BIT-INTERLEAVED CODED MODULATION WITH ITERATIVE DECODING USING CONSTELLATION SHAPING

Boon Kien Khoo, Stéphane Y. Le Goff, Bayan S. Sharif, and Charalampos C. Tsimenidis

School of Electrical, Electronic and Computer Engineering,
University of Newcastle upon Tyne,
NE1 7RU, United Kingdom.
Email: b.k.khoo@ncl.ac.uk

ABSTRACT

We investigate the association between constellation shaping and bit-interleaved coded modulation with iterative decoding (BICM-ID). To this end, we consider a technique which consists of inserting shaping block codes between mapping and channel coding functions in order to achieve constellation shaping. By assuming the example of a 2-bit/s/Hz 16-QAM BICM-ID, it is demonstrated using computer simulations that this technique can improve the performance of BICM-ID schemes by a few tenths of decibels.

1. INTRODUCTION

Recently, some methods for combining constellation shaping and bit-interleaved turbo-coded modulation have been proposed for the design of bandwidth- and power-efficient communication systems over additive white Gaussian noise (AWGN) channels [1], [2]. In this letter, we investigate the application of constellation shaping to bit-interleaved coded modulation with iterative decoding (BICM-ID) [3], [4]. To this end, we consider a shaping technique which consists of partitioning the basic constellation into several sub-constellations so that the lower energy signals are transmitted more frequently than their higher energy counterparts [5]. In practice, such technique can be implemented by inserting shaping block codes between mapping and channel coding functions. At the receiver side, an iterative decoding algorithm is used to exchange extrinsic information between the channel decoder, shaping decoder, and demapper blocks. Throughout this work, without loss of generality, we focus on the design of a 2-bit/s/Hz BICM-ID system employing a 16-ary quadrature amplitude modulation (QAM) constellation.

This letter is organized as follows: In Section 2, the structures of our BICM-ID transmitter and receiver are presented. Computer simulation results are shown in Section 3. Finally, conclusions are drawn in Section 4.

2. SYSTEM MODEL

2.1 Transmitter structure of BICM-ID using shaping coding

Fig. 1 shows the transmitter structure of a 2-bit/s/Hz 16-QAM BICM-ID scheme using shaping codes. A frame of N_b information bits is first encoded by a rate-R_c convolutional encoder. The resulting encoded sequence is divided into four parallel binary vectors by a serial-to-parallel (S/P) converter, which are then interleaved using a set of random interleavers π_j, $j \in \{1, 2, 3, 4\}$. The outputs of the first two interleavers are broken into L successive K-bit vectors

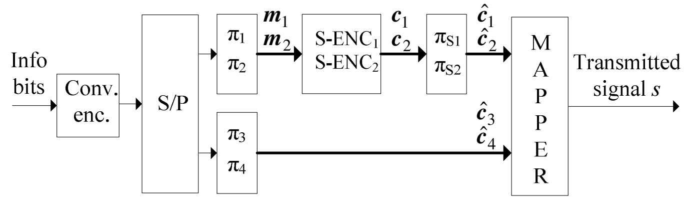

Figure 1: Structure of the 16-QAM BICM-ID transmitter.

$\boldsymbol{m}_j$, $j \in \{1, 2\}$. Each vector $\boldsymbol{m}_j$ is fed into a binary shaping encoder S-ENC$_j$, $j \in \{1, 2\}$, which generates a corresponding N-bit codeword $\boldsymbol{c}_j$. Both shaping encoders S-ENC$_j$ are identical and their rate R_s is given by $R_s = K/N$ $(N > K)$. We recall that shaping encoders are designed so that the probability of a zero in codewords $\boldsymbol{c}_j$ is maximized [2], [5]. Each sequence composed of L codewords $\boldsymbol{c}_j$, $j \in \{1, 2\}$, is then randomly interleaved (π_{Sj}). For convenience, we can hereafter view the resulting sequence as a succession of L codewords $\hat{\boldsymbol{c}}_j$, $j \in \{1, 2\}$.

