University of Wollongong

Research Online

1-1-2012

A low cost soft mapper for turbo equalization with high order modulation

Licai Fang
University Of Western Australia

Qinghua Guo
University of Wollongong, qguo@uow.edu.au

Defeng (David) Huang
University Of Western Australia, david.huang@uwa.edu.au

Sven Nordholm
Curtin University

Recommended Citation

A low cost soft mapper for turbo equalization with high order modulation

Abstract

In an MMSE based turbo equalization system, a soft mapper calculates the symbol mean and variance based on extrinsic Log-Likelihood-Ratios (LLRs) information coming from a Soft-Input Soft-Output (SISO) decoder. In this paper, we investigate the complexity of this module, and in particular, we employ a 3-segment linear approximation approach to calculate the mean and variance of data symbols from LLRs. For FPGA and VLSI implementation, we propose novel piecewise linear functions which map LLR to the mean and variance directly without the use of any two-variable-input multipliers. Simulation results for 16-QAM and 64-QAM show that the no multiplier approach has close BER performance to the 3-segment linear approximation approach with multipliers. © 2012 IEEE.

Keywords

order, high, equalization, turbo, modulation, mapper, low, soft, cost

Disciplines

Engineering | Science and Technology Studies

Publication Details

A Low Cost Soft Mapper for Turbo Equalization with High Order Modulation

Licai Fang*, Qinghua Guo*†, Defeng (David) Huang*, Sven Nordholm‡

*School of EECE, the University of Western Australia, WA 6009, Australia
†School of ECTE, the University of Wollongong, NSW 2522, Australia
‡Department of ECE, Curtin University, WA 6102, Australia
Email:licaifang@gmail.com, {qinghua.guo, david.huang}@uwa.edu.au, s.nordholm@curtin.edu.au

Abstract—In an MMSE based turbo equalization system, a soft mapper calculates the symbol mean and variance based on extrinsic Log-Likelihood-Ratios (LLRs) information coming from a Soft-Input Soft-Output (SISO) decoder. In this paper, we investigate the complexity of this module, and in particular, we employ a 3-segment linear approximation approach to calculate the mean and variance of data symbols from LLRs. For FPGA and VLSI implementation, we propose novel piecewise linear functions which map LLR to the mean and variance directly without the use of any two-variable-input multipliers. Simulation results for 16-QAM and 64-QAM show that the no multiplier approach has close BER performance to the 3-segment linear approximation approach with multipliers.

I. INTRODUCTION

In an MMSE based turbo equalization receiver, there needs a soft mapper module to calculate the symbol mean and variance from the extrinsic Log-Likelihood-Ratios (LLRs) of code bits coming from the Soft-Input Soft-Output (SISO) decoder [1]. Direct use of the mathematical expression for the mean and variance of the data symbols, involves the computational complexity $\mathcal{O}(Q2^Q)$ per data symbol where Q is the number of bits per symbol. For high order constellations, this may be too high for practical implementation. Fortunately, high order constellations normally have symmetric properties, so the real part and imaginary part can be calculated separately. This will reduce the complexity to $\mathcal{O}((Q/2)2^{Q/2})$. By exploring the characteristics of constellation such as square QAM with gray mapping, several methods have been proposed to further reduce the complexity [2] [3]. In [2], for 16-QAM and 64-QAM, equations were presented to calculate the symbol mean and variance from bit probabilities. But converting bit probabilities from LLRs needs exponential and division operations which are not suitable for an FPGA or VLSI based implementation. In [3], after a complex derivation and with the aid of Maclaurin series, equations were derived to calculate the mean from Log-Likelihood-Ratio (LLR) directly. In this paper, by using linear approximation to the bit probability, we get new equations for calculating the mean and variance from LLRs. A typical implementation of the soft mapper needs several two-variable-input multipliers. Whereas in application of limited hardware
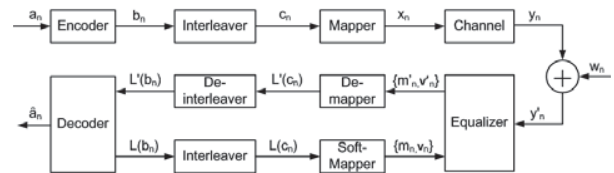
Fig. 1.   Turbo Equalization System Block Diagram

resource such as FPGA, the number of embedded multipliers are limited, so it is desirable to use other resources such as small size of distributed memory or block memory as an alternative. Motivated by this, we propose novel piecewise linear equations to calculate the mean and variance from LLRs without using multipliers.

