications is simulated for different system configurations, and the corresponding error-rate performance is measured. Figure 7 presents the obtained frame error-rate (FER) performance for 4 × 4 MIMO SM with QPSK, 16-QAM, and 64-QAM. In addition, the obtained FER results are compared to corresponding FER of a reference floating-point model. The analysis of the results has shown a performance loss below 0.2 dB for 64-QAM and below 0.1 dB for 16-QAM and QPSK at FER = 10^{-3}. Note that the FER values are recorded for 100 erroneous frames for each $\frac{E_b}{N_0}$ value.

4 Max-Log-MAP demapper

Iterative demapping was proposed firstly in [3] based on bit interleaved coded modulation (BICM) with additional soft feedback from the SISO convolutional decoder to the

Table 1 Quantization parameters related to Fig. 5 in signed two's complement representation

Index	Quantization			Index	Quantization		
	64-QAM	16-QAM	QPSK		64-QAM	16-QAM	QPSK
i1	Q 4.8	Q 4.8	Q 4.8	i13	Q 8.8	Q 7.9	Q 3.13
i2	Q 5.7	Q 5.7	Q 5.7	i14	Q 8.8	Q 9.7	Q 6.10
i3	Q 6.6	Q 6.6	Q 6.6	i15	Q 8.8	Q 8.8	Q 8.8
i4	Q 2.8	Q 2.8	Q 2.8	i16	Q 2.14	Q 2.14	Q 2.14
i5	Q 3.7	Q 3.7	Q 2.8	i17	Q 1.15	Q 1.15	Q 1.15
i6	Q 4.12	Q 4.12	Q 4.12	i18	Q 2.14	Q 2.14	Q 1.15
i7	Q 6.10	Q 6.10	Q 6.10	i19	Q 4.12	Q 4.12	Q 3.13
i8	Q 5.11	Q 5.11	Q 5.11	i20	Q 4.6	Q 4.6	Q 4.6
i9	Q 6.10	Q 6.10	Q 6.10	i21	Q 1.9	Q 1.9	Q 1.9
i10	Q 2.14	Q 2.14	Q 2.14	i22	Q 4.12	Q 3.13	Q 2.14
i11	Q 3.13	Q 3.13	Q 2.14	i23	Q 4.12	Q 3.13	Q 2.14
i12	Q 6.10	Q 3.13	Q 1.15	i24	Q 4.12	Q 2.14	Q 1.15

constellation demapper. For a system with convolutional code, BICM and 8-PSK modulation, 1 and 1.5-dB gains for BER performance were reported for Rayleigh flat fading channels and channels with AWGN, respectively. In [32], the impact of different mapping styles on the performance of BICM with iterative demapping for Rayleigh fading channels have been investigated. Iterative demapping has provided significant coding gains for several mapping schemes of QAM constellations. In [33], only a small gain of 0.1 dB was observed when the convolutional code was replaced by a turbo code. This result makes iterative demapping with turbo-like coding solutions unsatisfactory even though the added complexity is relatively small. On the other hand, SSD technique was introduced in [20] to improve the performance gains. An improvement exceeding 0.8-dB gain is observed at BER lower than 10^{-7} at the price of a relatively small added

