

Research Letter

Simplified LLRs for the Decoding of Single Parity Check Turbo Product Codes Transmitted Using 16QAM

Maher Arar, Claude D'Amours, and Abbas Yongacoglu

School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, Ontario, Canada K1N 6N5

Correspondence should be addressed to Maher Arar, marar051@uottawa.ca

Received 20 August 2007; Accepted 1 October 2007

Recommended by Guosen Yue

Iterative soft-decision decoding algorithms require channel log-likelihood ratios (LLRs) which, when using 16QAM modulation, require intensive computations to be obtained. Therefore, we derive four simple approximate LLR expressions. When using the maximum a posteriori probability algorithm for decoding single parity check turbo product codes (SPC/TPCs), these LLRs can be simplified even further. We show through computer simulations that the bit-error-rate performance of $(8, 7)^2$ and $(8, 7)^3$ SPC/TPCs, transmitted using 16QAM and decoded using the maximum a posteriori algorithm with our simplified LLRs, is nearly identical to the one achieved by using the exact LLRs.

Copyright © 2007 Maher Arar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

By using soft-input/soft-output (SISO) maximum a posteriori (MAP) decoders with iterative (i.e., turbo) decoding, it was shown in [1] that the Shannon limit can be approached within 0.7dB for a code rate r of 0.5, a block length of 65536 bits, and with 18 turbo iterations. The turbo principle with SISO decoding was later applied to other existing codes like single parity check (SPC) product code [2–4]. SISO decoders, including those used in decoding SPC/TPC, are normally derived for the binary input AWGN channel. In applications where high bandwidth efficiency is required, it becomes crucial to use higher-order modulation schemes such as 16QAM, which has a raw bandwidth efficiency of 4 bits/s/Hz compared to 1 bit/s/Hz for BPSK. Since SISO decoders require soft channel information for each bit, we need to extract this information from the received 16QAM symbols. A brute-force approach is to compute the LLRs using the exact expression, which requires intensive computations. Instead, several authors [5, 6] proposed the approximation of the LLR expressions for these bits.

In this letter, we take a similar approach and approximate the LLRs, but we do this based on different assumptions, which result in simpler LLR expressions. When the MAP algorithm is used for the soft decoding of SPC/TPC, computing these approximated LLRs requires four simple arithmetic

operations per 16QAM symbol. Simulation results show that, while being very simple, the proposed method's BER performance for SPC/TPC is virtually identical to the one achieved when using the exact LLRs, which are more cumbersome to compute.

2. BACKGROUND

In this paper, we will use the following mapping of bit u to channel symbol x : $u = 0 \rightarrow x = -1$ and $u = 1 \rightarrow x = +1$. We assume that the received channel observation y has been corrupted by AWGN n (with variance σ_n^2), that is, $y = x + n$.

2.1. Log-likelihood ratio and intrinsic information

In uncoded binary communication systems, we can extract a channel LLR $L_c(y)$, also called maximum a posteriori probability (MAP), from the received channel observation y using conditional probabilities as follows:

$$\begin{aligned} L(x | y) &= \log \frac{p(x = +1 | y)}{p(x = -1 | y)} = \log \left[\frac{p(y | x = +1)}{p(y | x = -1)} \cdot \frac{p(x = +1)}{p(x = -1)} \right] \\ &= \log \frac{p(y | x = +1)}{p(y | x = -1)} + \log \frac{p(x = +1)}{p(x = -1)}, \\ L(\hat{x}) &= L_c(y) + L_a(x). \end{aligned} \tag{1}$$

$L_a(x)$ is a priori LLR of symbol x that is equal to zero for equiprobable symbols, and $L_c(y)$ is an LLR related to channel measurement that can be found as [4]

$$L_c(y) = \frac{2}{\sigma_n^2} y. \quad (2)$$

Unfortunately, we may not be able to obtain σ_n^2 at the receiver, and consequently we approximate $L_c(y)$ simply as $L_c(y) = y$ (i.e., by setting σ_n^2 to 2). In other words, the LLR of the channel is the same as the channel observation obtained at the output of the matched filter. This soft information is normally referred to as *intrinsic* information in the literature as it is obtained from the actual channel measurement itself.

