ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

Spectrally Efficient Modulation and Turbo Coding for Wireless Communication in Gaussian Channel

Amer H. Al Habsi, Yahiea Al-Naiemy, Hussain M. Al-Rizzo, Robert Akl, and Maytham M. Hammood

Abstract - The objective of a communication system is to optimally utilize the scarce spectrum and to minimize the energy required to transmit a unit information entity while maintaining a certain performance measure, commonly expressed in terms of Bit Error Rate (BER). Turbo codes achieve performances very close to the channel capacity which is the absolute physical limit as proved by Shannon [1]. These Turbo codes, however, when used with BPSK modulation, as were originally presented [2], are not very spectrally efficient. In this paper, we present a method which combines Quadrature Amplitude Modulation (QAM) with Turbo codes. In this scheme, the information bits are coded using standard Turbo code algorithms then grouped into k-bit symbols to be modulated using the more spectrally efficient QAM modulation. Standard Turbo decoders like the MAP decoder are then used to decode the bit-symbols into bits. Certain functions which map received symbols to individual bits while preserving the soft information of the received symbols are proposed. These functions are specific to the particular QAM modulation used and to the particular mapping of the symbols to the constellation points. At the receiver, these functions are used to decouple the received symbols into component bit-symbols with low loss in the soft information. The gain achieved from the proposed soft decision algorithm allows the use of short interleavers with reduced channel delay and complexity introduced by the MAP decoder. This paper demonstrates that Turbo codes can be used with significantly smaller interleavers thus reducing the complexity and delay for decoding. Computer simulations* showed enhanced performance with high spectrum utilization. The system is shown to achieve better BER performance compared to a Reed Solomon code with similar rate and modulation. The Matlab code developed during this research is available upon request from Dr. Hussain M. Al-Rizzo.

***Index Terms*-QAM modulation, Shannon's capacity, MAP decoder, Turbo coding.**

I. INTRODUCTION

Shannon's capacity theorem proved the existence of a coding scheme which achieves reliable communication at a rate equal to or less than channel capacity [1]. However, no such scheme was given. Turbo codes achieve performance very close to Shannon's limits [2]. Berrou, and Glavieux [3] showed these codes performing within 1/2 dB of Shannon's limit at certain transmission rates. A Turbo encoder is composed of two or more convolutional encoders and an interleaver. An information bit-stream is fed to the first convolutional encoder and to the interleaver. The latter rearranges the bits and then feeds them to the second encoder. The two component encoders are usually, but not necessarily, of the same structure. The outputs of the Turbo encoder consist of the unmodified information bits and the outputs of the two encoders. Figure 1 shows a typical architecture of a Turbo encoder with two Recursive Systematic Convolutional (RSC) encoders. These encoders, when used as components of a Turbo encoder, are usually designed to be recursive with the output being fed back to the encoder.

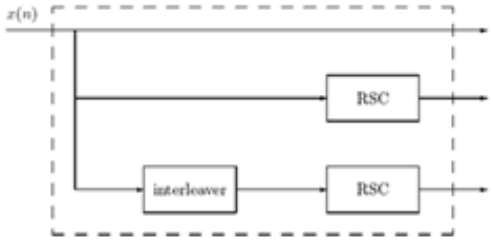


Fig 1: A Typical Turbo encoder

A RSC encoder consists of a shift register whose output, as well as the output of the internal flip-flops, are fed back to the input. Fig. 2 shows a RSC with a three flip-flop shift register. The flip-flops whose outputs are used in the feedback are chosen according to some mathematical formulation. Look-up tables exist that show the optimal flip-flop locations to be used for both feedbacks as well as output [4].

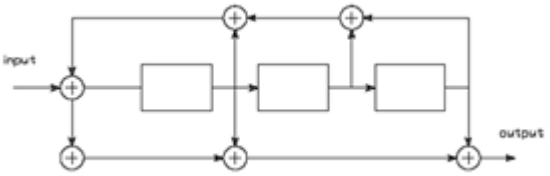


Fig 2: RSC encoder consisting of a three flip-flop shift register and modulo-2 adders.