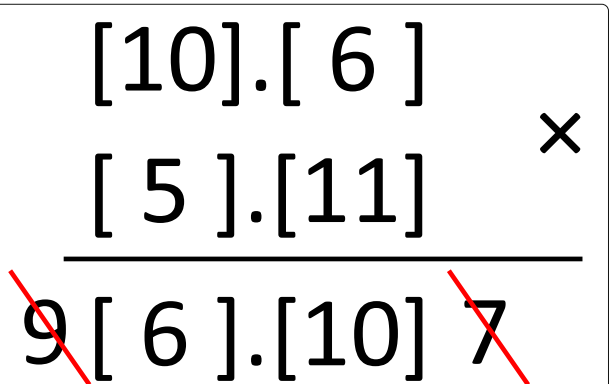

Fig. 6 Example for quantization of product value

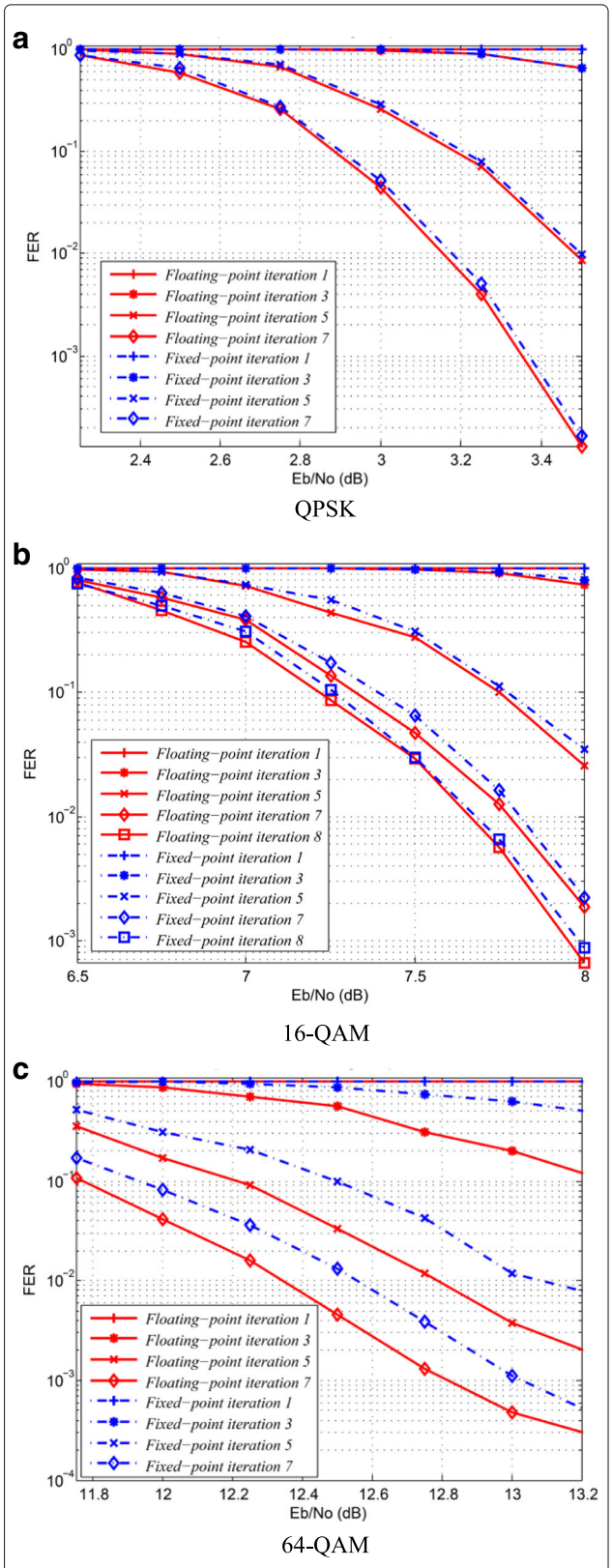

Fig. 7 a–**c** Floating-point vs. fixed-point FER performance comparison of turbo-equalization for 4× 4 MIMO, 1536 source bits, double binary turbo encoder, $\frac{1}{2}$ code rate, and fast fading Rayleigh channel

complexity without sacrificing the iterative process convergence. In [34], the use of iterative demapping shows performance improvement of 1.2 dB at BER of 10^{-6} for QAM BICM scheme with LDPC channel decoder over flat fading Rayleigh channel with 15% of erasures. The symbol-by-symbol maximum a posteriori (MAP) algorithm is the optimal algorithm for obtaining the outputs of the demapper. The MAP algorithm is likely to be considered of high complexity for hardware implementation in a real system basically because of the numerical representation of probabilities, non-linear functions, and mixed multiplications and additions of these values [27]. To avoid the number of complicated operations, certain simplifications are applied. Implementing the MAP algorithm in its logarithmic domain instead of probabilistic form reduces the computational complexity. Operating in logarithmic domain eliminates exponential operations and transforms multiplication/division operations into addition/subtraction operations. Max-Log-MAP demapping algorithm is a suboptimal direct transformation of the MAP algorithm into logarithmic domain; hence, values and operations are easier to handle.

4.1 Algorithmic overview

Depending on the transmitter configuration and propagation conditions, the input from the wireless channel can be either directly delivered to the demapper or passed through a channel equalizer as shown in Fig. 1. To reduce the computational complexity, the demapper works in logarithmic domain and produces probabilities $\tilde{\nu}$ on received sequence in the form LLRs, where ν represents the binary mapping of the transmitted sequence. The demapper computes the LLRs using the following expression [35]:

$$L(\tilde{\nu}_t^i) = ln\left[\frac{\sum_{x\in\mathcal{X}_1^i}\left(e^{-\frac{1}{2\sigma^2}|y_t-\rho_t.x|^2}\cdot\prod_{\substack{l=0\\l\neq i}}^{m-1}P\left(\hat{\nu}_t^l\right)\right)}{\sum_{x\in\mathcal{X}_0^i}\left(e^{-\frac{1}{2\sigma^2}|y_t-\rho_t.x|^2}\cdot\prod_{\substack{l=0\\l\neq i}}^{m-1}P\left(\hat{\nu}_t^l\right)\right)}\right] \tag{17}$$

where m is the number of bits per symbol, $i = 0, 1, \ldots, m-1$, $L\left(\tilde{\nu}_t^i\right)$ is the LLR of ith bit of transmitted symbol at time t, $\mathcal{X}_0^i$ and $\mathcal{X}_1^i$ are the symbol sets of constellation for which symbols have their ith bit equals $b \in \{0,1\}$, ρ_t is the channel fading coefficient and σ^2 is the AWGN variance, and $P\left(\hat{\nu}_t^l\right)$ is the probability of lth bit of symbol x computed through a priori information. To reduce the complexity, max-log approximation [27] is applied by using the following formulas:

$$\ln\frac{a}{b} = \ln(a) - \ln(b) \tag{18}$$

$$\ln\left(e^{\delta_1+\ldots+e^{\delta n}}\right) \approx \max_{i\in\{1,\ldots,n\}}\delta_i \tag{19}$$

$$\max(a) - \max(b) = \min(-b) - \min(-a) \tag{20}$$

The expression in (17) becomes:

$$L\left(\tilde{\nu}_t^i\right) \approx \min_{x\in\mathcal{X}_0^i}(D - Ap_i) - \min_{x\in\mathcal{X}_1^i}(D - Ap_i) \tag{21}$$

where

$$D = \frac{\left|y_t^I - \rho_t^I.x^I\right|^2 + \left|y_t^Q - \rho_t^Q.x^Q\right|^2}{2\sigma^2} \tag{22}$$

and

$$Ap_i = \sum_{\substack{l=0\\l\neq i \text{ and } \nu^l=1}}^{m-1} L\left(\hat{\nu}_t^l\right) \tag{23}$$

where ν^l is the lth bit of each received modulated symbol.

In the case of non-iterative demodulation, no a priori information is provided to the demapper. The expression of LLRs in (21) becomes:

$$L\left(\tilde{\nu}_t^i\right) \approx \min_{x\in\mathcal{X}_0^i}(D) - \min_{x\in\mathcal{X}_1^i}(D) \tag{24}$$

Moreover, for Gray mapped constellations, I and Q components are independent from each other; hence, the Euclidean distance is calculated in one dimension. In case where m is even, further simplification can be applied. The expression in (24) can be transformed in this case into the following expressions [36]:

$$L\left(\tilde{\nu}_t^i\right) \approx \min_{x\in\mathcal{X}(I)_0^i}\left(D^I\right) - \min_{x\in\mathcal{X}(I)_1^i}\left(D^I\right) \; for\; i = 0, 1, .., \frac{m}{2} - 1 \tag{25}$$

and

$$L\left(\tilde{\nu}_t^i\right) \approx \min_{x\in\mathcal{X}(Q)_0^j}(D^Q) - \min_{x\in\mathcal{X}(Q)_1^j}(D^Q) \; for\; j = \frac{m}{2}, .., m - 1 \tag{26}$$

where

$$D^I = \frac{\left|y_t^I - \rho_t^I.x^I\right|^2}{2\sigma^2}, D^Q = \frac{\left|y_t^Q - \rho_t^Q.x^Q\right|^2}{2\sigma^2}$$

and $\mathcal{X}(I)_b^i$ and $\mathcal{X}(Q)_b^j$ are the constellation point sets on I-axis and Q-axis with ith and jth bits of symbol x that have a value equals to b. Applying this simplification, $2^{\frac{m}{2}}$ one-dimensional Euclidean distances are computed instead of 2^m two-dimensional Euclidean distances for each LLR.