2.2. Extrinsic information

In a coded communication system, extra information can be extracted from the coded bits. In its simplest form, given the three bits u , u_1 , and u_2 that are related by parity equation $u = u_1 \oplus u_2$, where \oplus represents the modulo 2 addition operator, we have from [3]

$$\begin{aligned} L_{\text{ext}}(u) &= L(u_1 \oplus u_2) \\ &= \log \frac{1 + e^{L(u_1)} e^{L(u_2)}}{e^{L(u_1)} + e^{L(u_2)}} \\ &\approx -\text{sign}(L(u_1)) \cdot \text{sign}(L(u_2)) \cdot \min(|L(u_1)|, |L(u_2)|). \end{aligned} \quad (3)$$

In the general case, where the parity equation contains J bits, we have [3]

$$L_{\text{ext}}(u) \approx (-1)^{J-1} \prod_{j=1}^J \text{sign}(L(u_j)) \cdot \min_{j=1, \dots, J} (|L(u_j)|). \quad (4)$$

The above equation tells us that in coded systems, we can extract additional information, called *extrinsic* information, that we can use to improve the reliability of the soft decision, that is,

$$L(\hat{x}) = L_c(y) + L_a(x) + L_{\text{ext}}(y). \quad (5)$$

Equations (2), (4), and (5) form the core of the MAP algorithm [3] used to decode SPC/TPC. It can be easily shown that $L_c(\alpha y) = \alpha L_c(y)$ and $L_{\text{ext}}(\alpha y) = \alpha L_{\text{ext}}(y)$ for any positive constant α . In this case, and for equiprobable symbols, $L(\hat{x})$ will be scaled by a positive integer α and hence the sign will not be affected. Therefore, $L(\hat{x})$ can be normalized by α . This fact will be exploited to simplify the expressions for the 16QAM LLRs.

2.3. 16QAM modulation

Square 16QAM modulation is a bandwidth-efficient modulation technique. Each 16QAM symbol represents 4 bits giving a raw bandwidth efficiency of 4 bits/s/Hz.

One way of minimizing the overall bit error rate in digital communication systems is by using the Gray mapping,

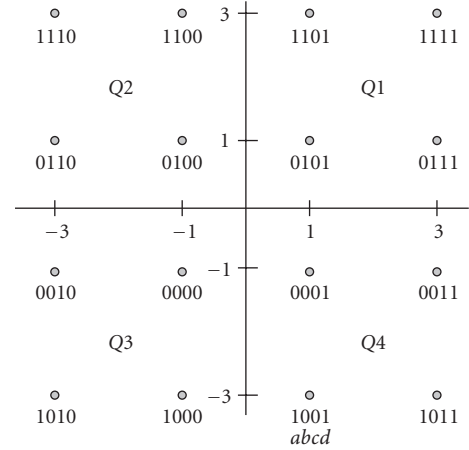


FIGURE 1: A Gray-coded 16QAM constellation. Q1–Q4 denote the quadrants.

which makes sure that each symbol differs from its closest neighbors by one bit only. One possible Gray-coded 16QAM constellation is shown in Figure 1.

2.4. LLR calculation for 16QAM

Computing the exact LLR for each bit in a 16QAM modulated signal is rather computationally intensive as this involves computing the ratio of the sum of 8 probabilities, where that bit is 1, to the sum of 8 probabilities, where that bit is 0. Mathematically, this involves the computation of the following expression for each bit $u \in \{a, b, c, d\}$ of the received channel observation $y = y_I + jy_Q$:

$$L(u) = \log \frac{\sum_{s_{1,u} \in \{S^{(u=1)}\}} e^{-|y - s_{1,u}|^2 / \sigma^2}}{\sum_{s_{0,u} \in \{S^{(u=0)}\}} e^{-|y - s_{0,u}|^2 / \sigma^2}}, \quad (6)$$

where $s_{1,u} = s_{1I,u} + js_{1Q,u}$ and $s_{0,u} = s_{0I,u} + js_{0Q,u}$ are the 16QAM symbols where bit $u \in \{a, b, c, d\}$ is equal to 1 and 0, respectively.