The interleaver permutes the input bit stream in some fashion. They have been used in digital communications for many years [5]. Traditionally, they have been used to reduce the effects of burst errors especially in systems that use block or convolutional codes. A block code of length *n* that is capable of correcting up to *t* errors in a block, for example, cannot correct the errors if they are more than *t* as would be caused by a burst of noise. However, if the bits are interleaved and spread over multiple blocks, then the effects of the burst error would be reduced. This is true because burst errors tend to be short in time. In Turbo coding, interleaving is done to

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

improve the performance not only in a burst channel, but in a random channel as well. Mathematically, an interleaver ***P***, takes an input bit stream arranged in a vector *x* and produces another vector y such that

$$y = \boldsymbol{P}\, x \tag{1}$$

The permutation matrix ***P***, consists of a single 1 in every row and column and zeros everywhere else. The following ***P*** is an example of a 5×5 interleaver,

$$P = \begin{bmatrix} 00100 \\ 00001 \\ 10000 \\ 01000 \\ 00010 \end{bmatrix} \tag{2}$$

An input sequence like $x = [x_0, x_1, x_2, x_3, x_4]^T$ will be interleaved to $y = [x_2, x_4, x_0, x_1, x_3]^T$.

At the receiver, a de-interleaver $\boldsymbol{P}^{-1}$, is used to re-order the received symbols. The de-interleaver is given by

$$P^{-1} = P^T \tag{3}$$

For the above example, the interleaver is

$$P^{-1} = \begin{bmatrix} 00100 \\ 00010 \\ 10000 \\ 00001 \\ 01000 \end{bmatrix} \tag{4}$$

While the matrix notation is convenient for analysis, in real hardware and/or software implementation, interleaving is done via a table lookup because the matrix is sparse, containing a single '1' in each column. The algorithm shown below describes how this is done. Here, ***Π*** is an array representing the column position of the '1' in each row of ***P*** where the first column is considered position '0'. For the example of ***P*** above, ***Π*** = [2, 4, 0, 1, 3].

Algorithm 1 Software Implementation of Interleaving

Input: $x(n)$: array of bits to be interleaved

Input: $\Pi(n)$: the interleaver array

Output: $y(n)$: output of interleaver

for $n = 0$ to $N - 1$ do

  $y(n) \leftarrow \Pi(x(n))$

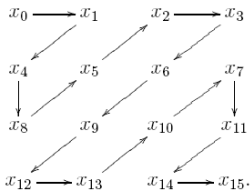
end for

Generally, the length of the interleaver, i.e., the size of the matrix ***P***, determines the error correcting capability of the Turbo code. In the original paper of Berrou, et al. [2], an interleaver of size 64K was used, i.e., it was a 65536 × 65536 matrix. Some interleavers are simply random permutation, while others are designed in a deterministic fashion. Considerable research has been done in the design of shorter and high performing interleavers. A simple deterministic interleaver arranges the input sequence in a rectangular array. The input is placed in the array row-wise but read column-wise. For example, the input sequence $x = [x_0, x_1, \ldots, x_{15}]$, can be put in a 4 ×4 array as:

$$\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15}, \end{matrix}$$

then read column-wise as $x = [x_0, x_4, x_8, x_{12}, x_1, \ldots, x_{15}]$.

Another commonly used deterministic interleaving involves arranging the input sequence in a rectangular (square if possible) array. Then the data is read in a zigzag fashion as shown below.



There are more complex, though deterministic, algorithms to generate interleavers. However, when the interleaver size *N* is large, then random interleavers perform well. To generate a random interleaver ***Π(n)***, first, the array ***Π(n)*** is initialized with the input sequence *0, 1,..., N-1*. Then the array is randomly and uniformly shuffled to produce the interleaver. It should be noted that random interleavers were used throughout the simulations reported in this paper. Their performance is comparable to the systematically designed ones, particularly for large interleaver sizes. The Turbo encoder of Fig.1 is a rate 1/3 where three bits are transmitted for every single information bit. The information bit is sent unmodified, a parity bit from the first RSC encoder and a parity bit from the second RSC encoder (after being interleaved) are also sent. The rate is in fact slightly less than 1/3. Usually after a block is sent through the encoders, *K* zeros, where *K* is the number of storage elements in the RSC, are inserted to the encoders to force them to return to certain states. Therefore, the actual rate is

$$r = \frac{N}{3N + 2K} \tag{5}$$

Since *K* is usually small and *N* is very large, the rate is approximately 1/3. Employing a puncturing scheme at the output of the convolutional encoders can increase this rate. With puncturing, not all bits are transmitted. For example, the rate can be increased to 1/2 by transmitting one of the outputs of the encoders alternately at each timing interval as depicted in Fig. 3. Other schemes can be employed to achieve different rates.