The remaining streams available at the output of interleavers π_j, $j \in \{3, 4\}$, can also be seen as composed of L successive N-bit codewords $\hat{\boldsymbol{c}}_j$, $j \in \{3, 4\}$. Finally, a vector of four bits $(\hat{c}_{i,1}, ..., \hat{c}_{i,4})$, where $\hat{c}_{i,j}$ is the i-th bit in the codeword $\hat{\boldsymbol{c}}_j$, $j \in \{1, 2, 3, 4\}$, is mapped onto a signal point of 16-QAM constellation. The spectral efficiency, η, of the system is given by

$$\eta = 2R_c(R_s + 1) \text{ bits/s/Hz.} \tag{1}$$

The 16-QAM constellation is divided into three sub-constellations S_x, $x \in \{1, 2, 3\}$, so that S_1 contains the four signal points with lowest energies, S_2 includes the eight signal points with medium energies, and S_3 is composed of the four signal points with highest energies. Bits $\hat{c}_{i,1}$ and $\hat{c}_{i,2}$, which are originally generated by the shaping encoders, are used to select one of these sub-constellations as follows: If $(\hat{c}_{i,1}, \hat{c}_{i,2})$ = (00), then S_1 is selected, whereas S_2 is chosen whenever $(\hat{c}_{i,1}, \hat{c}_{i,2})$ = (01) or (10). At last, the case $(\hat{c}_{i,1}, \hat{c}_{i,2})$ = (11) leads to the selection of S_3. With such mapping procedure, we guarantee that signals with high energies are transmitted less frequently than low-energy signals since $\Pr\{\hat{c}_{i,j} = 0\} > \Pr\{\hat{c}_{i,j} = 1\}$, $j \in \{1, 2\}$ [2], [5]. We can easily show that, when compared to 16-QAM with equiprobable signaling, the energy saving E_s, in decibels (dB), is given by

$$E_s = 10\log_{10}\left(\frac{5}{9 - 8P_0}\right), \tag{2}$$

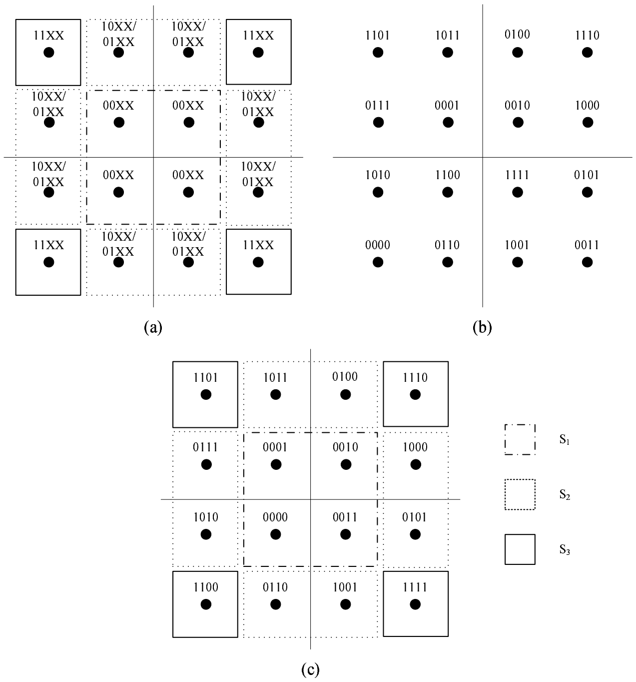

Figure 2: Different mappings for 16-QAM. (a) 16-QAM mapping required for the BICM-ID system with constellation shaping. (b) Schreckenbach's mapping. (c) Modified Schreckenbach's mappping.

where P_0 is the average probability of a zero at the shaping encoder output.