In case of passing the received symbols through SISO equalizer (Fig. 1), symbol y in expressions (21), (24),

(25), and (26) is replaced with $\tilde{x}$ (4). Also, the fading factor ρ and variance σ^2 in the up-mentioned expressions are replaced with g (10) and $g(1-g)\sigma_x^2$, respectively [29].

4.2 Max-Log-Map demapping algorithm towards implementation

The simplified expression in (21) exhibits four main computation steps:

1. Euclidean distance computation to find D
2. A priori LLRs summation to calculate Ap_i
3. Minimum operations referred by the min functions.
4. Subtraction operation of minimum values to determine $L\left(\tilde{v}_t^i\right)$ values

To determine output LLRs related to each received symbol y, computations of Euclidean distances and a priori LLR summation are repeated consecutively for all symbols of target constellation. Performing concurrently, these computations enhances the demapper execution performance. In a constellation with m bits per modulated symbol, 2^m Euclidean distances, and 2^m a priori LLR summation operations are needed. Their corresponding 2^m resultant differences are fed to m minimum finder operations to determine the minimum values relative to each bit. m subtractors are needed to determine the final LLR values. Thus, the complexity of the demapper implementation varies significantly with respect to m.

Recent emergent standards specify different mapping types starting from BPSK till 256-QAM as shown in Table 2. To meet with different standards specifications, the demapper supports the implementation of all required computations for variable modulation orders where m can range from 1 to 8. In general, the allocated hardware resources are not shared among different computational tasks (computing of Euclidean distance, a priori LLR summation, finding the minimum values, and subtraction operations to determine the final LLR values) to achieve the best execution performance. Furthermore, for operations depending on constellation size, sufficient resources are instantiated to suit the highest-order target constellation (256-QAM). A simple way to cope with modulation order variety is to store the constellation information (x^I, x^Q, and the binary mapping μ) in LUT, which contents are rewritten when system configuration changes. The size of the LUT, so-called *Constellation LUT*, varies according to the modulation order and mapping style. In fact, the depth of *Constellation LUT* equals the number of constellation points involved in determining the LLRs associated to one input symbol, whereas the width is constant and it is determined by the total number of bits representing the constellation information. Figure 8 shows the structure and organization of *Constellation LUT* for 16-QAM modulation scheme. The LUT in Fig. 8b contains the needed information of 16-QAM constellation presented in Fig. 8a when Gray mapping simplifications of expressions (25) and (26) are applied. The LUT in Fig. 8c represents the constellation information required when using the general Max-Log-MAP demapping algorithm expressed in (21).

Table 2 Supported modulation schemes in different standards

Standard	BPSK	QPSK	8-PSK	16APSK	32APSK	16-QAM	64-QAM	256-QAM
IEEE-802.16		✓				✓	✓	
IEEE-802.11	✓	✓				✓	✓	
LTE		✓				✓	✓	
LTE-Advanced		✓				✓	✓	
DVB-RCS		✓						
DVB-RCS2	✓	✓	✓			✓		
DVB-SH		✓	✓	✓				
DVB-S		✓						
DVB-S2		✓	✓	✓	✓			
DVB-T		✓				✓	✓	
DVB-T2		✓				✓	✓	✓

Furthermore, while exploring expressions (21), (25), and (26), one can notice that they share common arithmetic operations in computation of one-dimensional or two-dimensional Euclidean distances. In fact, computing of one two-dimensional distance is equivalent to compute two separate one-dimensional distances. Hence, same hardware resources can be used for different mapping styles. Figure 9 shows the operators used in Euclidean distance computation while targeting the highest parallelism. A separate operator is allocated for each required operation and the inversion operation is achieved using $\frac{1}{2x}$LUT instead of undergoing expensive ncomputations.

Moreover, supporting iterative demapping requires the implementation of operators that perform a priori LLR summation. To accommodate all target constellations, hardware implementation is set to meet with the requirements of highest-order target constellation (256-QAM). Figure 10a shows the operators used in the summation of a priori LLRs in case of 256-QAM modulation scheme, whereas Fig. 10b shows the eight subtraction operators used to realize D_i values expressed in the following equation:

$$D_i = D - Ap_i \tag{27}$$

where D_i represents the subtraction of summation of a priori LLRs corresponding to bit v^i from the computed Euclidean distance. In case of lower-order mod-