The reminder of this paper is organized as follows. Section II describes the turbo equalization system model. Then after analyzing the computational complexity of the soft mapper, we propose a no multiplier version of the soft mapper in Section III. Simulation results are shown in Section IV. Finally, we summarize this paper in Section V.

II. SYSTEM MODEL

We consider a turbo equalization system as shown in Fig. 1. At the transmitter side, information bits $\{a_n\}$ are encoded to a code sequence $\{b_n\}$, which is permuted to $\{c_n\}$ by an interleaver. $\{c_n\}$ is then grouped into length-Q subsequence $[c_{n,1}, c_{n,2}, ..., c_{n,Q}]$ and mapped to symbol $x_n \in \chi$ with binary mapping $M:\{0,1\}^Q \rightarrow \chi$, where $\chi = \{\alpha_i\}$ stands for a 2^Q-ary symbol alphabet with $\sum_{i=1}^{2^Q} \alpha_i = 0$ and $\sum_{i=1}^{2^Q} |\alpha_i|^2 /2^Q = Q$. After $\{x_n\}$ is transmitted over an ISI channel and corrupted by AWGN w_n, the receiver receives $\{y'_n\}$. As shown in Fig. 1, together with the *a-priori* symbol mean and variance $\{m_n, v_n\}$ from a soft mapper, the received signal $\{y'_n\}$ is then processed by the SISO equalizer, which outputs the extrinsic symbol mean and variance $\{m'_n, v'_n\}$. Then the soft de-mapper module converts $\{m'_n, v'_n\}$ to soft bit LLR $\{L'(c_n)\}$ which is deinterleaved and sent to the SISO decoder. On the other hand, the extrinsic information $\{L(b_n)\}$ from the SISO decoder are interleaved and sent to the soft mapper [4]. The soft mapper converts interleaved LLR $\{L(c_n)\}$ to $\{m_n, v_n\}$ which will be used by the MMSE equalizer. In this paper, we will focus on this soft mapper module.

III. SOFT MAPPER

The soft mapper calculates $\{m_n, v_n\}$ based on extrinsic LLR $L(c_n)$ using the following equations:

$$m_n = E(x_n) = \sum_{i=1}^{2^Q} \alpha_i p(x_n = \alpha_i) \qquad (1)$$

$$v_n = Cov(x_n, x_n) = \sum_{i=1}^{2^Q} |\alpha_i|^2 p(x_n = \alpha_i) - m_n^2 \qquad (2)$$

where each α_i corresponds to a binary vector $\mathbf{s}_i = [s_{i,1}, s_{i,2}, ..., s_{i,Q}]^T$, and the symbol's probability $p(x_n = \alpha_i)$ can be calculated as:

$$p(x_n = \alpha_i) = \prod_{j=1}^{Q} p(c_{n,j} = s_{i,j}) \qquad (3)$$

while $p(c_{n,j} = s_{i,j})$ is the probability of a code bit, which is normally represented by LLR:

$$L_j = ln\frac{p(c_{n,j} = 0)}{p(c_{n,j} = 1)} = ln\frac{p(c_{n,j} = 0)}{1 - p(c_{n,j} = 0)}. \qquad (4)$$