Fig. 2a shows the 16-QAM mapping required for our system. Note that the notation 'XX' in this figure denotes any pair of bits, i.e. '00', '01', '10' or '11'. Therefore, there is a very large number of suitable mappings for designing our BICM-ID scheme with constellation shaping. In [6], Schreckenbach proposed a 16-QAM mapping that optimizes the error performance of BICM-ID at high signal-to-noise ratio (SNR). Such mapping is indicated in Fig. 2b. From an error performance viewpoint, it is obviously desirable to employ a mapping which is as close to that given in Fig. 2b as possible. We found that it is actually possible to obtain a labelling compatible with Fig. 2a by slightly modifying Schreckenbach's mapping. The resulting mapping is shown in Fig. 2c and is obtained by simply swapping labels '0000' and '1100', as well as labels '0011' and '1111' in Schreckenbach's mapping.

2.2 Receiver structure of BICM-ID using shaping coding

The receiver structure of our BICM-ID system is shown in Fig. 3. In order to optimize the error performance, the demapper, shaping decoder and convolutional decoder exchange information in an iterative manner by employing soft-input soft-output (SISO) modules [7]. For each transmitted signal s, the corresponding received signal r is expressed as $r = s + n$, where n is a Gaussian noise sample with zero mean and variance σ^2. Based on sample r and the corresponding *a priori* log-likelihood ratios (LLRs) $L_a^d(\hat{c}_{i,j})$, $j \in \{1,2,3,4\}$, generated by both shaping and convolutional decoders, the demapper calculates an extrinsic LLR $L_e^d(\hat{c}_{i,j})$ associated with bit $\hat{c}_{i,j}$, $j \in \{1,2,3,4\}$, by using well known expressions [3], [4].

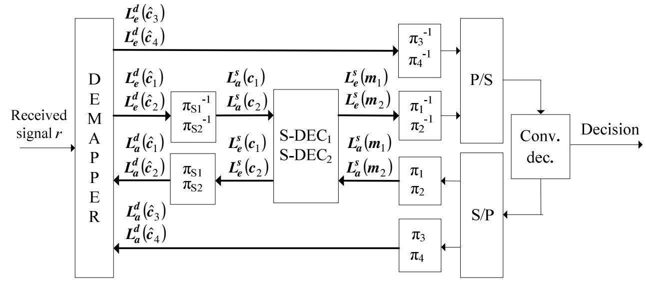

Figure 3: Structure of the 16-QAM BICM-ID receiver.

The LLR $L_e^d(\hat{\boldsymbol{c}}_j)$ of vector $\hat{\boldsymbol{c}}_j$, $j \in \{1,2\}$, is de-interleaved by π_{S1}^{-1} and π_{S2}^{-1}. The resulting LLR $L_a^s(\boldsymbol{c}_j)$ is then used as an *a priori* information by the shaping decoder. The purpose of the SISO shaping decoder S-DEC$_j$, $j \in \{1,2\}$, is to generate both extrinsic LLRs $L_e^s(\boldsymbol{c}_j)$ and $L_e^s(\boldsymbol{m}_j)$ which are associated with codewords $\boldsymbol{c}_j$ and $\boldsymbol{m}_j$, respectively. By using the maximum *a posteriori* (MAP) algorithm, the extrinsic LLR $L_e^s(m_{j,k}), k \in \{1,...,K\}$ corresponding to the k-th bit in message $\boldsymbol{m}_j$ can be computed as

$$L_e^s(m_{j,k}) = \ln\left[\frac{\Pr\{m_{j,k}=1, L_a^s(\boldsymbol{m}_j), L_a^s(\boldsymbol{c}_j)\}}{\Pr\{m_{j,k}=0, L_a^s(\boldsymbol{m}_j), L_a^s(\boldsymbol{c}_j)\}}\right], \quad (3)$$

where $L_a^s(\boldsymbol{m}_j)$ represents the extrinsic LLR generated by the convolutional decoder at the previous iteration. We could show that (3) is actually equivalent to