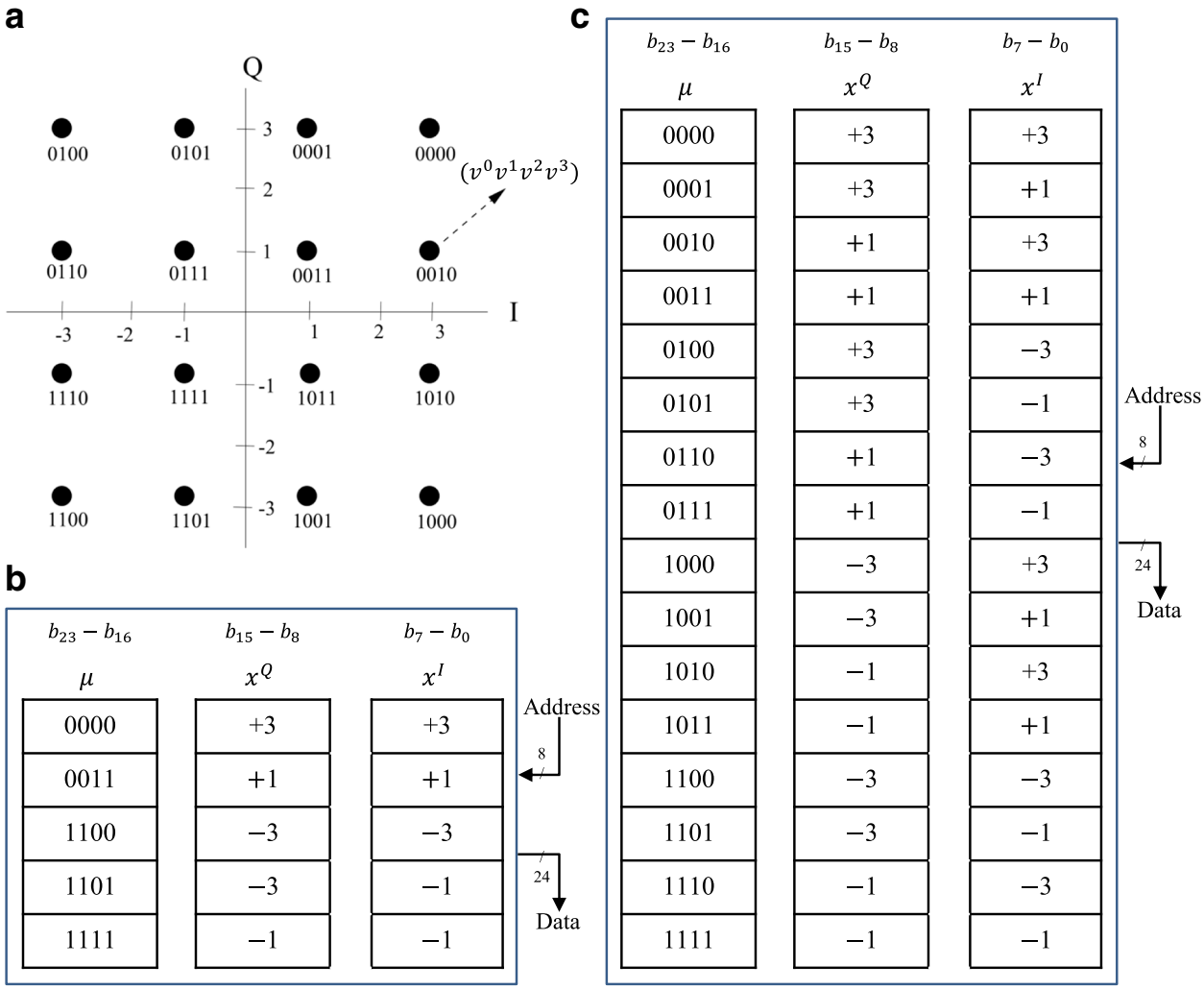

Fig. 8 *Constellation LUT* for 16-QAM modulation scheme. **a** 16-QAM constellation. **b** LUT contents for Gray mapped simplification. **c** LUT contents with no simplifications

