***Wireless Sensor Network*, 2009, 1, 245-256**
doi:10.4236/wsn.2009. 14031 Published Online November 2009 (http://www.scirp.org/journal/wsn).

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

Sami MEKKI[1], Jean-Luc DANGER[1], Benoit MISCOPEIN[2]
[1]*Institut Telecom/Telecom Paris Tech (ENST), Paris, France*
[2]*France 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 (IStI) 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 $\max_{10}^*$ 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{TC “1 Transmitter and Receiver Design.”\f f}

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 (IStI). The received signal over a time symbol T_s has the following expression

$$y_n(t) = \sum_{k=0}^{\infty} x_{n-k}(t) + z_n(t) \tag{1}$$

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

$$x_{n-k}(t) = p(t - A_{n-k}T_{slot}) \otimes h(t) \tag{2}$$

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.* $T_s = MT_{slot}$, and $A_{n,k}$ 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:

$$y_n(t) = \sum_{k=0}^{K-1} x_{n-k}(t) + z_n(t) \tag{3}$$

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

$$\mathrm{E}_{n,m} = \int_{nT_S+(m-1)T_{slot}}^{nT_S+(m)T_{slot}} \left(s_n(t) + z_n(t)\right)^2 dt \tag{4}$$

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 know 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

$$\mathrm{E}_{n,m} = \sum_{\ell=1}^{2L} (s_{n,m}^{\ell} + z_{n,m}^{\ell})^2 \tag{5}$$

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

Assuming $\sum_{\ell=1}^{2L}(s_{n,m}^{\ell})^2 \neq 0$, then the received energy follows a non-central *chi-squared* distribution

$$p(\mathrm{E}_{n,m} \mid B_{n,m}) = \frac{1}{2\sigma^2}\left(\frac{\mathrm{E}_{n,m}}{B_{n,m}}\right)^{\frac{L-1}{2}} e^{-\frac{(\mathrm{E}_{n,m}+B_{n,m})}{2\sigma^2}} I_{L-1}\left(\frac{\sqrt{B_{n,m}\mathrm{E}_{n,m}}}{\sigma^2}\right) \tag{6}$$

with $2L$ DoF and noncentrality parameter defined as $B_{n,m} \overset{\Delta}{=} \sum_{\ell=1}^{2L}(s_{n,m}^{\ell})^2$. 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

$$p(\mathrm{E}_{n,m} \mid 0) = \frac{1}{\sigma^{2L}2^L\Gamma(L)} \mathrm{E}_{n,m}^{L-1} e^{\frac{-\mathrm{E}_{n,m}}{2\sigma^2}} \tag{7}$$

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. $\mathrm{E}_{n,m}$) and per symbol (i.e. $E_n = (\mathrm{E}_{n,1}, \mathrm{E}_{n,2}, \ldots, \mathrm{E}_{n,M})$) as reference, in order to retrieve the transmitted symbol x_n . So the detector computes a conditioned probability $p(E_n \mid x_n)$ regarding the interfering symbols on x_n . It has been shown in [7] that the equalization is performed by computing

$$p(E_n \mid x_n) = \sum_{x_{n-1}} \cdots \sum_{x_{n-K+1}} \left(\prod_{m=1}^{M} p(\mathrm{E}_{n,m} \mid B_{n,m}) \prod_{k=1}^{K-1} \pi(x_{n-k}) \right) \tag{8}$$

where $\pi(x_{n-k})$ is the *a priori* probability provided by the SISO decoder and $p(\mathrm{E}_{n,m} \mid B_{n,m})$ is defined in Section 2. It was also established that the set of all the possible values that $B_{n,m}$ could take, has a finite cardinal. Figure 1 summarize the transmission and the receiver design under consideration.

In order to reduce the complexity and make the equalizer feasible, we investigate the implementation in finite precision.

Moreover the probability given by Equation (8) needs some mathematical simplifications and approximations of the probability density function (pdf) $p(\mathrm{E}_{n,m} | B_{n,m})$, corresponding either to the *central* (7) or *non-central chi-squared* (6) distribution. This will be investigated in the following section.

4. Chi-Squared Distribution Approximation for Hardware Implementation

The chi-squared distribution defined by (7) and (6) is a three variable function ($\mathrm{E}_{n,m}$, $B_{n,m}$ and σ^2). Thus, building a look-up table according to these parameters would occupy a great memory. For instance, if the energy distribution is coded in 7 bits and $\mathrm{E}_{n,m}$, $B_{n,m}$ and σ^2 are coded respectively in 14 -bit, 6 -bit and 6 -bit long, the space memory allocated to this look-up table would occupy 448 Mbits (or 56 Mbytes). This corresponds to a costly silicon area in a FPGA or ASIC technology and thus incompatible with low-cost constraints.

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 Σ is used to approximate a variable Ω, it is equivalent to match the mean and variance of Σ and Ω.

It is notably shown in [15], that a *chi-squared* distribution can be approximated by a Gaussian distribution. However the smaller the number of DoF $2L$, the larger the approximation error. Due to the large bandwidth W in UWB-IR, the number DoF could be big enough [16] to consider the Gaussian distribution as an approximation to the *chi-squared* density. For instance $2L$ is around 30 for $W = 3GHz$ and $T_{slot} = 5ns$. According to the previous *Remark*, the Gaussian approximation has the same mean and variance as the *non-central chi-squared* distribution, *i.e.* $\mathrm{E}_{n,m} : N(m_{\chi^2}, \sigma^2_{\chi^2})$, given by [17]:

$$m_{\chi^2} = 2L\sigma^2 + B_{n,m} \tag{9}$$

$$\sigma^2_{\chi^2} = 4L\sigma^4 + 4\sigma^2 B_{n,m} \tag{10}$$

This can be extended to the *central chi-squared* distribution by considering $B_{n,m} = 0$.