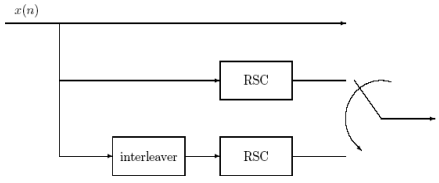


Fig 3: A Turbo encoder with puncturing.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

The performance, however, degrades, in general by increasing the rate via puncturing. For the rate 1/2, a possible puncturing matrix is given by

$$J = \begin{bmatrix} 11 \\ 10 \\ 01 \end{bmatrix} \quad (6)$$

The two columns represent the even and odd transmission periods and the rows represent the three outputs of the Turbo encoder: the information bit, the output of the first encoder and the output of the second encoder. For the matrix in (6) the information bits are sent at all times. However, the output of the upper RSC encoder is sent at even transmission time intervals and the lower encoder at odd times. When Turbo coding is used with BPSK signaling, the performance is very close to Shannon's limit [2, 3]. While BPSK is very simple to implement and its performance is well studied, it is not very bandwidth efficient. Plain BPSK transmits one bit per signaling interval. When Turbo coding is used, the spectral efficiency deteriorates even further. For the Turbo coding structure described in this section, the transmission rate is only1/3 bits per signaling interval. Puncturing a Turbo encoded bit stream and then using BPSK signaling for transmission improves the spectral efficiency slightly. Using the puncturing matrix of Eq. 6, for example, increases the spectral efficiency from 1/3 to 1/2 bits per transmission period. However, the rate will always be less than one bit per period when BPSK is used regardless of the puncturing scheme. In addition, puncturing deteriorates the performance of the Turbo code. Increasing the rate to 4/5 or 5/6, for example, severely degrades the Turbo code performance. While it is possible to use symbol-based (non-binary) Turbo coding directly, and look at all transitions while passing soft information between various states, the decoding becomes exceptionally complex when large interleaver sizes are used, which is key to the good performance of Turbo codes. It also takes prohibitively long times for decoding [6]. In this paper, we develop a Turbo coding scheme which uses QAM for signaling, and, therefore, has better spectral utilization efficiency. It is also considerably less complex than the full-fledged symbol-based non-binary Turbo coding described in [6]. In fact, it is just slightly more complex than the one with BPSK signaling per information bit; albeit, it can be up to an order of magnitude more spectrally efficient. Finally, a method of mapping the binary Turbo code to high-dimension modulations is described.

II. DESIGN OF THE TURBO QAM DECODER

In this section we present a construct which combines both the original Turbo codes of [2] which are well suited for BPSK, and QAM signaling for improved spectral utilization efficiency. In subsection II.1, the encoder design is presented. Also, some techniques in mapping the binary Turbo codes into *M*-ary Turbo codes are presented. Subsection II.2, demonstrates a decoding algorithm for the M-ary Turbo codes. To that end, a few soft mapping functions are proposed to approximate the soft information in the received symbols. Finally, the performance of the proposed QAM Turbo code technique is examined.

III. 1 ENCODER STRUCTURE

The encoder is similar to the standard binary Turbo encoder consisting of two RSC encoders and an interleaver as shown in Fig. 1. It is possible to use bit-wise interleavers or symbol-wise interleavers. A bit-wise interleaver permutes every bit to any position regardless of the modulation technique used. A symbol-wise interleaver on the other hand, operates on k-bit blocks instead of single bits, where $M = 2^k$ is the number of points in the constellation diagram. In other words, every k consecutive bits are treated as a symbol and when interleaved they move together as a block. Each block of k bits from each of the systematic input $x(n)$ as well as similar blocks from the outputs of the two RSC, $y_1(n)$ and $y_2(n)$ are associated with a point in the QAM constellation diagram and a waveform corresponding to the point is used for transmission. The waveforms from the systematic input as well as those from the two RSC's are sent sequentially. The constellation diagram is Gray-coded to reduce bit errors. Fig. 4 is an example of a 16-point Gray- coded constellation.