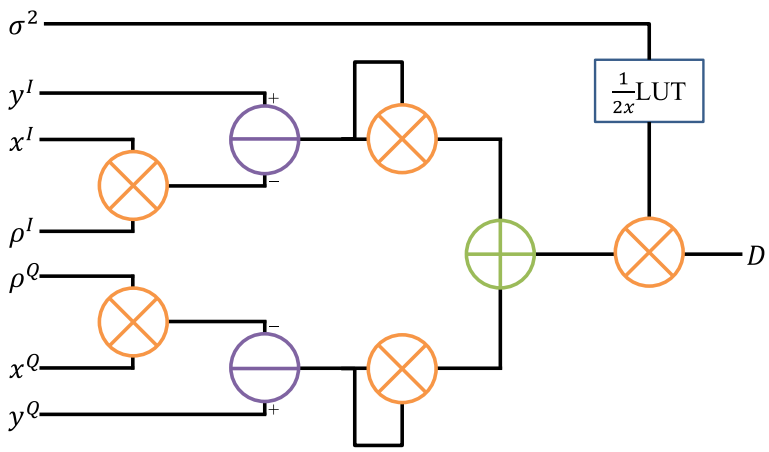

Fig. 9 Operators used in Euclidean distance computation

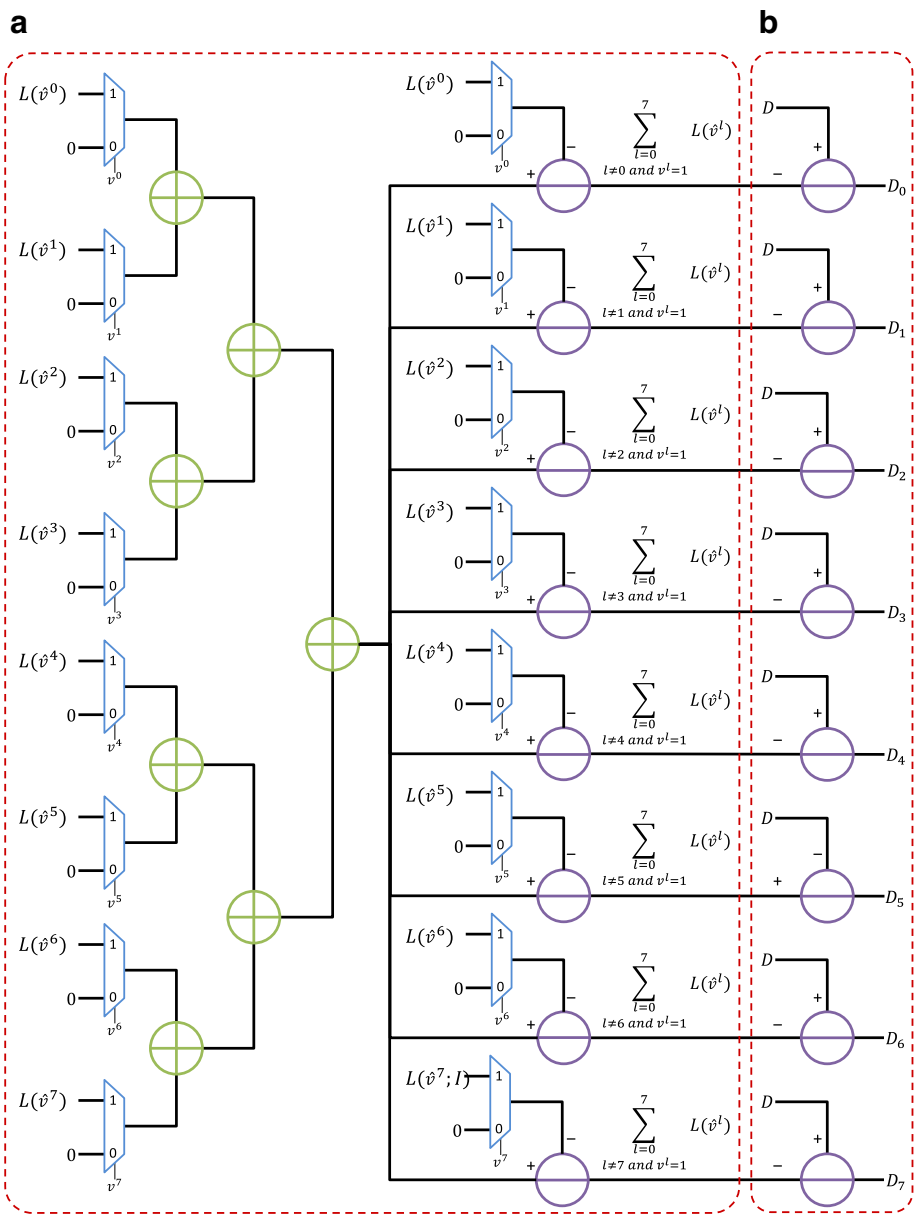

Fig. 10 Operators used in **a** a priori LLR summation and **b** D_i realization in case of 256-QAM modulation scheme

ulation schemes, the usage rate of hardware resources involved in a priori LLR summation decreases. For example, in case of 16-QAM modulation scheme only half of the hardware resources related to this operation will be activated.

Similarly to a priori LLR summation, the requirements of 256-QAM modulation scheme is adopted to implement hardware resources capable to perform minimum operations referred by the min functions and the subtraction operation of minimum values to determine $L\left(\tilde{v}_t^i\right)$ values expressed in (21). Sharing resources for different minimum operations leads to decreased throughput especially for high-order modulation schemes. Figure 11 presents the Minimum Finder operational unit for one LLR corresponding to bit v^i. Updating the minimum value depends on the value of v^i and the sign S of the resultant value of subtracting available minimum value from new D_i. The sign S represents the most significant bit (MSB) value of the difference.

$$S = \mathrm{MSB}\left(D_i - \min_{x_t \in \mathcal{X}_{0or1}^i} (D_i)\right) \tag{28}$$

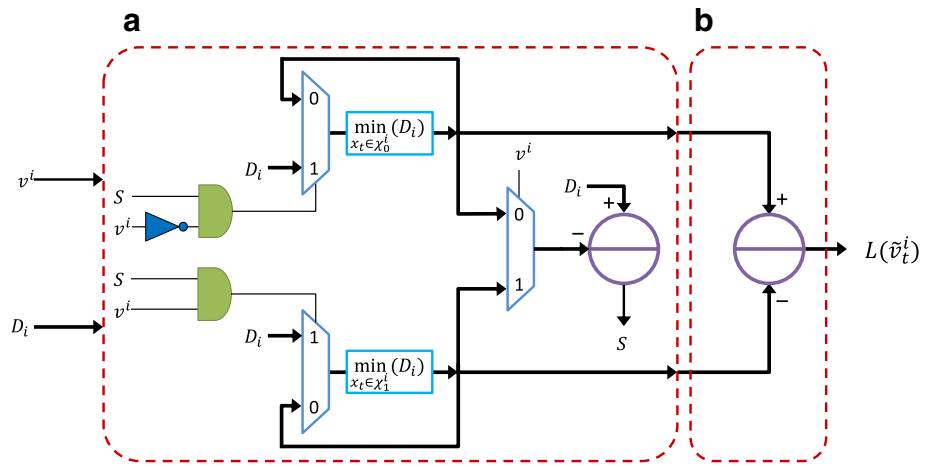

Fig. 11 a Minimum Finder operational unit and **b** subtractor used in subtraction operation of minimum pair

In addition, the figure shows the subtractor operator required to perform the subtraction operation of minimum pairs corresponding to symbol sets $\mathcal{X}_0^i$ and $\mathcal{X}_1^i$.

4.3 Quantization and fixed-point arithmetic

As for the equalizer module, all computational values are quantized according to defined precisions. Detailed analysis and long numerical simulations have been conducted for different configurations to find the required data width and accurate precisions for fixed-point representation of all parameters involved in Max-Log-MAP demapping algorithm. As discussed in previous subsection, the demapper implementation does not adopt sharing of hardware resources among different computational operation types. Hardware components are considered to deal with the same type of data. Hence, quantized computational parameters may have different data widths. Accordingly, bit-widths of each computational parameter is carefully selected to ensure least performance degradation. A trade-off between performance and implementation costs has been conducted.

On the other hand, fixed-point representation is used by placing virtual decimal point in between two bit locations to separate the number of bits representing integer and fractional parts. The square operation in calculating the Euclidean distance (22) implies the definite positivity of resultant parameters. This criterion is exploited to classify computational parameters into signed and unsigned numbers. Two's complement representation is used to represent signed values, whereas unsigned numbers are represented in binary representation which is considered simpler and does not impose extra bits. Figure 12 represents the devised quantization of different parameters used in all computational operations of the Max-Log-MAP demapping algorithm. Table 3 shows the values of parameter quantization in fixed-point representation. The notation $Q[I].[F]$ is used where $[I]$ and $[F]$ designate the number of bits for integer and fractional parts, respectively. The prefixes "US" and "S" indicate whether the parameter is considered unsigned binary number or signed binary number represented in two's complement representation.

