On the Implementation of a Probabilistic Equalizer for Low-Cost Impulse Radio UWB in High Data Rate Transmission

Sami MEKKI¹, Jean-Luc DANGER¹, Benoit MISCOPEIN²

¹Institut Telecom/Telecom Paris Tech (ENST), Paris, France
²France Télécom R & D, Meylan Cédex, France

Email: {mekki, danger}@enst.fr, benoit.miscopein@orange-ftgroup.com

Received April 11, 2009; revised June 9, 2009; accepted June 10, 2009

Abstract

This paper treats the digital design of a probabilistic energy equalizer for impulse radio (IR) UWB receiver in high data rate (100 Mbps). The aim of this study is to bypass certain complex mathematical function as a chi-squared distribution and reduce the computational complexity of the equalizer for a low cost hardware implementation. As in Sub-MAP algorithm, the max* operation is investigated for complexity reduction and tested by computer simulation with fixed point data types under 802.15.3a channel models. The obtained results prove that the complexity reduction involves a very slight algorithm deterioration and still meet the low-cost constraint of the implementation.

Keywords: Impulse Radio Ultra-Wideband, Probabilistic Energy Equalizer, Inter-Symbol Interference, Chi-Squared

1. Introduction

Ultra-wideband impulse radio is considered as a promising candidate for indoor communications and wireless sensor networks, as described in [1]. Despite the numerous advantages afforded by the ultra-wideband (UWB) [1], this system faces the technological limits which brake the development of impulse radio (IR) UWB. Coherent IR-UWB reception, based on Rake receiver is limited in number of implementable Rake fingers [2]. An alternative is given by the transmitter reference (TR) method [3], however the electronic architecture is more complex as it needs analog delay lines and mixers. Non-coherent energy detection receiver is far less complex as a few components like shottky diodes and capacitors suffice. Though, the energy detection is simple to implement, transmitting impulses at high data rate leads to inter-symbol interference (ISI) which decreases the performance of the receiver [4–6]. An efficient scheme is necessary to improve the system performance.

A probabilistic energy equalizer is proposed in [7], which handles different types of interference. Besides the ISI, the proposed equalizer could manage the intra-symbol interference, called also inter-slot interference (ISIt) in an $M$-array pulse position modulation. Nevertheless, equalization process is mathematically complex to implement. The problem is mainly located on the energy distribution which follows a chi-squared distribution [8] and on the number of multiplications required by the equalizer.

In this paper, the probabilistic equalizer defined in [7] is simplified by applying the Jacobi logarithm [9] where addition become max* operation (using Viterbi's notation [10]) and multiplications become additions. In order to make this possible, an approximation of the chi-squared distribution is considered and rewritten in the logarithmic domain as the probabilistic equalizer. The simplified equalizer is embedded into the iterative loop of a channel decoder which applies the Sub-MAP algorithm in the decoding process.

This article is organized as follows: Section 2 defines the system model under consideration, where energy distribution is established. Equalization principle is reviewed in Section 3. In Section 4, the energy distribution is approximated by a simple function for hardware implementation. Results with the approximated distribution are compared to the chi-squared distribution in Section 5. In the same Section, the hardware implementation results in fixed point precision data types are also depicted and compared to the theoretical results in floating point precision. In Section 6 we rewrite the probabilistic equalizer in the logarithmic domain to base 10 with respect to max* operation and to the approximated distribution. In Section 7, the complexity
and the performance of the logarithmic equalizer is studied and compared to the complexity of a linear equalizer. Finally, conclusion and forthcoming work in the field are given in Section 8.