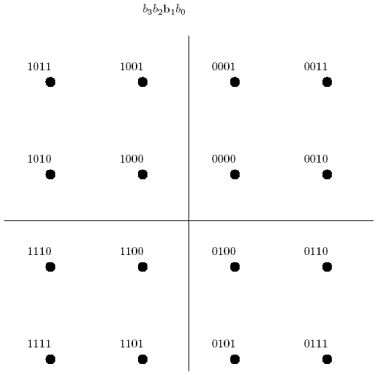


Fig 4: Gray coding for QAM-16 constellation. The scale of the axes is omitted for clarity but the constellation points are at the standard locations {±3, ±1} in both axes.

A subtle difference on the size of the interleaver between Turbo encoders when BPSK and when QAM is used for signaling is worth mentioning. In the BPSK case, the interleaver can be of any size *N*, though large interleavers are usually used because they achieve better performances. When QAM is used for signaling, then the interleaver size must be a multiple of the number of bits in a constellation point. For an *M*− point constellation where $M = 2^k$, the interleaver size *N* must be

$$N = nk = n \log_2 M, \quad (7)$$

for some integer n. Here again, long interleavers in general perform better than short ones

IV. 2 DECODING TURBO QAM

The decoding algorithm of the QAM Turbo codes investigated in this paper starts by decoupling the received symbol corresponding to *k* bits into individual symbols associated with individual bits. These symbols are then sent to a Turbo based MAP decoder or a soft output Viterbi decoder (SOVA) for example. One method to perform the decoupling

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

is to employ a hard decision on the received symbol, then associate *k* symbols to the bits corresponding to the bits of the hard decision constellation point. Doing so, however, loses the soft information of the received symbol. A pragmatic way to do such decoupling, without totally losing the soft information can be explained by looking at an example of a QAM¬16 Gray-coded constellation diagram shown in Fig. 4. Bit b_1 in the figure, for example, is dependent only on the *x* position and is constant along the *y*-axis. Let $r = [r_x, r_y]$ be the received symbol where r_x and r_y are the real and imaginary components respectively. If $r_x > 2$, then b_1 is more probable to be 1 than to be 0 regardless of the value of r_y. The same thing is true when $r_x < -2$. For $-2 < r_x < 2$, the probability of b_1 being 0 is more than that of it being 1.This probability increases as r_x goes to zero from both sides and decreases going away from zero. One can devise a semi-soft measure of these probability likelihood ratios by defining suitable increasing and decreasing functions in the appropriate ranges. The function is chosen to be linear within each interval. An example of such a function is

$$r_1(r) = r_x - 2, \text{ if } r_x > 0$$
$$r_1(r) = -r_x - 2, \text{ if } r_x < 0$$
$$r_1(r) = |r_x| - 2. \quad (8)$$

Where r_1 is associated with the bit b_1. An assumption is made here that the MAP decoder, which will process the r_i , uses the standard mapping where 0 is mapped to -1 and 1 to 1. Figure 5 illustrates the relation of the function $r_1(r)$ and b_1 which has been isolated from the other bits in the constellation symbols.

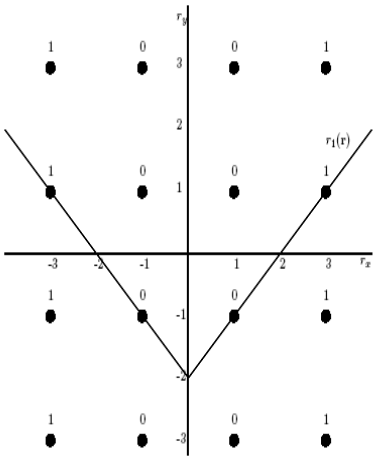


Fig 5: Semi-soft function $r_1(r)$

Similar functions can be devised for the other bits. Here are the other three for QAM-16. They are plotted in Figs. 6 to 8 and are defined as:

$$r_0(r) = |r_y| - 2,$$
$$r_2(r) = -r_y, \text{ and}$$
$$r_3(r) = -r_x \quad (9)$$

where r_0, r_2 and r_3 are associated with bits b_0, b_2 and b_3 respectively. We emphasize that these functions are specific to the Gray code shown in Fig. 4. However, it is trivial to transform these functions to any Gray coding.