Using these results and the aforementioned assumptions, we obtain the approximation for the energy distribution (noticed $\tilde{p}$) per slot, $\forall B_{n,m} \geq 0$ and $2L >> 2$ as

$$p(\mathrm{E}_{n,m} | B_{n,m}) \approx \tilde{p}(\mathrm{E}_{n,m} | B_{n,m}) = \frac{\exp\left(-\frac{(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2\sigma^2_{\chi^2}} \right)}{\sqrt{2\pi\sigma^2_{\chi^2}}} \tag{11}$$

Figure 2(a) shows the error measured by $| p(\mathrm{E}_{n,m} | B_{n,m}) - \tilde{p}(\mathrm{E}_{n,m} | B_{n,m}) |$ for $\mathrm{E}_{n,m} \geq 0$, $B_{n,m} > 0$ and $\sigma^2 = 1$.

Table 1. Look-up table Input/Output size with x^2 distribution.

Parameters	Quantization size
$\mathrm{E}_{n,m}$	14 bits
$B_{n,m}$	6 bits
σ^2	6 bits
$p(\mathrm{E}_{n,m}\|B_{n,m})$	7 bits
x^2 Table size	448 Mbits (56 Mbytes)

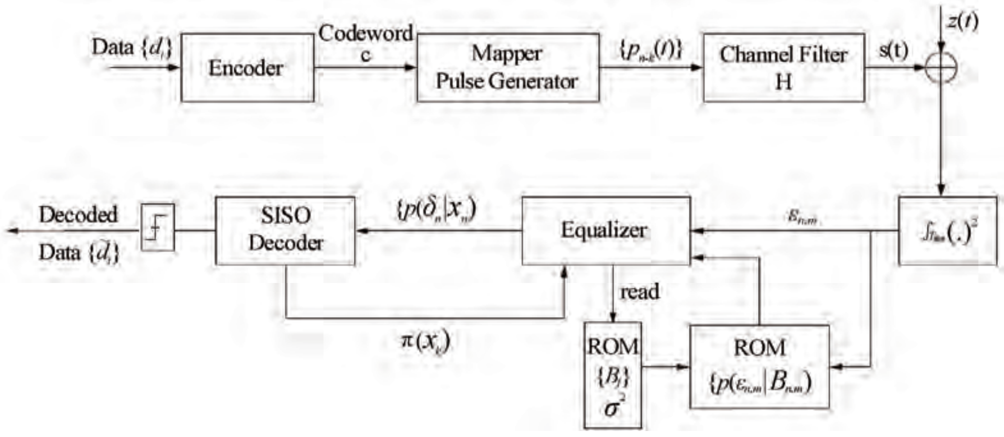
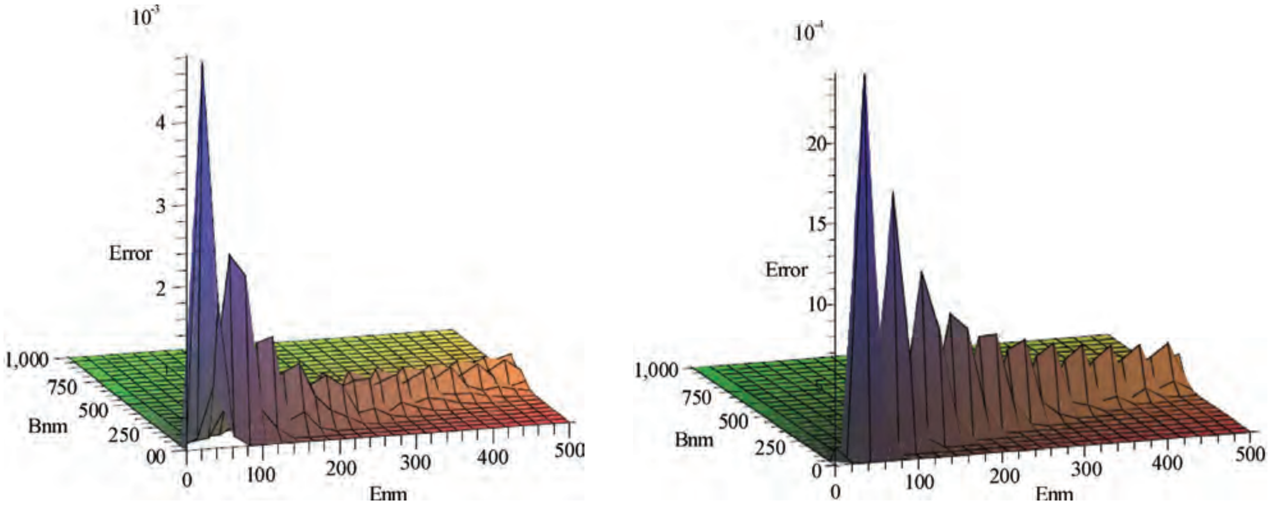


Figure 1. Transmitter and receiver design.