2. System Design

We consider an IR-UWB receiver based on energy detection. Data transmission is ensured via the M-array pulse position modulation (M-PPM) over a bandwidth W. Transmitting pulses over a high dispersive channel causes inter-symbol interference (ISI) and intra-symbol interference denoted as inter-slot interference (ISI). The received signal over a time symbol \( T_i \) has the following expression

\[
y_n(t) = \sum_{k=0}^{\infty} x_{n-k}(t) + z_n(t) = \sum_{k=0}^{\infty} y_{n-k}(t) + z_n(t)
\]

where \( z_n(t) \) is an additive white Gaussian noise with variance \( \sigma^2 \) and mean zero, and \( x_{n-k}(t) \) is the channel response of the \( (n-k) \)th transmitted symbol defined by:

\[
x_{n-k}(t) = p(t - A_{n-k}T_{slot}) \otimes h(t)
\]

where \( h(t) \) is the impulse channel response, \( \otimes \) denotes the convolution product, \( p(t) \) is the pulse shape, \( T_{slot} \) is the time slot duration for an M-PPM modulation, i.e. \( T_i = MT_{slot} \), and \( A_n \) takes value in \{0,1,\ldots, M-1\} according to transmitted symbol.

Let \( K \) denotes the number of interfering symbol assumed by the receiver, even though the real number of interfering symbol is greater. Thus for digital treatment the received signal (1) becomes a finite sum defined as:

\[
y_n(t) = \sum_{k=0}^{K-1} y_{n-k}(t) + z_n(t)
\]

The received energy per time slot \( T_{slot} \) in the \( n \)th received symbol is given by

\[
E_{n,m} = \int_{T_{slot}}^{T_{slot}+(m)T_{slot}} (s_n(t) + z_n(t))^2 \, dt
\]

where \( s_n(t) = \sum_{k=0}^{K-1} x_{n-k}(t) \).

Following the approach of Urkowitz [11], it was shown that the energy of a signal of duration \( T_{slots} \) can be represented as a sum of \( 2T_{slot}W \) samples in number which is known as the degrees of freedom (DoF). Let \( 2L \) stands for the DoF during a time slot \( T_{slot} \). Thus, the energy in the \( m \)th slot of \( n \)th symbol is given by

\[
E_{n,m} = \sum_{l=0}^{2L} (s_{n,m}^l + z_{n,m}^l)^2
\]

where \( s_{n,m}^l \) and \( z_{n,m}^l \) are respectively the \( l \)th sample of \( s_n(t) \) and \( z_n(t) \) in \( m \)th slot of \( n \)th symbol.

Assuming \( \sum_{l=0}^{2L} (s_{n,m}^l) \neq 0 \), then the received energy follows a non-central chi-squared distribution

\[
p(E_{n,m} | B_{n,m}) = \frac{1}{2\sigma^2} \left( \frac{E_{n,m}}{B_{n,m}} \right)^{\frac{l+1}{2}} e^{-\frac{(E_{n,m}+B_{n,m})}{2\sigma^2}} I_{L-1}\left( \frac{B_{n,m}E_{n,m}}{\sigma^2} \right)
\]

with \( 2L \) DoF and noncentrality parameter defined as

\[
B_{n,m} = \sum_{l=0}^{2L} (s_{n,m}^l)^2
\]

The function \( I_{L-1}(u) \) is the \((L-1)\)th-order modified Bessel function of the first kind [8]. If the noncentrality parameter is equal to zero; i.e. \( B_{n,m} = 0 \); the received energy follows a central chi-squared distribution defined as

\[
p(E_{n,m} | 0) = \frac{1}{\sigma^2} e^{-\frac{E_{n,m}}{2\sigma^2}}
\]

where \( \Gamma(z) \) is the gamma function [8].

The energy distribution is studied in next sections and simplified for hardware implementation.

3. Energy Equalization Principle

To benefit from the iterative process of a communication system, we consider a probabilistic equalizer that can be embedded into the iterative loop of a channel decoder based on SISO (Soft-Input/Soft-Output) decoding.