With (1) - (3), the computational complexity is $\mathcal{O}(Q2^Q)$. Many modulation schemes employ separable constellation structure, for example, the square 64-QAM modulation can be treated as two separated 8-PAM modulations for the real and imaginary components respectively. As a result, (1) can be changed to [2]:

$$\Re(m_n) = \sum_{i=1}^{2^{Q/2}} \beta_i p(\Re(x_n) = \beta_i) \qquad (5)$$

$$\Im(m_n) = \sum_{i=1}^{2^{Q/2}} \gamma_i p(\Im(x_n) = \gamma_i) \qquad (6)$$

where $\Re(\cdot)$ and $\Im(\cdot)$ represent the real and imaginary parts of a complex symbol $(\cdot)$ respectively and $\{\beta_i\}$ and $\{\gamma_i\}$ are the corresponding real and imaginary PAM constellation alphabets. Let $\widetilde{\Re}(v_n)$ and $\widetilde{\Im}(v_n)$ denote the contributions to v_n that come from symbol's real and imaginary parts, we have:

$$\widetilde{\Re}(v_n) = \sum_{i=1}^{2^{Q/2}} |\beta_i|^2 p(\Re(x_n) = \beta_i) \qquad (7)$$

$$\widetilde{\Im}(v_n) = \sum_{i=1}^{2^{Q/2}} |\gamma_i|^2 p(\Im(x_n) = \gamma_i). \qquad (8)$$

Using (7) and (8), (2) can then be calculated as follows:

$$v_n = \widetilde{\Re}(v_n) + \widetilde{\Im}(v_n) - \Re(m_n)^2 - \Im(m_n)^2. \qquad (9)$$

It can be seen that by using (5)-(9), the complexity is reduced to $\mathcal{O}((Q/2)2^{Q/2})$.
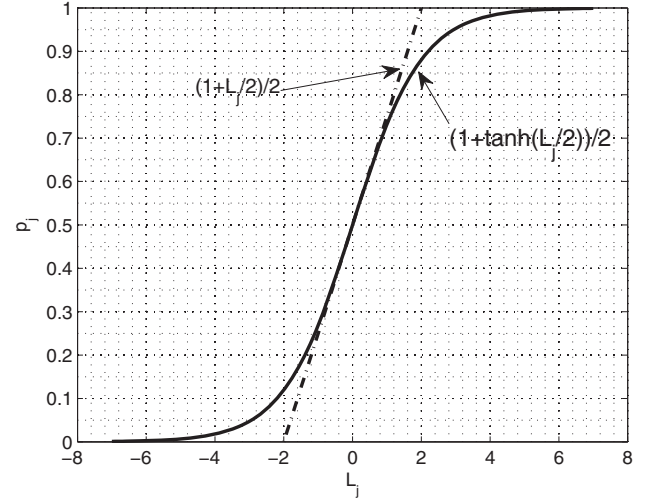

Fig. 2.   3-Segment Piecewise Linear Function

A. 3-segment Linear Approximation Approach

For QAM modulation with gray mapping, (5)-(8) can be further simplified [2]. The following equations are for 16-QAM with gray mapping[1]:

$$\Re(m_n) = (1 - 2p_1)(1 + 2p_2)/\sqrt{2.5} \qquad (10)$$

$$\Im(m_n) = (1 - 2p_3)(1 + 2p_4)/\sqrt{2.5} \qquad (11)$$

$$\widetilde{\Re}(v_n) = (1 + 8p_2)/2.5 \qquad (12)$$

$$\widetilde{\Im}(v_n) = (1 + 8p_4)/2.5 \qquad (13)$$

and for 64-QAM with gray mapping:

$$\Re(m_n) = (1 - 2p_1)(3 + 2p_2 - 2p_3 + 4p_2 p_3)/\sqrt{7} \qquad (14)$$

$$\Im(m_n) = (1 - 2p_4)(3 + 2p_5 - 2p_6 + 4p_5 p_6)/\sqrt{7} \qquad (15)$$

$$\widetilde{\Re}(v_n) = (9 + 16p_2 - 8p_3 + 32p_2 p_3)/7 \qquad (16)$$

$$\widetilde{\Im}(v_n) = (9 + 16p_5 - 8p_6 + 32p_5 p_6)/7 \qquad (17)$$

where the bit probability $p_j = p(c_{n,j} = 0)$ can be easily derived from LLR based on (4) with $tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$, as follows:

$$p_j = (1 + tanh(L_j/2))/2. \qquad (18)$$

Typically, (18) can be implemented with a lookup table. In this work, as an alternative, we use a piecewise linear approximation. From Maclaurin Series, we have

$$tanh(x) = x - \frac{1}{3}x^3 + \frac{1}{15}x^5 - \cdots . \qquad (19)$$

If we use the first order linear approximation $tanh(x) \approx x$, p_j can be calculated by a 3-segment linear function as follows (see Fig.2):

$$p_j = \begin{cases} 0, & L_j \leq -2 \\ (1 + L_j/2)/2, & -2 < L_j < 2 \\ 1. & L_j \geq 2 \end{cases} \qquad (20)$$

[1]The factors in (10)-(17) are different from those in [2] because we normalize constellation energy per symbol to Q (e.g. 4 for 16-QAM).

B. Implementation Without Multipliers

Using (10)-(17) and (9), besides the addition operations, the calculations of m_n and v_n also need several two-variable-input multipliers, and single-variable-input multipliers (e.g., $1/\sqrt{7}$, $1/7$). For some applications (e.g., using VLSI and FPGA), the resources of multipliers are too tight or too expensive to be implemented. This motivates us to propose piecewise linear functions without a multiplier to calculate the mean and variance.

Obviously, we can use small lookup tables to handle single-variable-input multiplier functions and x^2 function. But there still need several two-variable-input multipliers which are not suitable for lookup table implementation because of too many entries to be stored. In the following part, we propose an approach to remove all two-variable-input multipliers.

After applying a clipping function and convert L_j into $\hat{L}_j$ as follows:

$$\hat{L}_j = \begin{cases} -2, & L_j \leq -2 \\ L_j, & -2 < L_j < 2 \\ 2. & L_j \geq 2 \end{cases} \quad ,$$

the 3-segment linear approximation function of (20) can be re-written as: $p_j = (1 + \hat{L}_j/2)/2$. With this, we represent (10)-(13) for 16-QAM in terms of $\hat{L}_j$ as follows:

$$\Re(m_n) = (-\hat{L}_1/2)(2 + \hat{L}_2/2)/\sqrt{2.5} \quad (21)$$

$$\Im(m_n) = (-\hat{L}_3/2)(2 + \hat{L}_4/2)/\sqrt{2.5} \quad (22)$$

$$\widetilde{\Re}(v_n) = (5 + 2\hat{L}_2)/2.5 \quad (23)$$

$$\widetilde{\Im}(v_n) = (5 + 2\hat{L}_4)/2.5 \quad . \quad (24)$$

Similarly, for 64-QAM, (14)-(17) are re-written as:

$$\Re(m_n) = (-\hat{L}_1/2)(4 + \hat{L}_2 + \hat{L}_2\hat{L}_3/4)/\sqrt{7} \quad (25)$$

$$\Im(m_n) = (-\hat{L}_4/2)(4 + \hat{L}_5 + \hat{L}_5\hat{L}_6/4)/\sqrt{7} \quad (26)$$

$$\widetilde{\Re}(v_n) = (21 + 8\hat{L}_2 + 2\hat{L}_3 + 2\hat{L}_2\hat{L}_3)/7 \quad (27)$$

$$\widetilde{\Im}(v_n) = (21 + 8\hat{L}_5 + 2\hat{L}_6 + 2\hat{L}_5\hat{L}_6)/7 \quad . \quad (28)$$

To avoid the use of two-variable-input or multiple-variable-input multipliers, we then impose the following approximation,

$$\hat{L}_m\hat{L}_n = 0, \quad -2 < \hat{L}_m < 2 \ \& \ -2 < \hat{L}_n < 2. \quad (29)$$