(a) Error for $\sigma^2 = 1$

(b) Error for $\sigma^2 = 0.5$

Figure 2. Error measured by $|p(E_{n,m} | B_{n,m}) - \tilde{p}(E_{n,m} | B_{n,m})|$ **for** $\forall E_{n,m} \geq 0$, $B_{n,m} > 0$.

It is noticed that the error tends to zero as σ^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

$$\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \qquad (12)$$

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

$$\tilde{p}\,(E_{n,m} | B_{n,m}) = \frac{1}{\sqrt{\sigma^2_{\chi^2}}} \phi\left(\frac{E_{n,m} - m_{\chi^2}}{\sqrt{\sigma^2_{\chi^2}}} \right) \qquad (13)$$

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 Gaussien 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 firsts 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 $100 Mbit/s$. 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 $5ns$, so the receiver has a perfect CSI only over $25ns$. This duration is sufficient for channel models as CM1 and CM2, although their respective maximum excess delay are $80ns$ and $115ns$ as it is studied in [7]. However for highly dispersive channel as CM3 and CM4 with maximum excess delay of $140ns$ and $200ns$ 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 $\mathrm{E}_{n,m}$ 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: $\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$, $g(x) = 1/\sqrt{x}$ and $h(x) = 10^x$. 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 wl, *i.e.* the total number of bits used in the type, and the integer word length noted as iwl, *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 $< wl, iwl >$.

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.

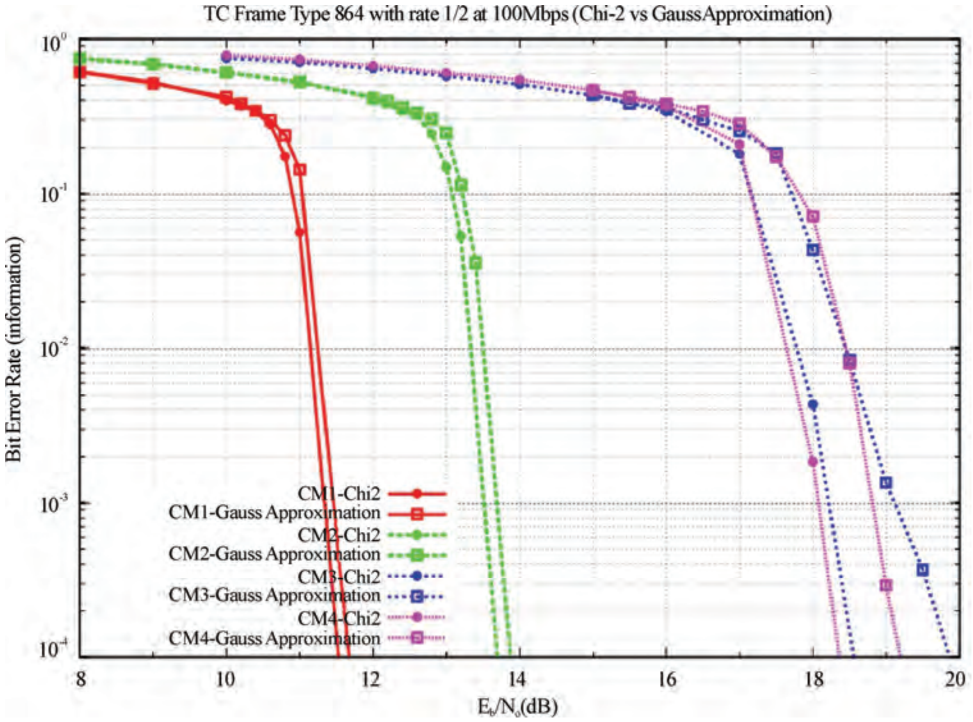


Figure 3. *Chi-squared* vs gaussian approximation in float precision using duo-binary turbo code at rate 1/2 with *K*=2.