3. SIMPLE APPROXIMATED EXPRESSIONS FOR CHANNEL LLRS

One way of simplifying (6) is to only retain the largest value in the numerator and the denominator, in other words, to retain only the value that has the smallest Euclidian distance. That is, given the channel observation y , we find the closest neighbor, where bit $u = 1$, and the closest neighbor, where bit $u = 0$, to compute the following approximated LLR:

$$\tilde{L}(u) = \log \frac{e^{-|y - \tilde{s}_{1,u}|^2 / \sigma_n^2}}{e^{-|y - \tilde{s}_{0,u}|^2 / \sigma_n^2}} \quad (7a)$$

$$= \frac{1}{\sigma_n^2} \left[|y - \tilde{s}_{0,u}|^2 - |y - \tilde{s}_{1,u}|^2 \right] \quad (7b)$$

$$\begin{aligned} &= 2 \frac{y_I}{\sigma_n^2} (\tilde{s}_{1I,u} - \tilde{s}_{0I,u}) + 2 \frac{y_Q}{\sigma_n^2} (\tilde{s}_{1Q,u} - \tilde{s}_{0Q,u}) \\ &+ \frac{1}{\sigma_n^2} (\tilde{s}_{0I,u}^2 - \tilde{s}_{1I,u}^2) + \frac{1}{\sigma_n^2} (\tilde{s}_{0Q,u}^2 - \tilde{s}_{1Q,u}^2), \end{aligned} \quad (7c)$$

where $\tilde{s}_{1,u}$ and $\tilde{s}_{0,u}$ are the closest (in Euclidean distance) 16QAM symbols to y , where bit u is equal to 1 and 0, respectively.

Equation (7c) can be further simplified in different ways depending on the bit in question (i.e., bit a , b , c , or d shown in Figure 1). We clearly see from Figure 1 that for bits a and b we have $s_{1I,u} = s_{0I,u}$, and consequently the expression in (7c) is simplified to

$$\tilde{L}(a) = 2 \frac{y_Q}{\sigma_n^2} (s_{1Q,a} - s_{0Q,a}) + \frac{1}{\sigma_n^2} (s_{0Q,a}^2 - s_{1Q,a}^2) \quad (8)$$

for bit a ; and

$$\tilde{L}(b) = 2 \frac{y_Q}{\sigma_n^2} (s_{1Q,b} - s_{0Q,b}) + \frac{1}{\sigma_n^2} (s_{0Q,b}^2 - s_{1Q,b}^2) \quad (9)$$

for bit b .

In (8), $s_{1Q,a} = 3$ and $s_{0Q,a} = 1$, or $s_{1Q,a} = -3$ and $s_{0Q,a} = -1$, depending on whether y_Q is positive or negative, respectively. Therefore, (8) can be written differently as follows:

$$\tilde{L}(a) = \frac{4}{\sigma_n^2} (|y_Q| - 2). \quad (10)$$

For bit b , we have $s_{1Q,b} = 1$ and $s_{0Q,b} = -1$ when $-2 \leq y_Q \leq +2$, $s_{1Q,b} = 1$, and $s_{0Q,b} = -3$ when $y_Q < -2$ and $s_{1Q,b} = 3$ and $s_{0Q,b} = -1$ when $y_Q > 2$. Therefore, (9) can be rewritten as follows:

$$\tilde{L}(b) = \begin{cases} \frac{4}{\sigma_n^2} y_Q, & \text{if } -2 \leq y_Q \leq +2, \\ \frac{8}{\sigma_n^2} (y_Q + 1), & y_Q < -2, \\ \frac{8}{\sigma_n^2} (y_Q - 1), & y_Q > +2. \end{cases} \quad (11)$$

Using similar analysis, we can find approximate expressions for $L(c)$ as

$$\tilde{L}(c) = \frac{4}{\sigma_n^2} (|y_I| - 2) \quad (12)$$

and for $L(d)$ as

$$\tilde{L}(d) = \begin{cases} \frac{4}{\sigma_n^2} y_I, & \text{if } -2 \leq y_I \leq +2, \\ \frac{8}{\sigma_n^2} (y_I + 1), & y_I < -2, \\ \frac{8}{\sigma_n^2} (y_I - 1), & y_I > +2. \end{cases} \quad (13)$$

The approximate LLRs in (11) and (13) can be further approximated in the region of interest (let us say $y_I = -4$ to $+4$ and $y_Q = -4$ to $+4$) by either of two methods: by linear interpolation across the entire region of interest or by linear interpolation across a specific segment of that region