Furthermore, the operands of addition and subtraction operations are prior adjusted to the same fixed-point representation. Sign extending or zero padding (adding zeros to lower or upper bits) techniques are applied based on the quantization characteristics of parameters prior and post the adjustment. Before performing addition or subtraction operations, the operands are 1-b sign-extended to avoid underflow or overflow occurrence. The inversion operation of variance σ^2 is achieved using a LUT instead of undergoing expensive computations. The LUT contains 8-b positive values which are required to represent the inverse values $\left(\frac{1}{2\sigma^2}\right)$. At each index, the stored value represents the quantized $\frac{1}{2x}$ value of the index value x.

4.4 Performance evaluation

In order to evaluate the efficiency of the quantization parameters, the application performance is verified. Also, the computation accuracy due to adopted fixed-point arithmetic is evaluated. To measure the impact of quantization errors on the demapper performance, a methodology based on bit-true simulation of the fixed-point application has been utilized. For various system configurations, a software model implementing the devised

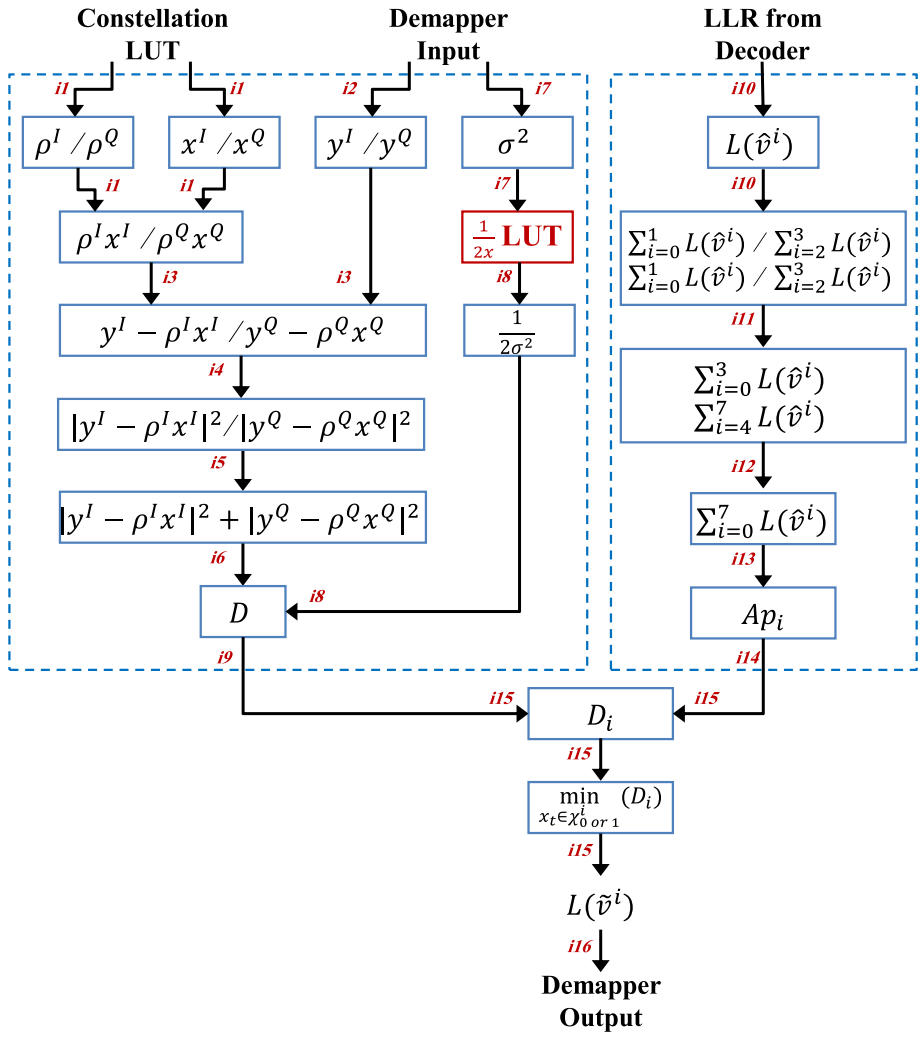

Fig. 12 Parameter quantization for Max-Log-MAP demapping algorithm

quantization and fixed-point specifications is used to simulate the demapper functionality. Accordingly, the corresponding frame error-rate (FER) performances of the receiver are recorded. Figure 13 presents the obtained FER curves compared to the reference floating-point curves. The analysis of the results has shown a performance loss below 0.05 dB for QPSK and 16-QAM at FER $= 10^{-2}$. Note that the FER values are recorded for 100 erroneous frames for each $\frac{E_b}{N_0}$ value. The obtained results shows a slight degradation in error-rate performance of the receiver. Thus, the effect of the quantization errors on the generated output LLR values is insignificant.

Table 3 Quantization parameters related to Fig. 12 in fixed-point representation

Index	Quantization	Index	Quantization
i1	S-Q 2.6	i9	US-Q 17.8
i2	S-Q 4.6	i10	S-Q 11.0
i3	S-Q 4.12	i11	S-Q 12.0
i4	S-Q 5.12	i12	S-Q 13.0
i5	US-Q 8.8	i13	S-Q 14.0
i6	US-Q 9.8	i14	S-Q 14.8
i7	US-Q 0.8	i15	S-Q 19.8
i8	US-Q 8.0	i16	S-Q 20.8

5 Conclusions

Fixed-point arithmetic and data quantization affect the performance of algorithmic implementation. In this paper, related issues to the fixed-point arithmetic of MMSE MIMO linear turbo-equalization and Max-Log-Map demapping are discussed for all algorithmic parameters and steps. An efficient quantization and fixed-point representation have been presented. Their impact is illustrated upon the FER performance for different system configurations. Only a slight degradation in the FER