$$L_e^s(m_{j,k}) = \ln\left[\frac{\sum_{\boldsymbol{m}\in\Omega_k^1} \exp\left\{\sum_{l=1,l\neq k}^{K} t_l L_a^s(m_{j,l}) + \sum_{n=1}^{N} t_n\left(L_a^s(c_{j,n}) - L_n\right)\right\}}{\sum_{\boldsymbol{m}\in\Omega_k^0} \exp\left\{\sum_{l=1,l\neq k}^{K} t_l L_a^s(m_{j,l}) + \sum_{n=1}^{N} t_n\left(L_a^s(c_{j,n}) - L_n\right)\right\}}\right], \quad (4)$$

where Ω_k^t, $t \in \{0,1\}$, denotes the set of all messages $\boldsymbol{m}$ whose k-th bit is equal to t. In this equation, $t_l \in \{0,1\}$ is the value of the l-th bit in the message $\boldsymbol{m}$ under consideration and $t_n \in \{0,1\}$ is the value of the n-th bit in the codeword $\boldsymbol{c}$ associated to this particular message. In (4), L_n represents the *a priori* knowledge, regarding the n-th bit in codewords $\boldsymbol{c}_j$, available before the first decoding iteration starts. It is computed as

$$L_n = \ln\left[\frac{1-P_n}{P_n}\right], \quad (5)$$

where P_n designates the probability that the n-th bit in a codeword $\boldsymbol{c}_j$ is equal to zero. By using an expression very similar to (4), it is possible to calculate LLRs $L_e^s(c_{j,n}), n \in \{1,...,N\}$. The extrinsic LLRs $L_e^s(\boldsymbol{c}_1)$ and $L_e^s(\boldsymbol{c}_2)$ are further fed into the demapper as an *a priori* information for the next iteration, while $L_e^s(\boldsymbol{m}_1)$, $L_e^s(\boldsymbol{m}_2)$, $L_e^d(\hat{\boldsymbol{c}}_3)$ and $L_e^d(\hat{\boldsymbol{c}}_4)$ are processed by the convolutional decoder in the same iteration.

Finally, the extrinsic LLRs generated by the convolutional decoder are split into four parallel vectors and interleaved via π_j, $j \in \{1,2,3,4\}$. The first two vectors $L_a^s(\boldsymbol{m}_1)$ and $L_a^s(\boldsymbol{m}_2)$ are further fed into the shaping decoder for the next iteration, while $L_a^d(\hat{\boldsymbol{c}}_3)$ and $L_a^d(\hat{\boldsymbol{c}}_4)$ together with $L_a^d(\hat{\boldsymbol{c}}_1)$ and $L_a^d(\hat{\boldsymbol{c}}_2)$ are fed back to the demapper.

R_c	R_s	K	N	E_s (dB)	P_0
2/3	1/2	7	14	3.63	0.854
		6	12	3.58	0.851
		5	10	3.38	0.838
3/5	2/3	6	9	2.41	0.766
		4	6	2.22	0.750
4/7	3/4	6	8	1.85	0.717
		3	4	1.55	0.688

Table 1: Several possible configurations for the design of a 2-bit/s/Hz 16-QAM BICM-ID scheme with shaping coding. For each configuration, the energy saving and the corresponding value of P_0 are indicated.

3. SIMULATION RESULTS

We now consider a 2-bit/s/Hz BICM-ID system employing a 16-QAM constellation. The error performance of this system was evaluated using various low-complexity shaping codes whose characteristics are indicated in Table 1 [2], [5]. Since the computational complexity of the shaping decoding varies exponentially with K ($|\Omega_k^1| = |\Omega_k^0| = 2^{K-1}$ in (4)), we decided in this letter to only consider short-length shaping codes for which $K \leq 7$.

The rate-R_c channel codes are obtained by puncturing a 4-state rate-1/2 recursive and systematic convolutional (RSC) code with generator polynomials (7, 5). The iterative decoding at the receiver side is performed in 10 iterations, and the Log-MAP algorithm is used for the decoding of the convolutional code.