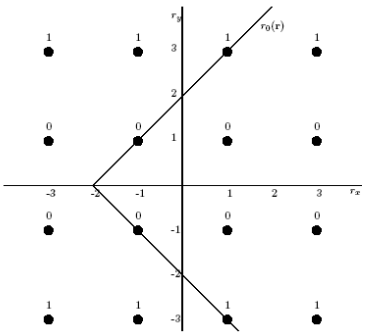


Fig 6: Semi-soft function $r_0(r)$

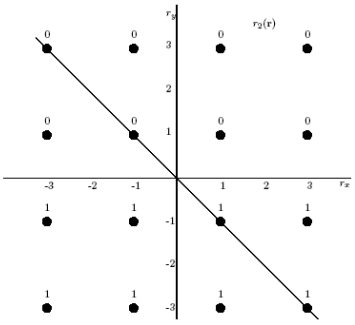


Fig 7: Semi-soft function $r_2(r)$

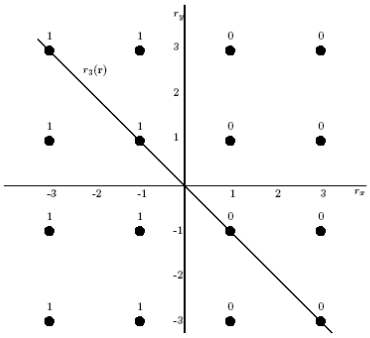


Fig 8: Semi-soft function $r_3(r)$

Using a similar procedure it is possible to decouple bits from square QAM with more points. The semi-soft functions need to be defined appropriately though and might even depend on the signal to noise ratio E_b/N_0. The extension is simpler when the number of points in the constellation is of the form $M = 2^{2n}$ for some integer *n*. When *M* is an odd power of 2, like 32, for example, then the semi-soft functions are slightly complex even if the constellation points are Gray-coded. Below is a description of how that extension can be done for $M = 64$. Let 64-QAM constellation diagram shown in Fig. 9 is used in a communication system which employs Turbo coding. The points in the diagram are Gray-coded to reduce the BER. Similar to the case for QAM-16, we note that the bits in even positions, i.e. b_0,b_2 and b_4 depend only on the *y*- position of the received signal. Similarly, the bits in odd positions depend only on the *x*- value of the received signal. To decouple a received symbol into symbols related to the individual bits, we use similar functions to those used in QAM-¬16. Let $r = [r_x, r_y]$ be the received symbol. Bit *b1* depends on the *x*- position of the received symbol as shown in Table 1.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

r_x	-7	-5	-3	-1	1	3	5	7
b_1	1	0	0	1	1	0	0	1

Table 1: Value of bit b_1 at various points in the constellation diagram for QAM-64 of Fig. 9.

Here again we assume that the BPSK MAP decoder uses the mapping where 0 is mapped to -1 and 1 is mapped to 1. The function $r_1(r)$ is used to estimate the soft output measure for the bit *b1*. That estimate, along with the estimates for the other bits, is sent to the MAP decoder. An approximation of r_1 can be defined by linearly interpolating between the $x-$ location of the constellation points and extrapolating outside the points' range. The function r_1 can be defined as

$$r_1(r) = |r_x - 4| - 2, \text{ if } r_x > 0,$$
$$= |-r_x - 4| - 2, \text{ if } r_x < 0, \quad (10)$$

and this can be written more compactly as

$$r_1(r) = ||r_x| - 4| - 2. \quad (11)$$

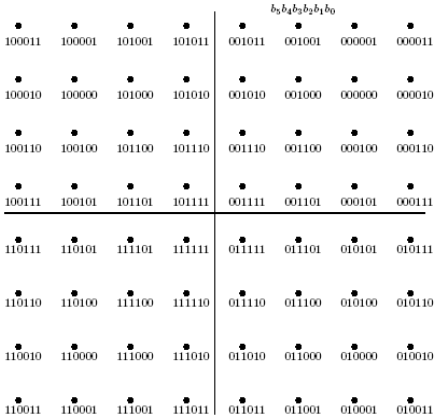


Fig 9: QAM-64 Gray-coded constellation. The scale of the axes is omitted for clarity but the constellation points are at the standard locations {±7, ±5, ±3, ±1} in both axes.

For the other bits, b_0, b_2, b_3, and b_5, appropriate soft-output functions are defined. Tables 2 and 3 show the dependence of these bits on the real (r_x) and imaginary (r_y) components of received symbol *r*.

r_y	-7	-5	-3	-1	1	3	5	7
b_0	1	0	0	1	1	0	0	1
b_2	0	0	1	1	1	1	0	0
b_4	1	1	1	1	0	0	0	0

Table 2: Values of bits b_0, b_2 and b_4 at various points in the constellation diagram for QAM-64 of Fig. 9.

r_x	-7	-5	-3	-1	1	3	5	7
b_3	1	0	0	1	1	0	0	1
b_5	0	0	1	1	1	1	0	0

Table 3: Values of bits b_3 and b_5 at various points in the constellation diagram for QAM-64 of Fig. 9.