As a result, we get the no multiplier version of equations (21)-(24) for 16-QAM as follows:

$$\hat{\Re}(m_n) = \begin{cases} -3\hat{L}_1/2, & \hat{L}_2 = 2 \\ -\hat{L}_1/2, & \hat{L}_2 = -2 \\ -\hat{L}_1. & -2 < \hat{L}_2 < 2 \end{cases}$$

$$\Re(m_n) \approx \hat{\Re}(m_n)/\sqrt{2.5}, \quad (30)$$

$$\hat{\Im}(m_n) = \begin{cases} -3\hat{L}_3/2, & \hat{L}_4 = 2 \\ -\hat{L}_3/2, & \hat{L}_4 = -2 \\ -\hat{L}_3. & -2 < \hat{L}_4 < 2 \end{cases}$$

$$\Im(m_n) \approx \hat{\Im}(m_n)/\sqrt{2.5}, \quad (31)$$

$$\widetilde{\Re}(v_n) = (5 + 2\hat{L}_2)/2.5 \quad (32)$$

$$\widetilde{\Im}(v_n) = (5 + 2\hat{L}_4)/2.5 \quad (33)$$

and equations (25)-(28) for 64-QAM as the following:

$$\hat{\Re}(m_n) = \begin{cases} 6 + \hat{L}_3/2, & \hat{L}_1 = -2 \ \& \ \hat{L}_2 = 2 \\ 2 - \hat{L}_3/2, & \hat{L}_1 = -2 \ \& \ \hat{L}_2 = -2 \\ 4 + \hat{L}_2, & \hat{L}_1 = -2 \ \& \ -2 < \hat{L}_2 < 2 \\ -6 - \hat{L}_3/2, & \hat{L}_1 = 2 \ \& \ \hat{L}_2 = 2 \\ -2 + \hat{L}_3/2, & \hat{L}_1 = 2 \ \& \ \hat{L}_2 = -2 \\ -4 - \hat{L}_2, & \hat{L}_1 = 2 \ \& \ -2 < \hat{L}_2 < 2 \\ -3\hat{L}_1, & -2 < \hat{L}_1 < 2 \ \& \ \hat{L}_2 = 2 \\ -\hat{L}_1, & -2 < \hat{L}_1 < 2 \ \& \ \hat{L}_2 = -2 \\ -2\hat{L}_1. & -2 < \hat{L}_1 < 2 \ \& \ -2 < \hat{L}_2 < 2 \end{cases}$$

$$\Re(m_n) \approx \hat{\Re}(m_n)/\sqrt{7}, \quad (34)$$

$$\hat{\Im}(m_n) = \begin{cases} 6 + \hat{L}_5/2, & \hat{L}_3 = -2 \ \& \ \hat{L}_4 = 2 \\ 2 - \hat{L}_5/2, & \hat{L}_3 = -2 \ \& \ \hat{L}_4 = -2 \\ 4 + \hat{L}_4, & \hat{L}_3 = -2 \ \& \ -2 < \hat{L}_4 < 2 \\ -6 - \hat{L}_5/2, & \hat{L}_3 = 2 \ \& \ \hat{L}_4 = 2 \\ -2 + \hat{L}_5/2, & \hat{L}_3 = 2 \ \& \ \hat{L}_4 = -2 \\ -4 - \hat{L}_4, & \hat{L}_3 = 2 \ \& \ -2 < \hat{L}_4 < 2 \\ -3\hat{L}_3, & -2 < \hat{L}_3 < 2 \ \& \ \hat{L}_4 = 2 \\ -\hat{L}_3, & -2 < \hat{L}_3 < 2 \ \& \ \hat{L}_4 = -2 \\ -2\hat{L}_3. & -2 < \hat{L}_3 < 2 \ \& \ -2 < \hat{L}_4 < 2 \end{cases}$$