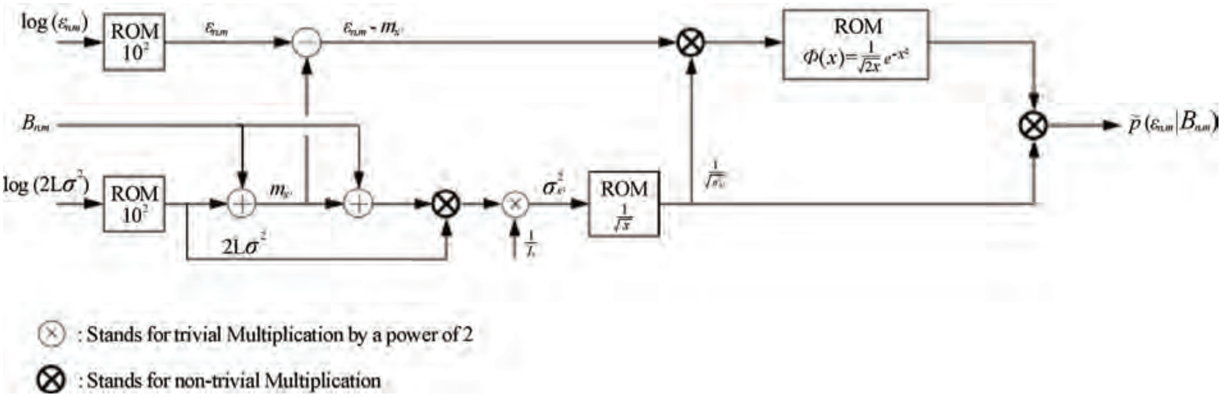

log ($\varepsilon_{n,m}$)
ROM 10^2
$\varepsilon_{n,m}$
$\varepsilon_{n,m} - m_{\chi^2}$
ROM $\Phi(x)=\frac{1}{\sqrt{2x}}e^{-x^2}$
$B_{n,m}$
$\tilde{p}(\varepsilon_{n,m}|B_{n,m})$
log ($2L\sigma^2$)
ROM 10^2
m_{χ^2}
$2L\sigma^2$
$\sigma^2_{\chi^2}$
ROM $\frac{1}{\sqrt{x}}$
$\frac{1}{\sqrt{\sigma_{\chi^2}}}$
$\frac{1}{J_s}$
: Stands for trivial Multiplication by a power of 2
: Stands for non-trivial Multiplication

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

Table 3 lists the Input/Output size look–up table necessary for density computation.

Table 2. Parameters size definition.

Parameters	Quantization size $< wl, iwl >$
$\log E_{n,m}$	$< 6,2 >$
$E_{n,m}$	$< 14,1 >$
$B_{n,m}$	$< 6,2 >$
σ^2	$< 6,1 >$
m_{χ^2}	$< 12,2 >$
$\sigma^2_{\chi^2}$	$< 12,2 >$
□ $p\ (E_{n,m} \mid B_{n,m})$	$< 13,3 >$
$p(E_n \mid x_n)$	$< 13,6 >$
$\pi(x_k)$	$< 4,1 >$

We notice that the total memory occupied by the look-up tables is lower than the *chi-squared* 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.

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

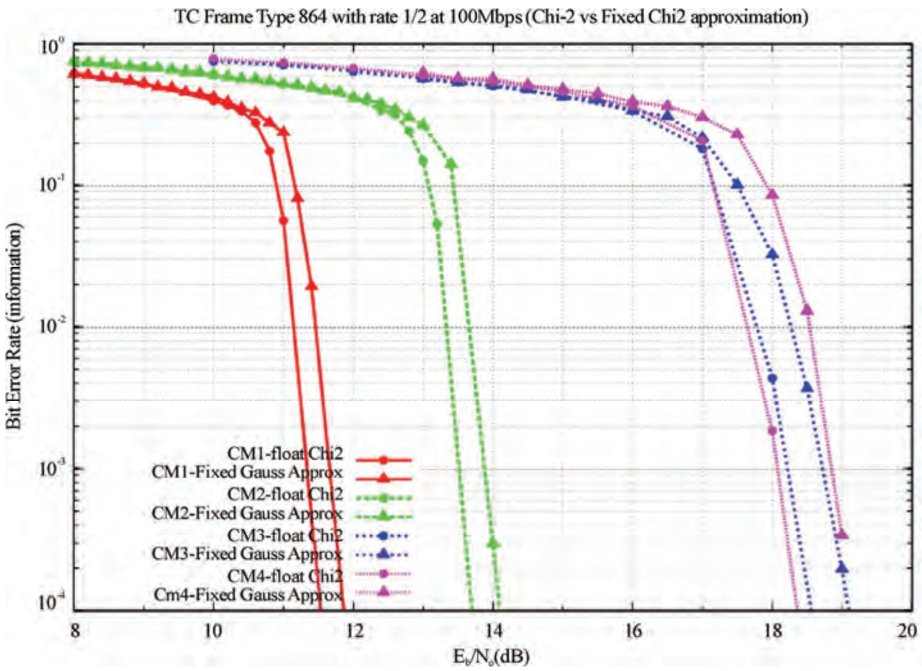


Figure 5. *chi-squared* float precision versus the Gaussian approximation in fixed point precision for K=2.

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 $1/2^{13}$, which is lower than the considered maximum error $\varepsilon = 5\times 10^{-3}$, 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]

$$p(\overset{\overset{Mtimes}{\downarrow}}{E_n} \mid x_n) = \underbrace{\sum_{x_{n-1}} \cdots \sum_{x_{n-K+1}}}_{M^{K-1} cases} \left(\underbrace{\underbrace{\prod_{m=1}^{M} p(\mathrm{E}_{n,m} \mid B_{n,m})}_{(M-1)Mul} \underbrace{\times}_{1Mul} \underbrace{\prod_{k=1}^{K-1} \pi(x_{n-k})}_{(K-2)Mul}}_{(M+K-2)Mul} \right) \tag{14}$$

According to the decomposition in (14), $p(E_n \mid x_n)$ requires $(M+K-2)M^{K-1}$ multiplication. We should notice that $p(E_n \mid x_n)$ 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 $\prod_{m=1}^{M} p(\mathrm{E}_{n,m} \mid B_{n,m})$, we calculate M times $p(\mathrm{E}_{n,m} \mid B_{n,m})$. With respect to the architecture in Figure 4, $p(\mathrm{E}_{n,m} \mid B_{n,m})$ 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.* $p(\mathrm{E}_{n,m} \mid B_{n,m})$. 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_{10}^{*} function which operates in a logarithm to base 10 defined as

$$\begin{aligned} max_{10}^{*}(a,b) &\overset{\Delta}{=} \log(10^{a}+10^{b}) \\ &\overset{\Delta}{=} \max(a,b)+\log(1+10^{-|a-b|}) \end{aligned} \tag{15}$$

the max_{10}^{*} 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_{10}^{*} is an associative operator (see APPENDIX 1 for the proof):

$$max_{10}^{*}(a,b,c) = max_{10}^{*}[max_{10}^{*}(a,b),c] \tag{16}$$

For notation simplicity we consider

$$max_{10}^{*}(a_1,a_2,\ldots,a_N) = max_{10\,i\in\{1,\ldots,N\}}^{*}(a_i) \tag{17}$$