The soft functions r_0, r_2, r_3, r_4, and r_5 defined below are used as measures for the soft output which is fed to the MAP decoder. These functions, which are shown in Figs. 10 and 11, correspond to the bits with the same indices so that r_0 corresponds to bit b_0 and r_2 to b_2 and so on.

$$r_0(r) = ||r_y| - 4| - 2, \quad (12a)$$
$$r_2(r) = -(|r_y| - 2), \quad (12b)$$
$$r_3(r) = -(|r_x| - 2), \quad (12c)$$
$$r_4(r) = -r_y, \text{ and} \quad (12d)$$
$$r_5(r) = -r_x. \quad (12e)$$

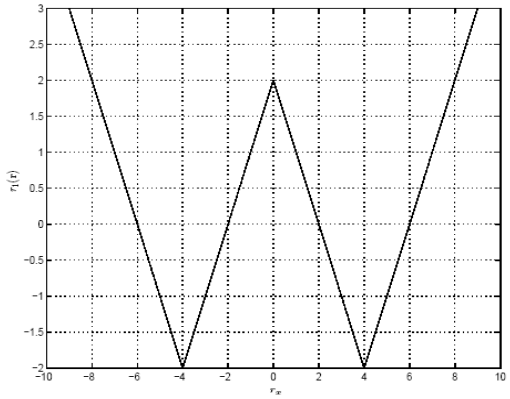


Fig 10: The soft function $r_1(r)$ for QAM-64 for Gray-coded constellation shown in Fig. 9.

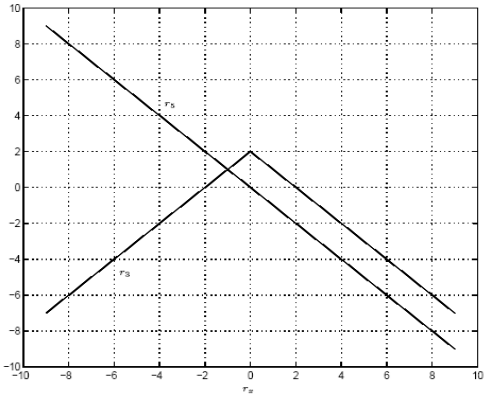


Fig 11: The soft functions $r_3(r)$ and $r_5(r)$ for QAM-64 for Gray-coded constellation shown in Fig. 9.

V. 3 MAPPING TO HIGH-DIMENSIONAL QAM

It is possible to slightly improve the communication system performance simply by choosing the constellation points appropriately. By choosing the constellation points to be closely packed with equal distances to each other in an $n-$ dimensional space such that the sum of the square norms of all the points is fixed, the performance improves compared to "square" packing. However, doing that even in a 2-dimensional space, leaves the points with more than 4 close neighbors. This makes it hard to use Gray coding so that neighboring points differ in only 1 bit in their binary representation, thus greatly improves the BER. A different approach for mapping the constellation points into higher dimensional spaces involves using the constellation points unequally [7, 8]. In this approach, the points with large norms, i.e., the ones carrying more energy are used less frequently than the ones with low norms. In a typical square constellation, for example, the outer points will be used less frequently than the inner points. To achieve that, though, the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

constellation points carry a variable number of bits, as opposed to the equi-probable usage of the constellation points in which each point carry the same number of bits. When the constellation points are chosen according to the continuous Gaussian distribution in each dimension, then the maximum gain is possible. The gain, called the shaping gain, is about 1.53 dB when infinite dimensions are used. However, for finite dimensions and when discrete quantization of the Gaussian distribution is used, only modest gain is achieved. In addition, this method achieves the gain at the expense of reducing the entropy, and hence, the data rate. Moreover, the technique adds considerable complexity to the system.