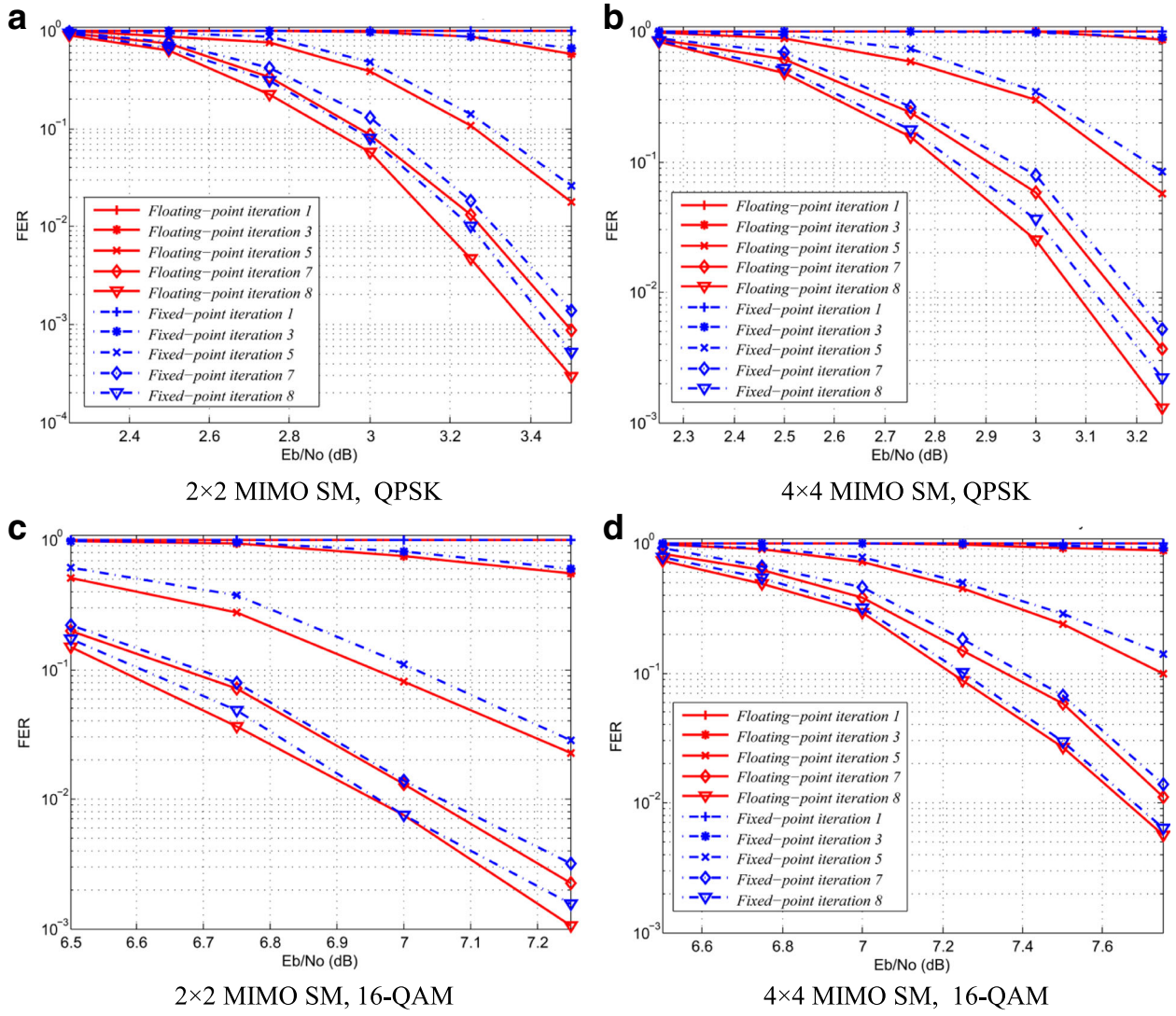

Fig. 13 Floating-point vs. fixed-point FER performance comparison turbo demodulation of 1536 source bits, $\frac{1}{2}$ code rate, and fast fading Rayleigh channel. **a** 2 × 2 MIMO SM, QPSK. **b** 4 × 4 MIMO SM, QPSK. **c** 2 × 2 MIMO SM, 16-QAM. **d** 4 × 4 MIMO SM, 16-QAM

performance of the receiver is observed when implementing the equalizer and demapper modules which utilize the devised quantization and fixed-point arithmetic rather than floating-point arithmetic.

Authors' contributions

MR investigated the algorithms and conducted the numerical study to determine the quantization and fixed-point representation of all parameters and computational values involved in the algorithms, evaluated the impact of devised quantization and fixed-point arithmetic on the error-rate performance, and wrote the paper. AB scientifically supervised the work, contributed in determining the quantization and fixed-point representation, and participated in writing the paper. MJ, YM, and YA academically supervised the work for IMT Atlantique and Lebanese University. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]Lebanese International University, School of Engineering, Mousaytbeh, Beirut, Lebanon. [2]Institut Mines-Telecom, IMT Atlantique, CNRS UMR 6285 Lab-STICC, Brest, France. [3]Faculty of Sciences, Lebanese University, Hadath, Lebanon.

Received: 23 November 2016 Accepted: 3 October 2017
Published online: 23 October 2017

References

1. D Menard, R Serizel, R Rocher, O Sentieys, Accuracy constraint determination in fixed-point system design. EURASIP J. Embed. Syst. **2008**(1) (2008). doi:10.1155/2008/242584
2. C Berrou, A Glavieux, P Thitimajshima, in *Proc. of IEEE International Conference on Communications (ICC)*. Near Shannon limit error-correcting coding and decoding: turbo-codes. 1, vol. 2 (IEEE, Geneva, 1993), pp. 1064–1070