Using the $\max^{*}{}_{10}$operation defined in (15) , the output of the probabilistic equalizer in the logarithmic domain is

[1] Mul stands for Multiplication in (14).

given by:

$$\log p(E_n \mid x_n) = \\ \max^*_{10\,x_{n-1},\ldots,x_{n-K+1}}\left(\sum_{m=1}^{M}\log p(\mathrm{E}_{n,m} \mid B_{n,m}) + \sum_{k=1}^{K-1}\log \pi(x_{n-k})\right) \quad (18)$$

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. $p(E_n \mid x_n) \in [0,1]$, a normalization process is applied as follow:

$$\overline{p}(E_n \mid x_n) = \frac{p(E_n \mid x_n)}{\sum_{x_n} p(E_n \mid x_n)} \quad (19)$$

where $\overline{p}$ is the normalized probability at the output of the equalizer. In the logarithmic domain normalization becomes

$$\log \overline{p}(E_n \mid x_n) = \log p(E_n \mid x_n) - \log\left(\sum_{x_n} p(E_n \mid x_n)\right) \quad (20)$$

$$= \log p(E_n \mid x_n) - \max^*_{10\,x_n} \log p(E_n \mid x_n) \quad (21)$$

Gaussian approximation studied in Section 4 is assumed for equalization in the logarithmic domain so that the energy distribution is feasible or implementable.

Thus, the approximated energy distribution in logarithmic domain is equal to:

$$\log \square\, p\ (\mathrm{E}_{n,m} \mid B_{n,m}) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log \sigma^2_{\chi^2} - \frac{(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2\sigma^2_{\chi^2}\ln 10} \quad (22)$$

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_{slot} . So, the only available data is $\log \mathrm{E}_{n,m}$, $\log B_{n,m}$ and $\log(2L\sigma^2)$. Expending $\sigma^2_{\chi^2}$ in (22), we get

$$\log \square\, p\ (\mathrm{E}_{n,m} \mid B_{n,m}) \propto -\frac{1}{2}\log[2\sigma^2(2L\sigma^2 + 2B_{n,m})] - \frac{L(\mathrm{E}_{n,m} - m_{\chi^2})^2}{4L\sigma^2(2L\sigma^2 + 2B_{n,m})\ln 10} \quad (23)$$

$$\propto -\frac{1}{2}\log(2\sigma^2) - \frac{1}{2}\log(2L\sigma^2 + 2B_{n,m}) - \frac{L(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2.10^{\log(2L\sigma^2(2L\sigma^2+2B_{n,m}))}\ln 10} \quad (24)$$

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

$$\log \square\, p\ (\mathrm{E}_{n,m} \mid B_{n,m}) \propto -\frac{1}{2}\max^*_{10}\left[\log(2L\sigma^2), \log 2 + \log B_{n,m}\right] \\ - \frac{L(\mathrm{E}_{n,m} - m_{\chi^2})^2}{2.10^{\log(2L\sigma^2)+\log(2L\sigma^2+2B_{n,m})}\ln 10} \quad (25)$$

the devision part in (25) can be transformed into multiplicationas follows:

$$\log \square\, p\ (\mathrm{E}_{n,m} \mid B_{n,m}) \propto -\frac{1}{2}\max^*_{10}\left[\log(2L\sigma^2), \log 2 + \log B_{n,m}\right] \\ - \frac{L}{2\ln 10}(\mathrm{E}_{n,m} - m_{\chi^2})^2 10^{-\gamma} \quad (26)$$

where γ is defined as

$$\gamma = \log(2L\sigma^2) + \max^*_{10}[\log(2L\sigma^2), \log 2 + \log B_{n,m}] \quad (27)$$

and $\mathrm{E}_{n,m} - m_{\chi^2}$ is easily calculated as follows

$$\mathrm{E}_{n,m} - m_{\chi^2} = 10^{\log \mathrm{E}_{n,m}} - \left(10^{\log(2L\sigma^2)} + 10^{\log B_{n,m}}\right) \quad (28)$$

It is noticed that the energy distribution in logarithmic domain is achieved by the mean of two lookup tables. A first ROM for the $\max^*_{10}$ function and a second ROM for 10^x 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.