Computer simulations were performed by selecting the shaping codes corresponding to the highest energy saving for each configuration in Table 1. Fig. 4 to Fig. 6 illustrate the bit error rate (BER) performance versus signal-to-noise ratio E_b/N_0, where E_b is the energy transmitted per information bit and N_0 is the one-sided noise power spectral density, obtained with the 2-bit/s/Hz 16-QAM BICM-ID schemes using frame sizes N_b = 2000, 6800 and 13400 bits, respectively. For comparison sake, the performance of an equivalent 2-bit/s/Hz 16-QAM BICM-ID system without shaping is also plotted. It can be seen that the use of the shaping code results in a significant performance improvement for both frame sizes. For every frame size, we remark that the maximum value of the shaping gain at BER = 10^{-3} is approximately equal to 0.7 dB. This substantial gain is achieved using the $(K = 7, N = 14, R_c = 2/3)$ configuration. However, in this case, one can notice that there is an error-floor effect occuring at BER level below 10^{-3}. We believe that this error floor is due to the relatively poor error-correcting capabilities of a punctured rate-2/3 4-state convolutional code. Nevertheless, such performance can still be of interest in some applications for which a BER of 10^{-3} is the target.

The error floor level can be reduced by decreasing the channel code rate R_c, i.e. increasing the shaping code rate R_s. For instance, it is seen from Fig. 5 that the error floor only occurs below BER = 10^{-5} when using the $(K = 6, N = 8, R_c = 4/7)$ configuration. However, we notice that decreasing the channel code rate R_c results in smaller shaping gains at high BERs ($\approx 10^{-3}$). There is therefore a compromise between performance at high BERs and performance at medium BERs. We observe that the best trade-off is achieved

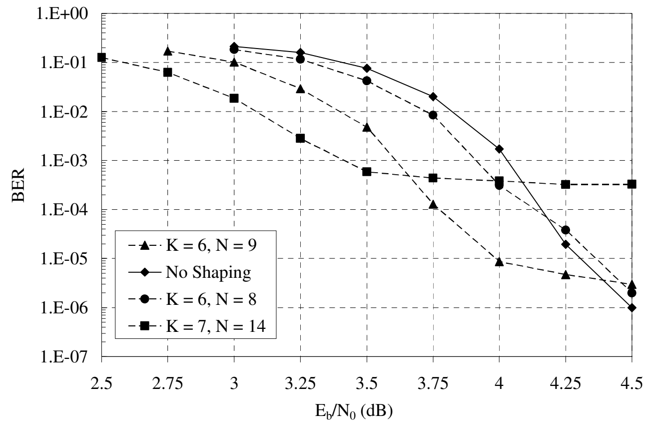

Figure 4: Performance comparison over Gaussian channel between several 2-bit/s/Hz 16-QAM BICM-ID scheme using convolutional coding with 10 iterations and frame sizes N_b = 2000 information bits per frame. Performance of equivalent BICM-ID system without shaping is also shown.

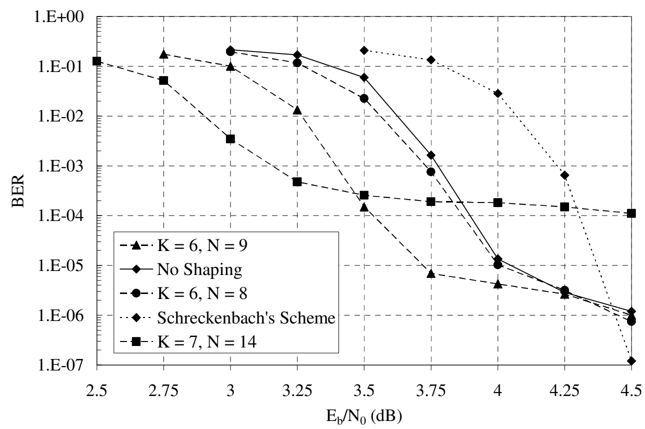

Figure 5: Performance comparison over Gaussian channel between several 2-bit/s/Hz 16-QAM BICM-ID scheme using convolutional coding with 10 iterations and frame sizes N_b = 6800 information bits per frame. Performance of equivalent BICM-ID system without shaping is also shown.