Thus, the considered equalizer takes the accumulated energy per slot (i.e. \( E_{n,m} \)) and per symbol (i.e. \( E_n = (E_{n,1}, E_{n,2}, \ldots, E_{n,M}) \)) as reference, in order to retrieve the transmitted symbol \( x_n \). So the detector computes a conditioned probability \( p(E_n | x_n) \) regarding the interfering symbols on \( X_n \). It has been shown in [7] that the equalization is performed by computing

\[
p(E_n | x_n) = \sum_{x_{n-K}} \cdots \sum_{x_{n-K+1}} \left[ \prod_{m=1}^{M} p(E_{n,m} | B_{n,m}) \prod_{k=1}^{K-1} \pi(x_{n-k}) \right]
\]

where \( \pi(x_{n-k}) \) is the \( a \ priori \) probability provided by the SISO decoder and \( p(E_{n,m} | B_{n,m}) \) is defined in Section 2. It was also established that the set of all the
However the smaller the number of DoF, the larger the approximation error. Due to the large bandwidth in UWB-IR, the number of DoF could be big enough [16] to consider the Gaussian distribution as an approximation to the chi-squared density. For instance, if is around for and . According to the previous Remark, the Gaussian approximation has the same mean and variance as the non-central chi-squared distribution, i.e.,

\[
m_{x^2} = 2L\sigma^2 + B_{n,m}
\]

\[
\sigma_{x^2}^2 = 4L\sigma^4 + 4\sigma^2 B_{n,m}
\]

This can be extended to the central chi-squared distribution by considering .

An approximation for the chi-squared distribution is thus necessary. In the literature, there are some proposals for the calculation of the non-central chi-squared distribution [12] and the use of the normal approximation to the chi-squared distribution [13,14], but those approximations require high bit precision and are therefore too complex for digital design.

An intuitive approximation can be found by considering the Remark in [15] which stands that when a variable \( \Sigma \) is used to approximate a variable \( \Omega \), it is equivalent to match the mean and variance of \( \Sigma \) and \( \Omega \).

It is notably shown in [15], that a chi-squared distribution can be approximated by a Gaussian distribution.

\[
p(E_{n,m} | B_{n,m}) \approx p(E_{n,m} | B_{n,m}) = \frac{\exp - \left( \frac{(E_{n,m} - m_{x^2})^2}{2\sigma_{x^2}^2} \right)}{\sqrt{2\pi\sigma_{x^2}^2}}
\]

Figure 2(a) shows the error measured by

\[
| p(E_{n,m} | B_{n,m}) - p(E_{n,m} | B_{n,m}) | \text{ for } E_{n,m} \geq 0, B_{n,m} > 0 \text{ and } \sigma^2 = 1.
\]

| Table 1. Look-up table Input/Output size with \( x^2 \) distribution. |
|---|---|
| Parameters | Quantization size |
| \( E_{n,m} \) | 14 bits |
| \( B_{n,m} \) | 6 bits |
| \( \sigma^2 \) | 6 bits |
| \( p(E_{n,m} | B_{n,m}) \) | 7 bits |
| \( x^2 \) Table size | 448 Mbits (56 Mbytes) |

Copyright © 2009 SciRes.
It is noticed that the error tends to zero as \( \sigma^2 \) decreases (Figure 2 (b)). According to [7], the energy equalizer operates at \( \sigma^2 < 1 \); i.e. \( \sigma^2 = 1 \) corresponds to \( SNR = -3dB \) for a pulse energy equals to unity in coded system. So, the maximum error, considered between the chi-squared and Gaussian distributions, is \( \varepsilon = 5 \times 10^{-3} \) as shown in Figure 2(a). We denote the normal function by

\[
\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \tag{12}
\]

Using (9), (10) and (12), equation (11) can be rewritten as follows

\[
p(E_{n,m} | B_{n,m}) = \frac{1}{\sqrt{\sigma_x^2}} e^{-2E_{n,m} - m_x^2} \sqrt{\sigma_x^2} \tag{13}
\]

As the energy distribution is simply deduced from the normal function \( \phi(t) \), the digital implementation can only use two look-up tables. The first one contains the values of the normal function \( \phi(t), \forall t \geq 0 \). The second one contains the values of the ratio \( 1/\sqrt{x}, \forall x > 0 \). The input/output precision of the look-up tables will be analyzed in the simulation Section according to the hardware constraints.