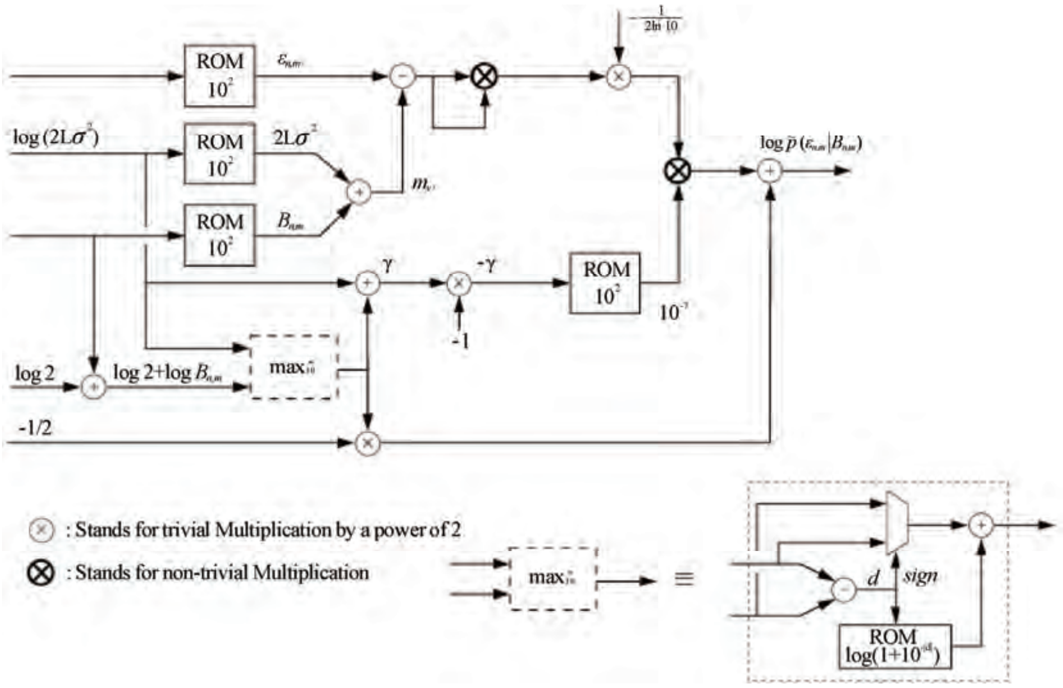


Figure 6. Energy distribution architecture for logarithmic equalizer.

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.

Parameters	Quantization size $<wl,iwl>$
$\log E_{n,m}$	<6,2>
$\log B_{n,m}$	<8,2>
$\log(2L\sigma^2)$	<7,4>
m_{χ^2}	<8,4>
$\log\tilde{p}\ (E_{n,m}\,\vert\,B_{n,m})$	<7,4>
$\log p(E_n\,\vert\,x_n)$	<6,4>
$\log\pi(x_k)$	<6,4>

Table 5. ROM input/output size.

Parameters	Input size	Output size	Table size (Kbits)
$g(x)=\log(1+10^{-\vert d\vert})$	<6,2>	<4,0>	0.25
$h(x)=10^x$	<9,5>	<8,3>	4

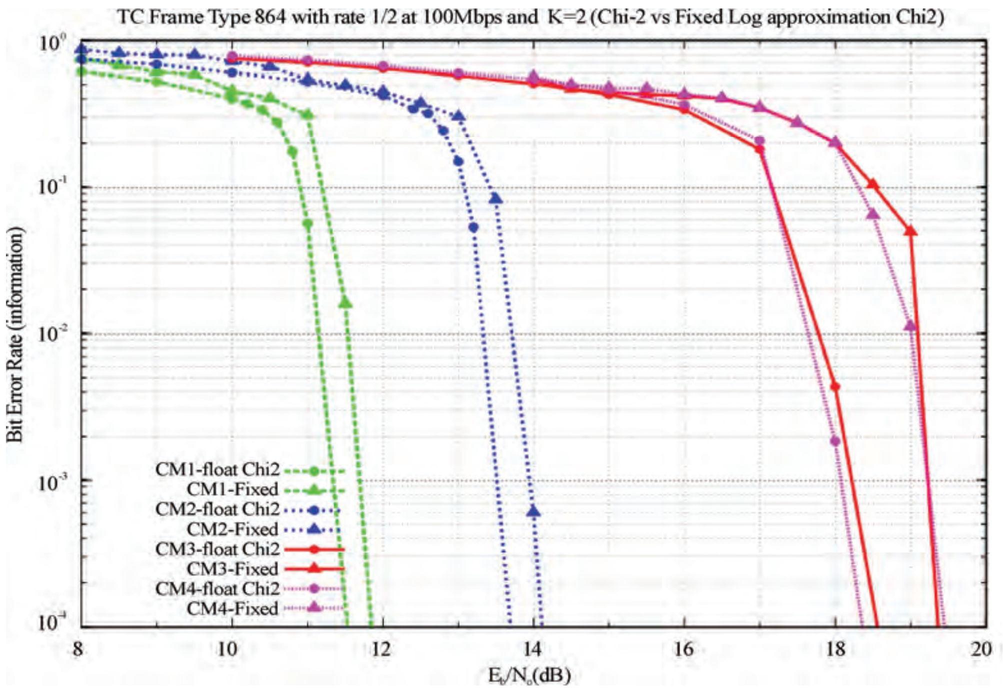


Figure 7. ***Chi-squared*** **float precision versus the logarithmic Gaussian approximation in fixed point precision for *K*=2.**