VI. SIMULATION RESULTS

In this paper, we propose using multiple square QAM constellations each of which uses 2 dimensions. Then, in each QAM we use the soft functions presented above. While in using this method we forego the modest shaping gain, we acquire more bit rates and reduced complexity. The performance of the Turbo code combined with QAM signaling was estimated using Monte Carlo simulation. The system was setup as presented in section 2 which is summarized as follows. A uniform Bernoulli number generator is setup to produce "independent" binary outputs. These outputs represent the information bits to be transmitted. No source encoder is used because each bit has a probability of 1/2 of being either 0 or 1. Since the entropy of the source is 1 bit per generated outcome, it is not possible to compress the data and hence, adding a source encoder simply adds to the complexity of the simulated system without improving on the capacity usage. In each iteration, a total of N bits are generated where N is the size of the interleaver to be used in the Turbo encoder. The N bits are Turbo encoded using the two component RSC's. These results in three sets of N bits: the uncoded bits, the output of the first encoder and the output of the second encoder, which are passed through an inter-leaver before being RSC encoded as shown in the Fig. 12.

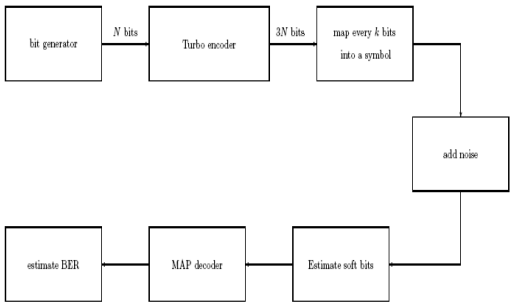


Fig 12: Combined Turbo coded QAM simulation setup.

In addition to the $3N$ bits, there are a few bits which results from flushing the RSC with zeros to force one of them to go to the all zero state. Each k bits from the source generator as well as from the outputs of the two RSC's are grouped together and mapped to a point in an $M = 2^k$ point constellation diagram. Noise, simulated by a Gaussian random number generator, is then added to the points. The variance of the additive noise is varied to simulate various signal-to-noise ratios. Soft outputs of the component bits are estimated from the noise corrupted points using the soft functions described in subsection II.2. These in turn are passed to a MAP decoder which estimates the transmitted bits. Finally, the decoded bits are compared to the transmitted bits and the BER is estimated. Fig. 13 shows the performance in terms of BER for the Turbo code combined with QAM signaling using an interleaver of size 128 without any puncturing. The performance is contrasted to that of an uncoded QAM-16 system and a rate 7/15 Reed-Solomon (RS) code with hard decision decoding in GF(24). The figure demonstrates a noticeable performance gain over both the uncoded and the RS coded system. We see that there is about 6 dB gain when using the Turbo code when combined with QAM signaling compared to the RS code at BER of 10^{-5}.

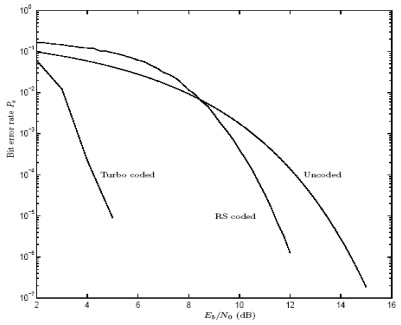


Fig 13: Simulated BER using QAM-16 with soft output Turbo codes with N = 128.

Also shown are the uncoded QAM performance and a rate 7/15 RS coded QAM-16. It is assumed that Gray coding is used in all cases. Similar gains are also observed when using higher constellation sizes. Fig. 14 shows the performance using QAM-256 modulation in conjunction with Turbo coding. The figure also demonstrates the gain over a RS code of rate 127/255 in GF(28) with hard decision decoding.

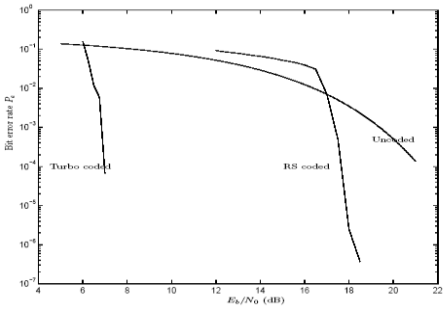


Fig 14: Simulated BER using QAM-256 with soft output Turbo codes, N = 2048.