3. X Li, J Ritcey, Bit-interleaved coded modulation with iterative decoding. IEEE Commun. Lett. **1**(6), 169–171 (1997). doi:10.1109/4234.649929
4. C Douillard, M Jezequel, C Berrou, A Picart, P Didier, A Glavieux, Iterative correction of inter symbol interference: turbo-equalization. Eur. Trans. Telecommun. (ETT). **6**, 507–511 (1995)
5. M Rizk, A Baghdadi, M Jezequel, Y Mohanna, Y Atat, Nisc-based soft-input soft-output demapper. IEEE Trans. Circ. Syst. II. **62**(11), 1098–1102 (2015). ISSN=1549-7747
6. M Rizk, A Baghdadi, M Jezequel, Y Mohanna, Y Atat, in *Proc. of the IEEE International Conference on Communications and Information Technology (ICCIT)*. Flexible and efficient architecture design for MIMO MMSE-IC linear turbo-equalization (IEEE, Beirut, 2013), pp. 340–344
7. T Matsumoto, in *Smart Antennas: State of the Art*, ed. by T Kaiser. Iterative (turbo) signal processing techniques for MIMO signal detection and equalization (Hindawi Publishing Corporation, New York, 2005). Chap. 7
8. S Yang, L Hanzo, Fifty years of MIMO detection: the road to large-scale MIMOs. IEEE Commun. Surv. Tutor. **17**(4), 1941–1988 (2015)
9. R Bidan, C Laot, D Leroux, in *Proc. of International Symposium on Turbo Codes and Related Topics*. Fixed-Point Implementation of an Efficient Low-Complexity Turbo-Equalization Scheme, (Brest, 2003), pp. 415-418. https://hal.archives-ouvertes.fr/hal-00917695
10. R Bidan, C Laot, D Leroux, in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*. Real-time MMSE turbo-equalization on the TMS320C5509 fixed-point DSP, vol. 5 (IEEE, Montreal, 2004), pp. V-325-8
11. M Schwall, D Leuck, FK Jondral, in *Proc. of IEEE 78th Vehicular Technology Conference (VTC Fall)*. Efficient fixed-point implementation of a SC-MMSE turbo equalizer (IEEE, Las Vegas, 2013), pp. 1–5
12. M Schwall, T Bose, FK Jondral, in *Proc. of 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. On the performance of SC-MMSE-FD equalization for fixed-point implementations (IEEE, Bremen, 2014), pp. 97–101
13. C Novak, P Fertl, G Matz, in *Proc. of IEEE International Symposium on Information Theory*. Quantization for soft-output demodulators in bit-interleaved coded modulation systems (IEEE, Seoul, 2009), pp. 1070–1074
14. S Haddad, A Baghdadi, M Jezequel, Complexity adaptive iterative receiver performing TBICM-ID-SSD. EURASIP J. Adv. Signal Process. **2012**(1), 131 (2012)
15. I Ali, U Wasenmüller, N Wehn, A high throughput architecture for a low complexity soft-output demapping algorithm. Adv. Radio Sci. **13**, 73–80 (2015)
16. C Douillard, M Jezequel, C Berrou, J Tousch, N Pham, N Brengarth, in *Proc. of the International Symposium on Turbo Codes and Related Topics (ISTC)*. The Turbo Code Standard for DVB-RCS, (Brest, 2000), pp. 535–538. https://hal.archives-ouvertes.fr/hal-00917695
17. C Berrou, C Douillard, M Jezequel, Multiple parallel concatenation of circular recursive systematic convolutional (CRSC) codes. Ann. Télécommun. **54**(3-4), 166–172 (1999)
18. E Zehavi, 8-PSK trellis codes for a Rayleigh channel. IEEE Trans. Commun. **40**(5), 873–884 (1992)
19. G Caire, G Taricco, E Biglieri, Bit-interleaved coded modulation. IEEE Trans. Inf. Theory. **44**(3), 927–946 (1998)
20. C Abdel Nour, C Douillard, in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*. On lowering the error floor of high order turbo BICM schemes over fading channels (IEEE, San Francisco, 2006), pp. 1–5
21. AF Molisch, *Wireless Communications*. (Wiley, United Kingdom, 2011)
22. E Biglieri, *Coding for Wireless Channels*. (Springer, USA, 2005)
23. D Gesbert, M Shafi, D-s Shiu, PJ Smith, A Naguib, From theory to practice: an overview of MIMO space-time coded wireless systems. IEEE J. Sel. Areas Commun. **21**(3), 281–302 (2003). ISSN=0733-8716
24. TL Marzetta, BM Hochwald, Capacity of a mobile multiple-antenna communication link in Rayleigh flat fading. IEEE Trans. Inf. Theory. **45**(1), 139–157 (1999). ISSN=0018-9448
25. E Biglieri, J Proakis, S Shamai, Fading channels: information-theoretic and communications aspects. IEEE Trans. Inf. Theory. **44**(6), 2619–2692 (1998)
26. L Bahl, J Cocke, F Jelinek, J Raviv, Optimal decoding of linear codes for minimizing symbol error rate(corresp.) IEEE Trans. Inf. Theory. **20**(2), 284–287 (1974)
27. P Robertson, P Hoeher, E Villebrun, Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding. Eur. Trans. Telecommun. (ETT). **8**(2), 119–125 (1997)
28. C Laot, R Le Bidan, D Leroux, Low-complexity MMSE turbo equalization: a possible solution for EDGE. IEEE Trans. Wirel. Commun. **4**(3), 965–974 (2005)
29. C Berrou, *Codes and Turbo Codes*. (Springer, Paris, 2010)
30. MT Gamba, G Masera, A Baghdadi, in *Proc. of the International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. Iterative MIMO Detection: Flexibility and Convergence Analysis of SISO List Sphere Decoding and Linear MMSE Detection (IEEE, Split, 2010), pp. 175–179
31. D Menard, P Quémerais, O Sentieys, in *XI European Signal Processing Conference (EUSIPCO 2002)*. Influence of fixed-point DSP architecture on computation accuracy (EURASIP, Toulouse, 2002)
32. A Chindapol, J Ritcey, Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels. IEEE J. Sel. Areas Commun. **19**(5), 944–957 (2001). ISSN=0733-8716. doi:10.1109/49.924878
33. I Abramovici, S Shamai, On turbo encoded BICM. Ann. Télécommun. **54**(3), 225–234 (1999)
34. C Abdel Nour, C Douillard, in *Proc. of the International Symposium on Turbo Codes and Related Topics (ISTC)*. Improving BICM performance of QAM constellations for broadcasting applications, (2008), pp. 55–60. doi:10.1109/TURBOCODING.2008.4658672
35. C Abdel Nour, Spectrally Efficient Coded Transmission for Wireless and Satellite Applications. PhD thesis, Elec. Dept., Telecom Bretagne, Brest, France (2008)
36. E Akay, E Ayanoglu, in *Proc. of the IEEE International Conference on Communications (ICC)*. Low complexity decoding of bit-interleaved coded modulation for M-ary QAM, vol. 2 (IEEE, Paris, 2004), pp. 901–905