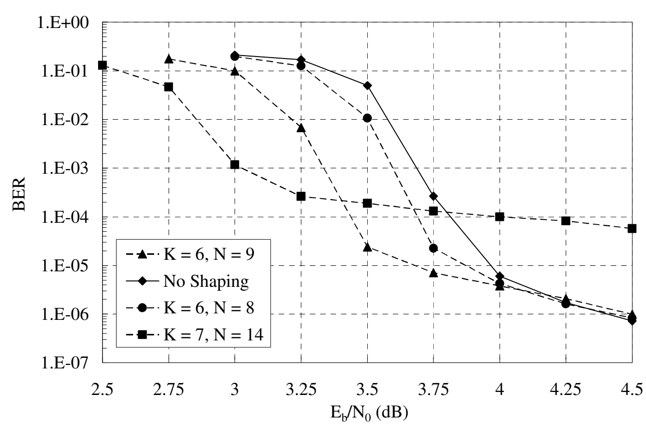

Figure 6: Performance comparison over Gaussian channel between several 2-bit/s/Hz 16-QAM BICM-ID scheme using convolutional coding with 10 iterations and frame sizes N_b = 13400 information bits per frame. Performance of equivalent BICM-ID system without shaping is also shown.

with the $(K=6, N=9, R_c=3/5)$ configuration. Using this particular configuration, we obtain shaping gains equal to 0.30dB, 0.35 dB and 0.30 dB at BER = 10^{-5}, for frame sizes of 2000, 6800 and 13400 bits, respectively.

It is interesting to compare our results with those achieved using Schreckenbach's mapping for a 16-QAM BICM-ID scheme without constellation shaping [6]. Note that, 6800 bits per frame used in our system is equivalent to the interleaver size of 10000 bits considered in [6]. At BER = 10^{-3}, our scheme using the $(K=7, N=14)$ configuration outperforms this BICM-ID scheme by 1.1 dB, whereas at BER = 10^{-5}, our scheme with the $(K=6, N=9)$ configuration is 0.75 dB better than this scheme.

4. CONCLUSION

In this letter, we investigated the gain which can be obtained by applying constellation shaping to the design of BICM-ID. We showed that the error performance of a 2-bit/s/Hz 16-QAM convolutional coding scheme can be improved by 0.7 dB at BER = 10^{-3}, and 0.35 dB at BER = 10^{-5}, when employing carefully selected shaping codes.

REFERENCES

[1] D. Raphaeli and A. Gurevitz, "Constellation shaping for pragmatic turbo-coded modulation with high spectral efficiency," *IEEE Trans. Commun.*, vol. 52, pp. 341–345, March 2004.

[2] S. Y. Le Goff, B. S. Sharif and S. A. Jimaa, "Bit-interleaved turbo-coded modulation using shaping coding," *IEEE Commun. Letters* vol. 9, pp. 246-248, March 2005.

[3] X. Li and J. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *Electron. Letters*, vol. 34, pp. 942–943, May 1998.

[4] S. ten Brink, J. Speidel, and R. Yan, "Iterative demapping and decoding for multilevel modulation," *in Proc. IEEE Globecom Conference*, Sydney, Australia, Nov. 1998, pp. 579–584.

[5] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inform. Theory*, vol. 36, pp. 726–740, July 1990.

[6] F. Schreckenbach, N. Görtz, J. Hagenauer and G. Bauch, "Optimized symbol mappings for bit-interleaved coded modulation with iterative decoding," *in Proc. IEEE Globecom Conference*, San Francisco, USA, Dec. 2003, pp. 3316-3320.

[7] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks," *Europ. Trans. Telecommun.*, vol. 9, pp. 155-172, March–April 1998.