5. Performance of the Approximated Linear Equalizer

In this section, computer simulations have been run to assess the performance of the linear energy equalizer with the approximated Gaussian distribution defined by (13).

The BER computation has been performed via simulations in both floating point precision and fixed point precision data types. In the first part of simulations, we compare the performance of the receiver with the Gaussian approximation (11) and with the exact calculation of the chi-squared distribution in floating point precision. Second part of simulations has been run in fixed point data types with the approximated distribution.

The block fading multipath channel is generated randomly according to IEEE 802.15.3a UWB channel models [18]. Channel estimation is out of the scope of this paper. The channel state information (CSI) is assumed perfectly known at the receiver side. Nevertheless, channel parameters can be approached by the mean of the expectation-maximization (EM) algorithm as studied in [19] or by a set of a specific training sequence.

5.1. Chi-Squared Versus Gaussian Approximation Simulations in Double Precision

We consider an UWB-IR system as defined in Figure 1. Transmission is ensured by a 4-PPM modulation at 100Mbit/s. Thus we get 2 bits per transmitted symbol. We have implemented a duo-binary turbo code as it is defined in the standards [20,21]. This channel coder is chosen because it is suited to QPSK (quadratic phase shift keying) and 4-PPM modulations. The encoded data, at the input of the encoder, are 864-bit long blocks. The turbo encoder rate is 1/2 and 10 iterations of the SISO decoder are performed at the receiver side. The equalizer is jointly implemented into the iterative loop of the decoder to benefit from the iterative process of the decoder. The efficiency of the energy equalizer will not
be treated in this paper, the reader should refer to [7] for more details concerning the equalizer performances.

The receiver assumes that there are only two interfering symbols, i.e., \( K = 2 \) and \( P = 5 \), but the real number of interfering symbols could be more. The CSI is assumed over \( P \) time slots duration and not otherwise. In our case, for a data rate of 100 Mbps, the time slot duration is 5\( \mu \)s, so the receiver has a perfect CSI only over 25\( \mu \)s. This duration is sufficient for channel models as CM1 and CM2, although their respective maximum excess delay are 80\( \mu \)s and 115\( \mu \)s as it is studied in [7]. However for highly dispersive channel as CM3 and CM4 with maximum excess delay of 140\( \mu \)s and 200\( \mu \)s respectively, channel knowledge should be extended to \( K = 3 \). Nevertheless, we consider only simulations with \( K = 2 \) for the Gaussian approximation performance comparison.

It is noticed that the results with Gaussian approximation match the chi-squared performances in floating point precision even for highly dispersive channel such CM3 and CM4 with a slight degradation of performance.

**5.2. Fixed Point Precision Simulations**

The fixed point precision is subject to hardware constraints. The duo-binary turbo coder hardware implementation is out of the scope of this paper. The digital design of the channel coder is furnished by Turbo-Concept for an optimum efficiency [22]. The energy detector of UWB platform is a logarithmic one [23]. To guarantee the scalar value of the energy \( E_{n,m} \) for equalization, a look-up table of the function \( 10^x \) is required. Computer simulations in fixed point precision are achieved by means of the SystemC class sc_fix [24]. The Gaussian approximation for energy equalization are computed through the look-up tables of the functions:

\[
\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}, \quad g(x) = \frac{1}{\sqrt{x}} \quad \text{and} \quad h(x) = 10^x .
\]

Figure 4 shows the Gaussian approximation computation architecture for the chi-squared distribution.

According to the class sc_fix of SystemC, a signed or an unsigned object are defined by two parameters: the total word length noted as \( w_1 \), i.e. the total number of bits used in the type, and the integer word length noted as \( iw_1 \), i.e. the number of bits that are on the left of the binary point (.) in a fixed point number. The remaining bits stand for the fractional part of the object. Hence each object is represented by a pair of parameters noted \(< w_1, iw_1 >\).