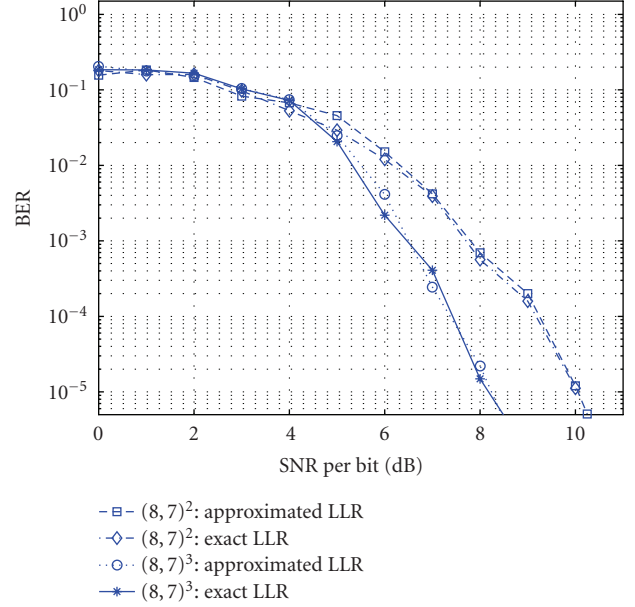


FIGURE 2: BER performance of $(8,7)^2$ and $(8,7)^3$ SPC/TPCs using exact and approximated LLRs after three iterations.

(i.e., $-2 \leq y_I \leq +2$ and $-2 \leq y_Q \leq +2$). We here choose the latter method to obtain

$$\begin{aligned} \tilde{L}(b) &\approx \frac{4}{\sigma_n^2} y_Q, \\ \tilde{L}(d) &\approx \frac{4}{\sigma_n^2} y_I, \end{aligned} \quad (14)$$

respectively.

As mentioned in Section 2.2, the MAP algorithm for decoding SPC/TPC is insensitive to the same positive scaling factor. Therefore, (10), (12), and (14) can be simplified to yield very simple expressions for the channel LLRs:

$$\begin{aligned} \tilde{L}(a) &= |y_Q| - 2, \\ \tilde{L}(b) &= y_Q, \\ \tilde{L}(c) &= |y_I| - 2, \\ \tilde{L}(d) &= y_I. \end{aligned} \quad (15)$$

Computing $\tilde{L}(b)$ and $\tilde{L}(d)$ require no arithmetic operation, whereas the computation of $\tilde{L}(a)$ and $\tilde{L}(c)$ requires computing two absolute values and two simple additions. When compared to (15), the exact LLR expression in (6) clearly requires substantial computations.

It should be emphasized that since the derived expressions for the LLRs in (10), (12), and (14) are not derived with any particular decoding algorithm in mind, they can be used with any iterative decoder such as LDPC and convolutional turbo decoders (of course as long as Gray-coded 16QAM modulation is used). These approximated expressions still only require a fraction of the complexity as compared to (6). If we set the noise variance σ_n^2 in these expressions to an arbitrary scalar (in this case to 4), as practically

and commonly done when using these decoders, we obtain the same expressions in (15).

4. SIMULATION RESULTS

Figure 2 compares the BER performance of two SPC/TPCs using the exact LLR expression in (6) for finding the channel soft information to that decoded using the approximate LLR expressions in (15). In both cases, we clearly see that the performance is nearly identical. It should be emphasized that the performance achieved using the proposed method for calculating the channel LLRs costs only a tiny fraction of the computational effort required to achieve the same performance using the exact LLR expression. Furthermore, the proposed simplified LLRs do not require knowledge of noise variance.

5. CONCLUSION

In this letter, we introduced simple LLRs for 16QAM modulation for use with SISO decoding of SPC/TPC. Simplified LLRs for general M-QAM modulation schemes can be easily derived using similar analysis to the one presented in this letter.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes. 1," in *Proceedings of IEEE International Conference on Communications (ICC '93)*, vol. 2, pp. 1064–1070, Geneva, Switzerland, May 1993.
- [2] D. M. Rankin and T. A. Gulliver, "Single parity check product codes," *IEEE Transactions on Communications*, vol. 49, no. 8, pp. 1354–1362, 2001.
- [3] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.
- [4] B. Sklar, "A primer on turbo code concepts," *IEEE Communications Magazine*, vol. 35, no. 12, pp. 94–102, 1997.
- [5] R. Pyndiah, A. Picart, and A. Glavieux, "Performance of block turbo coded 16-QAM and 64-QAM modulations," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '95)*, vol. 2, pp. 1039–1043, Singapore, November 1995.
- [6] S. Allpress, C. Luschi, and S. Felix, "Exact and approximated expressions of the log-likelihood ratio for 16-QAM signals," in *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers (ACSSC '04)*, vol. 1, pp. 794–798, Pacific Grove, Calif, USA, November 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