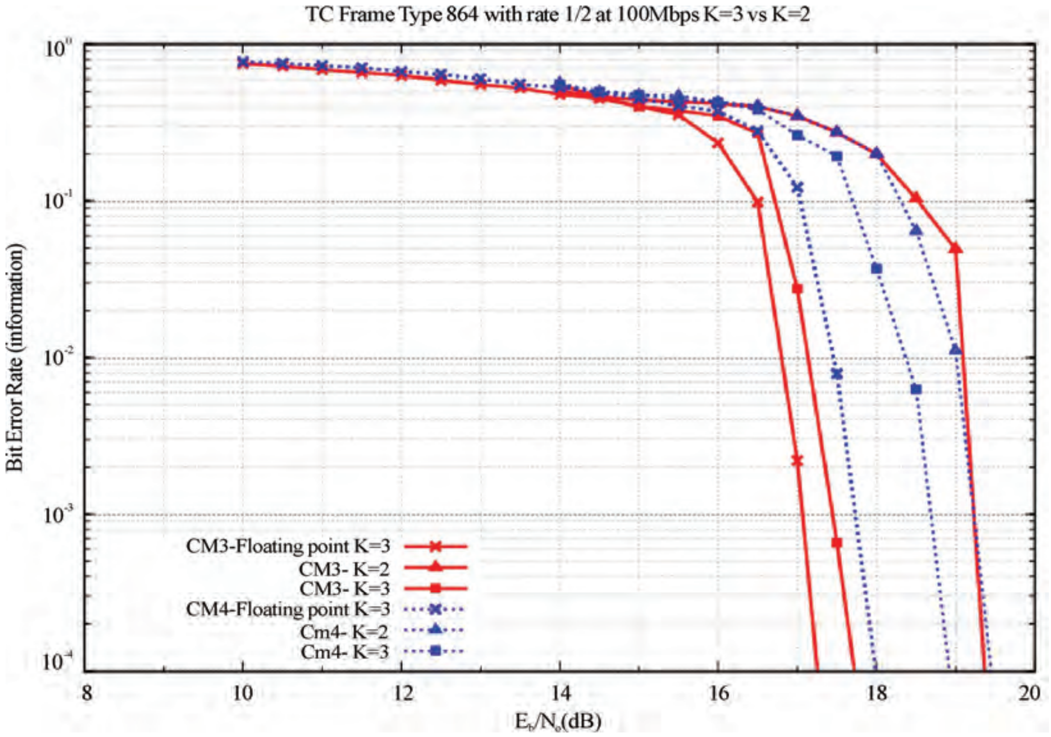


Figure 8. Simulation in fixed point data types for CM3 and CM4.

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 $6.4Gmultiplication/s$. Regarding the memory size, the logarithmic approximati-

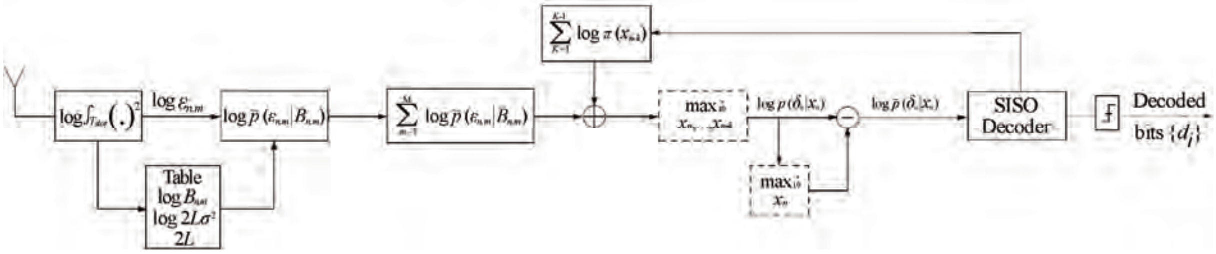

$\log\int_{T_{dot}}(.)^2$
$\log \varepsilon_{n,m}$
$\log \bar{p}(\varepsilon_{n,m}|B_{n,m})$
Table
$\log B_{n,m}$
$\log 2L\sigma^2$
$2L$
$\sum_{m=1}^{M} \log \bar{p}(\varepsilon_{n,m}|B_{n,m})$
$\sum_{K=1}^{K-1} \log \pi(x_{n-k})$
$\max^*_{10}$
$x_{n_1} \dots x_{n-k}$
$\log p(\delta_i|x_n)$
$\max^*_{10}$
x_n
$\log \bar{p}(\delta_i|x_n)$
SISO
Decoder
Decoded
bits $\{d_j\}$

Figure 9. Equalizer architecture in the logarithmic domain.

Table 6. Complexity requirement with 4-PPM and *K*=2.

	Number of Multiplications		
Function	Linear x^2	Linear x^2 approximated	Logarithmic x^2 approximated
$p(E_n \| x_n)$	$(M{+}K{-}2)M^K$ $3.2^0 GMultip/s$	$(M{+}K{-}2)M^K$ $3.2\ GMultip/s$	0
$p(E_{n,m} \| B_{n,m})$	0	$3M^{K+1}$ $9.6\ GMultip/s$	$2M^{K+1}$ $6.4\ GMultip/s$
Total Equalizer Multiplications	3. 2 *GMultip/s*	12.8 *GMultip/s*	6.4 *GMultip/s*
Total required memory per level of parallelism	*448 Mbit*	30.25 *Kbit*	16.25 *Kbit*

on requires $16.25 Kbits$. 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 $400 Mhz$ and the frequency of table access is $3.2 Ghz$ 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.* $16.25 Kbits \times 8 = 130 Kbits$) which is lower than the required memory in linear domain ($242 Kbits$). 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 a 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

[1] L. Yang and G. B. Giannakis, “Ultra-wideband communications: an idea whose time has come,” IEEE Signal Processing Magazine, Vol. 21, No. 6, pp. 26–54, November 2004.