Simulations have been carried out with different parameter sizes. Table 2 shows the word sizes of the parameters considered for the fixed point simulations. Reducing the number of bits for each variable involves a significant performance decrease.

![Figure 3. Chi-squared vs gaussian approximation in float precision using duo-binary turbo code at rate 1/2 with K=2.](image)
Table 3 lists the Input/Output size look-up table necessary for density computation.

**Table 2. Parameters size definition.**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Quantization size $&lt;\omega_i,\omega_f&gt;$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log E_{\omega}$</td>
<td>$&lt;6,2&gt;$</td>
</tr>
<tr>
<td>$E_{\omega}$</td>
<td>$&lt;14,1&gt;$</td>
</tr>
<tr>
<td>$B_{\omega}$</td>
<td>$&lt;6,2&gt;$</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>$&lt;6,1&gt;$</td>
</tr>
<tr>
<td>$m^2_{\omega}$</td>
<td>$&lt;12,2&gt;$</td>
</tr>
<tr>
<td>$\sigma^2_{\omega}$</td>
<td>$&lt;12,2&gt;$</td>
</tr>
<tr>
<td>$p\left(E_{\omega} ,</td>
<td>, B_{\omega}\right)$</td>
</tr>
<tr>
<td>$p\left(E_{\omega} ,</td>
<td>, x_{\omega}\right)$</td>
</tr>
<tr>
<td>$\pi(x_{\omega})$</td>
<td>$&lt;4,1&gt;$</td>
</tr>
</tbody>
</table>

We notice that the total memory occupied by the look-up tables is lower than the $\chi^2$ look-up table as it is described in Table 1.

Simulations according to Table 2 and 3 under the same conditions as for double precision lead to the results depicted in Figure 5.

**Table 3. Look-up table input/output size.**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Input size</th>
<th>Output size</th>
<th>Table size (Kbits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$</td>
<td>$&lt;8,2&gt;$</td>
<td>$&lt;18,0&gt;$</td>
<td>4.5</td>
</tr>
<tr>
<td>$g(x) = 1/\sqrt{x}$</td>
<td>$&lt;12,2&gt;$</td>
<td>$&lt;6,4&gt;$</td>
<td>24</td>
</tr>
<tr>
<td>$h(x) = 10^x$</td>
<td>$&lt;6,2&gt;$</td>
<td>$&lt;14,1&gt;$</td>
<td>0.875</td>
</tr>
</tbody>
</table>

Figure 4. Approximated energy distribution architecture for the linear equalizer with $x^2$ approximation.

Figure 5. $\chi^2$ float precision versus the Gaussian approximation in fixed point precision for $K=2$. 

Copyright © 2009 SciRes.
Results in fixed point precision data types are close to those obtained in double precision with chi-squared distribution. We notice, that even the quantization error of the energy distribution is around $1/2^{33}$, which is lower than the considered maximum error $e = 5 \times 10^{-3}$, the receiver performances are slightly degraded compared to double precision simulations.

5.3. Complexity of the Linear Equalizer

The linear equalizer with a chi-squared distribution has an expensive lookup table (Table 1). Due to the size of the ROM and its cost, the chi-squared distribution is approximated by a simple implementable function (13) with some memories.

According to Figure 4 and equation (8), the amount of non trivial multiplications in the linear domain is about $(4M + K - 2)M^K$ in a symbol period. We denote by trivial multiplication the multiplication by a power of 2 which is equivalent to a shift in hardware implementation. The detail of multiplications is as follows\(^1\)

$$
M_{\text{lin}} = \sum_{n=1}^{M} \sum_{x_{n}=1}^{K} \prod_{m=1}^{M} p(E_{n,m} | B_{n,m}) \times \prod_{k=1}^{2} \pi(x_{n,k})
$$

(14)

According to the decomposition in (14), $p(E_n | x_n)$ requires $(M + K - 2)M^{K-1}$ multiplication. We should notice that $p(E_n | x_n)$ is computed $M$ times per symbol duration. Thus, in a symbol period, the total amount of multiplication is equal to $(M + K - 2)M^K$ multiplications.