$$\Im(m_n) \approx \hat{\Im}(m_n)/\sqrt{7}, \quad (35)$$

$$\hat{\Re}(v_n) = \begin{cases} 37 + 6\hat{L}_3, & \hat{L}_2 = 2 \\ 5 - 2\hat{L}_3, & \hat{L}_2 = -2 \\ 25 + 12\hat{L}_2, & -2 < \hat{L}_2 < 2 \ \& \ \hat{L}_3 = 2 \\ 17 + 4\hat{L}_2, & -2 < \hat{L}_2 < 2 \ \& \ \hat{L}_3 = -2 \\ 21 + 8\hat{L}_2 + 2\hat{L}_3. & -2 < \hat{L}_2 < 2 \ \& \ -2 < \hat{L}_3 < 2 \end{cases}$$

$$\widetilde{\Re}(v_n) \approx \hat{\Re}(v_n)/7, \quad (36)$$

$$\hat{\Im}(v_n) = \begin{cases} 37 + 6\hat{L}_6, & \hat{L}_5 = 2 \\ 5 - 2\hat{L}_6, & \hat{L}_5 = -2 \\ 25 + 12\hat{L}_5, & -2 < \hat{L}_5 < 2 \ \& \ \hat{L}_6 = 2 \\ 17 + 4\hat{L}_5, & -2 < \hat{L}_5 < 2 \ \& \ \hat{L}_6 = -2 \\ 21 + 8\hat{L}_5 + 2\hat{L}_6. & -2 < \hat{L}_5 < 2 \ \& \ -2 < \hat{L}_6 < 2 \end{cases}$$

$$\widetilde{\Im}(v_n) \approx \hat{\Im}(v_n)/7, \quad (37)$$

We note that after using lookup table handling factors of $1/\sqrt{2.5}$, $1/\sqrt{7}$, $1/2.5$, $1/7$ and x^2 function, the above equations contain only shift and addition operations[2]. We also note that the method of using (29) recursively can also be extended to other high order constellations to remove two-variable-input or multiple-variable-input multipliers.

[2]Operations like $\hat{L}_2/2$, $3\hat{L}_2$, $6\hat{L}_2$ and $8\hat{L}_2$ can be implemented using shift and addition operations. For example, $6\hat{L}_2$ can be implemented by $4\hat{L}_2 + 2\hat{L}_2$.
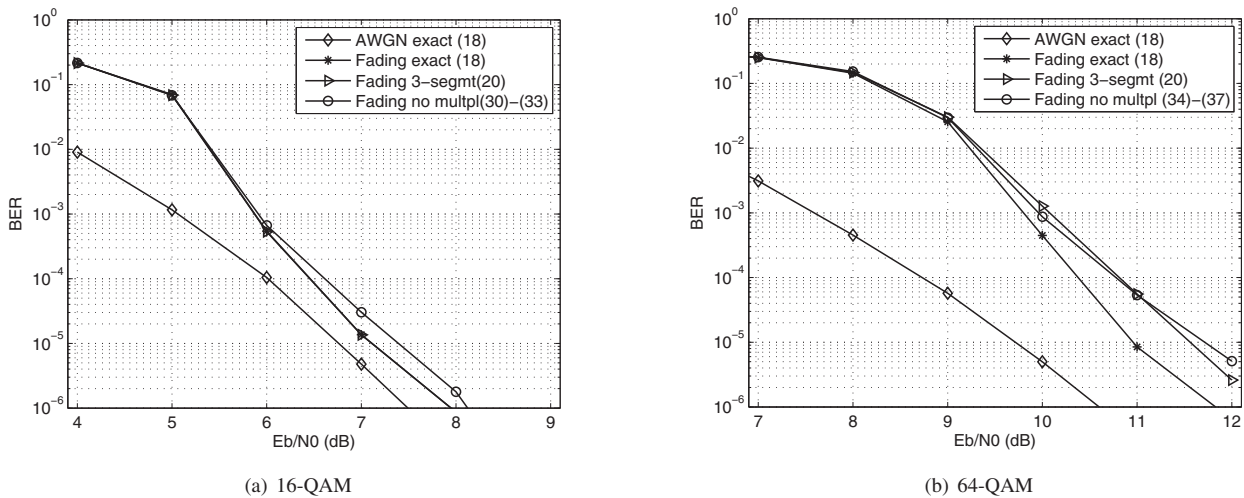
(a) 16-QAM