[2] J. D. Choi and W. E. Stark, “Performance of ultra-wideband communications with suboptimal receivers in multipath channels,” IEEE Journal on Selected Areas in Communications, Vol. 20, No. 9, pp. 1754–1766, December 2002.

[3] R. Hoctor and H. Tomlinson, “Delay-hopped transmitted reference RF communications,” IEEE Conference on Ultra-Wideband Systems and Technologies, pp. 265–269, May 2002.

[4] V. Lottici, L. Wu, and Z. Tian, “Inter-symbol interference mitigation in high-data-rate uwb systems,” IEEE International Conference on Communications, pp. 4299–4304, June 2007.

[5] Y. Zhang, H. Wu, Q. Zhang, and P. Zhang, “Interference mitigation for coexistence of heterogeneous ultra-wideband systems”, EURASIP Journal on Wireless Communications and Networking, pp. 1–13, 2006.

[6] M. E. Sahin and H. Arslan, “Inter-symbol interfrence in high data rate uwb communications using energy detector receivers,” IEEE International Conference on UWB, ICU, pp. 176–179, September 2005.

[7] S. Mekki, J. L. Danger, B. Miscopein, J. Schwoerer and J. J. Boutros, “Probabilistic equalizer for ultra-wideband energy detection,” IEEE 67th Vehicular Technology Conference (VTC), pp. 1108–1112, May 2008.

[8] M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, December 1972.

[9] J. A. Erfanian and S. Pasupathy, “Low-complexity parallel-structure symbol-by-symbol detection for ISI channels,” IEEE Pacific Rim Conf. Communications, Computers and Signal Processing, pp. 350–353, 1989.

[10] A. J. Viterbi, “An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes,” IEEE Journal On Selected Areas In Communications, Vol. 16, No. 2, pp. 260–264, February 1998.

[11] H. Urkowitz, “Energy detection of unknown deterministic signals,” Proceedings of the IEEE, Vol. 55, No. 4, pp. 523–531, April 1967.

[12] A. H. M. Ross, “Algorithm for calculating the noncentral chisquare distribution,” IEEE Transactions on Information Theory, Vol. 45, No. 4, pp. 1327–1333, May 1999.

[13] N. C. Severo and M. Zelen, “Normal approximation to the chisquared and non-central F probability functions,” Biometrika, Vol. 47, No. 3/4, pp. 411–416, December 1960.

[14] L. Canal, “A normal approximation for the chi-square distribution,” Computational Statistics & Data Analysis, Vol. 48, No. 4, pp. 803–808, April 2005.

[15] J.-T. Zhang, “Approximate and asymptotic distributions of chi-squared-type mixtures with applications,” Journal of the American Statistical Association, Vol. 100, pp. 273–285, March 2005.

[16] R. Saadane, D. Aboutajdine, A. M. Hayar, and R. Knopp, “On the estimation of the degrees of freedom of in-door UWB channel,” IEEE 61st, Vehicular Technology Conference (VTC), Vol. 5, pp. 3147–3151, May 2005.

[17] J. G. Proakis, Digital Communications, Second Edition, New York, McGraw Hill, 1989.

[18] J. Foerster, “Channel modeling sub-committee report final,” IEEE P802.15-02/368r5-SG3a, Tech. Rep., 18 November 2002.

[19] S. Mekki, J. L. Danger, B. Miscopein, and J. J. Boutros, “EM channel estimation in a low-cost UWB receiver based on energy detection,” IEEE International Symposium on Wireless Communication Systems 2008 (ISWCS 08), pp. 214–218, October 2008. [Online]. Available: http://samimekki.free.fr/.

[20] “Digital video broadcasting (DVB); interaction channel for satellite distribution systems,” ETSI EN 301 790 V.1.3.1, Tech. Rep., March 2004.

[21] “Digital video broadcasting (DVB); interaction channel for satellite distribution systems; guidelines for the use of en 301 790,” ETSI TR 101 790 V.1.2.1, Tech. Rep., January 2003.

[22] “TC1000-xX DVB-RCS Turbo Decoder v2.1,” Turbo-Concept, Tech. Rep., February 2005.

[23] AD8318 1 MHz to 8 GHz, 70 dB Logarithmic Detector/Controller. [Online]. Available: http://www.analog.com/en/prod/0%2C2877%2CAD8318%2C00.html.

[24] SystemC User’s Guide, version 2.0. [Online]. Available: http://www.systemc.org.

[25] W. J. Gross and P. G. Gulak, “Simplified MAP algorithm suitable for implementation of turbo decoders,” Electron ics Letters, Vol. 34, No. 16, pp. 1577–1578, August 1998.

Appendix

max_{10}^{*} Proprieties

$$max_{10}^{*}(a,b,c) = max_{10}^{*}[max_{10}^{*}(a,b),c] \tag{29}$$

Proof.

From the definition of max_{10}^{*} we have

$$max_{10}^{*}(a,b,c) = \log(10^{a} + 10^{b} + 10^{c}) \tag{30}$$

in other hand we can write

$$10^{a} + 10^{b} = 10^{\log(10^{a}+10^{b})} = 10^{max_{10}^{*}(a,b)} \tag{31}$$

Rewriting (29) taking on consideration (30), we get

$$max_{10}^{*}(a,b,c) = \log(10^{max_{10}^{*}(a,b)} + 10^{c}) \tag{32}$$

$$= max_{10}^{*}[max_{10}^{*}(a,b),c] \tag{33}$$