Once the number of multiplications is established for (14) per symbol period, we divide up each terms. The a priori probability, i.e. $\pi(x_{n,k})$, will not be discussed since it is provided by the SISO decoder. So our analysis is focused rather on the energy distribution. In the term $\prod_{m=1}^{M} p(E_{n,m} | B_{n,m})$, we calculate $M$ times $p(E_{n,m} | B_{n,m})$.

With respect to the architecture in Figure 4, $p(E_{n,m} | B_{n,m})$ requires 3 non-trivial multiplication. Hence, per symbol period there is $3M^{K-1}$ additional multiplications on the back of $(M + K - 2)M^K$ This leads to $(4M + K - 2)M^K$ multiplications calculated by the equalizer per time symbol.

As example, we consider a 4-PPM modulation at 100Mbps and $K = 2$ at the receiver side, this leads to 12.8Gmultiplication/s and a total memory of 30.25Kbits according to Table 3 for the energy distribution computing, i.e. $p(E_{n,m} | B_{n,m})$. We should notice that the equalizer computes the energy distribution $M^{K-1}$ times per time symbol. So, at 100Mbps for $K = 2$ with a 4-PPM the frequency of table access is about 3.2GHz. If we consider a hardware that runs at 400MHz, the level of parallelism to achieve the energy distribution is equal to 8. Thus, the total amount of memory is a factor of 8, i.e. 30.25Kbits $\times 8 = 242Kbits$.

The next par of this paper will be focused on complexity reduction of the probabilistic equalizer by the mean of Sub-MAP algorithm known also as Jacobi algorithm [9].

6. Complexity Reduction of the Probabilistic Energy Equalizer

Even though the defined equalizer is implementable in hardware devices, the expensive area required by the equalizer could be decreased by simple computational method. This is made possible by computing in the logarithmic domain where only additions and comparisons operations with small memories are required. The channel decoder should operate also in the logarithmic domain, in order to get the better performance of the receiver. The decoder properties are not discussed in this paper. However, Sub-MAP algorithm, also called Max-Log-MAP or Dual Viterbi [21,10] based decoder is a good candidate for joint decoder equalizer receiver in the logarithmic domain.

As in the Sub-MAP decoding [10,25], we consider the $max^*$ function which operates in a logarithm to base 10 defined as

$$
max^*(a, b) = \log(10^a + 10^b)
$$

(15)

the $max^*$ operation is essentially a max operation adjusted by a correction factor carried out by a lookup table, i.e a read-only memory (ROM); which outputs the correction term $\log(1 + 10^{-a-b})$ given the input $(a-b)$ in hardware implementation. As the $max^*$ property, $max^*$ is an associative operator (see APPENDIX 1 for the proof):

$$
max^*_{10}(a, b, c) = max^*_{10}[max^*_{10}(a, b), c]
$$

(16)

For notation simplicity we consider

$$
max^*_{10}(a_1, a_2, \ldots, a_N) = max^*_{10}(i=1,\ldots,N)(a_i)
$$

(17)

Using the $max^*$operation defined in (15), the output of the probabilistic equalizer in the logarithmic domain is
Considering the result of equation (18), it is noticed that the multiplication operations are replaced by comparators and adders which are costless and easy to implement. Since the equalizer output should be a probability, i.e. \( p(E_n | x_n) \in [0,1] \), a normalization process is applied as follow:

\[
\widehat{p}(E_n | x_n) = \frac{p(E_n | x_n)}{\sum_{y_n} p(E_n | x_n)}
\]