(b) 64-QAM

Fig. 3. Performance of MMSE Turbo Equalization System with Different Mapper

IV. SIMULATION RESULT

We consider a turbo equalization system with an MMSE equalizer [4]. A rate-1/2 convolutional code with generator $(23, 35)_8$ and square 2^Q-QAM modulation with gray mapping are used, and the APP decoder is implemented using the BCJR algorithm [5]. For fading channel, we use a 16-tap independent quasi static block fading Rayleigh channel model, i.e., the channel taps h_j for $j = 0, 1, ..., 15$ are independently generated from a complex Gaussian distribution PDF $\mathcal{CN}(h_j; 0, \frac{1}{16})$, and then the energy of the generated channel taps is normalized to 1. During the simulations, we use random symbol level interleaver [3] and assume perfect channel information is available [1]. The frame length is 1024 and the number of iterations is 10.

When simulating approximations (30)-(37), we use 7-bits to represent each LLR, and lookup tables with 64 entries of 8 bits for factors $1/\sqrt{2.5}$, $1/\sqrt{7}$, $1/2.5$, $1/7$ and x^2 function. We simulate system BER performance with different mapper equations under fading channel and also present the BER performances under AWGN channel for reference. The exact result denotes the performance of using equations (10)-(17) and (18), while the 3-segment approximation result denotes the performance of using (10)-(17) and (20).

Fig. 3(a) shows the BER performance for 16-QAM. It can be seen that the 3-segment approximation (20) has nearly the same performance as the exact one using (18), while the no multipliers version of using (30)-(33) has a slightly worse Eb/N0 performance under BER of 10^{-4}. From Fig. 3(b), for 64-QAM, we can see that 3-segment approximation (20) and no multiplier version have similar performance, and they all have about 0.6dB Eb/N0 performance loss at BER of 10^{-5} compared with the exact one of (18). However, our

approach can remove all two-variable-input multipliers (e.g. 8 multipliers in [7] for 64-QAM) in the soft mapper module.

V. CONCLUSION

In this paper, we have investigated the computational complexity of the soft mapper in a turbo equalization receiver. The use of a 3-segment linear approximation of the bit probability enables system BER performance close to the exact one under 16-QAM and induce small performance loss under 64-QAM. Then, based on this 3-segment linear approximation, we proposed a novel no multiplier approach to calculate the mean and variance from LLRs directly. Simulations show that no multiplier approach has similar performance as the approach with multipliers.

REFERENCES

[1] M. Tuchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Trans. on Information Theory*, vol. 57, no. 2, pp. 920-952, Feb. 2011.

[2] A. Tomasoni, M. Ferrari, D. Gatti, F. Osnato, and S. Bellini, "A Low Complexity Turbo MMSE Receiver for W-LAN MIMO Systems," *inProc. IEEE Int. Conf. Communications*, June 2006, pp. 4119-4124.

[3] S. Sun, T. T. Tjhung, and Y, Li, "An Iterative Receiver for Groupwise Bit-Interleaved Coded QAM STBC OFDM," *IEEE Vehicular Technology Conference-Spring*, vol. 3, pp. 1256–1260, May 2004.

[4] Q. Guo and D. Huang, "A concise representation for the soft-in soft-out LMMSE detector," *IEEE Commun. Lett.*, vol. 15, no. 5, pp. 566-568, May 2011.

[5] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, 1974

[6] Petit, P.F., "Turbo-equalization for QAM Constellations," Ph.D. dissertation, Institute for Telecommunications Research, the University of South Australia, August, 2002.

[7] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid State Circuits*, vol. 46, no. 7, pp. 1754-1765, Jul. 2011

---

[3]As shown in [6], for gray mapping, the bit level interleaver has worse BER performance than the symbol level interleaver. Therefore, we only present simulation results with symbol level interleaver.