One should notice that with normalization process at the output of the equalizer, the redundant constants are removed. In addition, due to hardware restraint, the energy detector is a logarithm to base 10 detector as in [23] which provides logarithmic energies per time slot \( T_{slo} \). So, the only available data is \( \log E_{n,m} \), \( \log B_{n,m} \) and \( \log(2L\sigma^2) \). Expanding \( \sigma^2 \) in (22), we get

\[
\log p(E_{n,m} | B_{n,m}) \propto -\frac{1}{2} \log[2\sigma^2(2L\sigma^2 + 2B_{n,m})] - \frac{L(E_{n,m} - m_{x^2})^2}{4L\sigma^2(2L\sigma^2 + 2B_{n,m}) \ln 10}
\]

\[
\propto -\frac{1}{2} \log(2\sigma^2) - \frac{1}{2} \log(2L\sigma^2 + 2B_{n,m}) - \frac{L(E_{n,m} - m_{x^2})^2}{2.10 \log(2L\sigma^2(2L\sigma^2 + 2B_{n,m})) \ln 10}
\]

where the symbol \( \propto \) means “proportional to” and the function \( \log \) stands for the logarithmic to base 10. Rewriting (24) taking into account \( \max \) and removing the redundant constant such as \( -\frac{\log(2\sigma^2)}{2} \), leads to

\[
\log p(E_{n,m} | B_{n,m}) \propto -\frac{1}{2} \max_{y_n} \{ \log[2(2L\sigma^2)\log 2 + \log B_{n,m}] \}
\]

\[
- \frac{L(E_{n,m} - m_{x^2})^2}{2.10 \log(2L\sigma^2(2L\sigma^2 + 2B_{n,m})) \ln 10}
\]

the division part in (25) can be transformed into multiplications as follows:

\[
\log p(E_{n,m} | B_{n,m}) \propto -\frac{1}{2} \max_{y_n} \{ \log[2(2L\sigma^2)\log 2 + \log B_{n,m}] \}
\]

\[
- \frac{L(E_{n,m} - m_{x^2})^2}{2 \ln 10} \times 10^{-\gamma}
\]

where \( \gamma \) is defined as

\[
\gamma = \log(2L\sigma^2) + \max_{y_n} \{ \log[2(2L\sigma^2)\log 2 + \log B_{n,m}] \}
\]

and \( E_{n,m} - m_{x^2} \) is easily calculated as follows

\[
E_{n,m} - m_{x^2} = 10^{\log E_{n,m} - \left( \log(2L\sigma^2) + 10 \log B_{n,m} \right)}
\]

It is noticed that the energy distribution in logarithmic domain is achieved by the mean of two lookup tables. A first ROM for the \( \max \) and a second ROM for \( 10^\gamma \) function. The memories size will be treated in the simulation Section.

The advantage of working in logarithmic domain is that the amount of multiplications is confined only on the energy distribution. Figure 6 depicts the new energy distribution architecture implemented in digital design. This architecture is used for the probabilistic equalizer complexity study.
7. Performance of the Logarithmic Equalizer with the Approximated Distribution

7.1. Fixed Point DataTypes Simulation in Logarithm Domain

The performances of the logarithmic equalizer with the approximated distribution are simulated in fixed point precision and compared to the chi-squared distribution in double precision. Reception is ensured by a logarithmic energy detector [23]. Simulations in fixed point precision are carried out by the mean of the class sc_fix of SystemC as in 5.2. Table 4 shows the word sizes of parameters considered for the fixed point simulations.

With respect to the equalizer expression (18) and to the approximated energy distribution in logarithmic domain (26), we consider two ROM types whose sizes are defined in Table 5.

Figure 7 shows the results obtained if the receiver assumes that there are only 2 interfering symbols, i.e. CSI is known only over $P=5$ slots, however the real number of interfering symbols could be more.

We notice that for less dispersive channel such as CM1 and CM2, the results in fixed point precision data types are close to those obtained in double precision with chi-squared distribution. Regarding the results for highly dispersive channel (CM3 and CM4), we get a loss of 1dB at $BER=10^{-4}$. According to [7], the receiver could be improved if the supposed number of interfering symbols are bigger than 2, especially in highly dispersive channel. It has been proven that for channel models CM3 and CM4, the optimal compromise is to consider $K=3$ [7].

Simulations run with $K=3$ for CM3 and CM4 in fixed point data types are depicted in Figure 8. Although the complexity is slightly increased due to cardinal of the set $\{B_{n,m}\}$, i.e. $|\{B_{n,m}\}|=88$ for $K=3$ [7], the receiver is improved of 2dB for CM3 at $BER=10^{-4}$.

Table 4. Parameters size definition.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Quantization size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log F_{n,m}$</td>
<td>&lt;6,2&gt;</td>
</tr>
<tr>
<td>$\log B_{n,m}$</td>
<td>&lt;8,2&gt;</td>
</tr>
<tr>
<td>$\log (2L \sigma^2)$</td>
<td>&lt;7,4&gt;</td>
</tr>
<tr>
<td>$m_f^2$</td>
<td>&lt;8,4&gt;</td>
</tr>
<tr>
<td>$\log \mathbb{P}(E_{n,m}</td>
<td>B_{n,m})$</td>
</tr>
<tr>
<td>$\log \mathbb{P}(E_{n,m}</td>
<td>x_n)$</td>
</tr>
<tr>
<td>$\log \sigma(x_n)$</td>
<td>&lt;6,4&gt;</td>
</tr>
</tbody>
</table>

Table 5. ROM input/output size.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Input size</th>
<th>Output size</th>
<th>Table size (Kbits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g(x) = \log(1+10^{-d/2})$</td>
<td>&lt;6,2&gt;</td>
<td>&lt;4,0&gt;</td>
<td>0.25</td>
</tr>
<tr>
<td>$h(x) = 10^x$</td>
<td>&lt;9,5&gt;</td>
<td>&lt;8,3&gt;</td>
<td>4</td>
</tr>
</tbody>
</table>
7.2. Complexity of the Logarithmic Equalizer

According to the equalizer expression (18) and Figure 9, the computational complexity of the equalizer in terms of non-trivial multiplication is equal to $2M^{K+1}$ multiplication per symbol. Thus, with 4-PPM modulation at 100Mbps and $K = 2$, we get $64G\text{multiplication/s}$. Regarding the memory size, the logarithmic approximati-
on requires 16.25Kbits. Comparing to the linear equalizer the complexity in the logarithm domain, i.e. number of multiplications and the memory size, is promising for a low cost hardware implementation. For instance, if the hardware runs at 400Mhz and the frequency of table access is 3.2Ghz for the same example quoted before, we require 8 level of parallelism to achieve the energy distribution. So the total required memory is a factor of 8; i.e. 16.25Kbits × 8 = 130Kbits) which is lower than the required memory in linear domain (242Kbits). Moreover, the required bits for each parameters in Table 4 are shorter than in Table 2.

7.3. Complexity Summary

Table 6 is the synthesize of the complexity requirement for the linear and the logarithmic equalizer with the chi-squared distribution approximation. We notice that the logarithmic equalizer with the approximated energy distribution is far the less complex for hardware implementation, since it allows a compromise between the number of multiplications and the size of the required memory for equalization calculation.

8. Conclusion

In this paper, we have shown how a complex and costly probabilistic equalizer is simplified for digital design by using the logarithmic domain. A first simplification concerns the energy distribution which is approximated by a Gaussian distribution instead of a chi-squared. This leads to reduce significantly the required memory for distribution computation. The second simplification is to calculate all the probabilities in the logarithmic domain by the mean of the $max_{10}$ operation. Hence, the computational complexity of the equalizer is highly reduced compared to the linear equalizer. Moreover, only two lookup table types are required for equalizer calculation in logarithmic domain. Computer simulations demonstrated the performance of the receiver in finite precision. It showed, that for highly dispersive channels such as CM3 and CM4, the receiver is still able to equalize and decode the transmitted informations with a slight increase in complexity.

As perspective, some operations or memories could even be simplified or reduced by the mean of polynomial approximations with a negligible loss on the receiver performance. This could be a subject of investigation for future research.

9. References


From the definition of $\max_{10}^*$ we have
\[
\max_{10}^*(a, b, c) = \log(10^a + 10^b + 10^c)
\] (30)