Also shown are the uncoded QAM performance and a rate 127/255 RS coded QAM-256 on GF (28) with hard decision decoding. It is assumed that Gray coding is used in all cases. Fig. 15 shows the performance of the method developed in this paper (point F) with respect to Shannon's capacity limit at BER of 10^{-5}. Also shown in the figure are the operating points

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

of other modulation and coding techniques. Point (A) represents the performance of binary rate 1/2 Turbo code. Points (B) and (C) represent uncoded BPSK and QPSK respectively. Point (D) is the operating point of a parallel concatenated trellis coded modulation using PSK-8 for modulation [8]. Finally, point (E) represents the operating point of a 64-state TCM using PSK-8 for modulation [6]. It can be seen from the figure that the schemes of points (D) and (E) are closer to the Shannon's limit than the scheme of point (F) in terms of the required SNR to achieve the given BER performance. However, the latter is more spectral efficient.

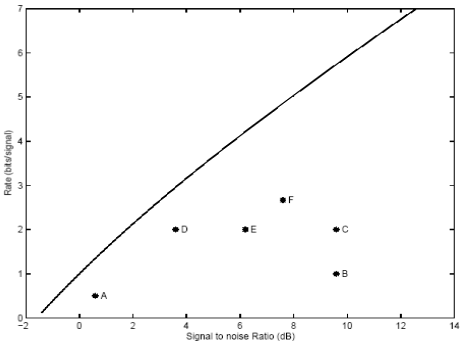


Fig 15: Performance of QAM-256 with Turbo code using semi-soft functions with respect to Shannon's limit (F). Other points are: (A) binary rate 1/2 Turbo code, (B and C) uncoded BPSK and QPSK, (D) PC-TCM on 8PSK and (E) 64-state TCM on PSK8.

VII. CONCLUSIONS

Turbo codes have established themselves as very powerful error correcting codes over AWGN channels. They can also be used over Rayleigh fading channels, albeit, they become more powerful when combined with other techniques to combat fading. In this paper, we have developed a few soft mapping functions to approximate the soft information in the received symbols. These in turn are passed to a standard binary Turbo decoder like the MAP decoder or the Soft-Output Viterbi Decoder (SOVA). This method shows substantial gains over other error correcting code methods while using the available bandwidth more efficiently. Computer simulations showed as much as 6 dB gain over a Reed Solomon (RS) code of approximately the same rate and using the same modulation for an error rate of 10^{-5}. The gains, however, come at an increased computational cost, especially for large interleaver sizes.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," Bell Sys. Tech. J., vol. 27, pp.379–423, October 1948.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," Proc. IEEE Int. Communications Conf., pp.1064–1070, May 1993.

[3] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," IEEE Trans. Communications, vol. 44, pp.1261–1271, May 1996.

[4] J. Proakis, Digital Communication, McGraw-Hill, New York, NY, 4th edition, 2000.

[5] D. J. Costello, J. Hagenauer, H. Imai, and S. B.Wicker "Applications of error-control coding," IEEE Trans. Information Theory, vol. 44, pp.2513–2560, October 1998.

[6] L. Hanzo, T. H. Liew, and B. L. Yeap, Turbo Coding, Turbo Equalization and Space-Time Coding, John Wiley and Sons, Chichester, West Sussex, England, 2002.

[7] S. Le Goff, A. Glavieux, and C. Berro, "Turbo codes and high spectral efficiency modulation," pp. 645–649, 1994.

[8] D. Raphaeli, and A. Gurevitz, "A new high performance turbo encoder scheme for bandwidth efficient modulation," The 22nd Convention of Electrical and Electronics Engineers, Israel. vol. 1, pp. 1019–1022, December 2002.

[9] D. Divsalar, and F. Pollara, "Turbo trellis coded modulation with iterative decoding for mobile satellite communications,". IMSC97, 1997.

AUTHOR BIOGRAPHY

Amer Al Habsi, he got his PhD from University of Arkansas at Little Rock, Arkansas, USA, 200, MS, from Northeastern University, Boston, Massachusetts, USA, 1998, and BS, Biomedical Engineering, Case Western Reserve University, Cleveland, OH, USA, 1994. His research Interest, Signal Processing, Communication Theory, Channel Coding. His related Experience, Supervisor, Petroleum Development Oman (PDO), Fahud, 1993-1994. He is one of the Professional Membership in IEEE Communication Society.

Yahiea Al-Naiemy was born on July 3, 1971 in Diyala, Iraq. In 1990, he was enrolled at the Higher Institution of Telecommunications and Post before transferring to Al-Mustansiriyah University/College of Engineering - Electrical Engineering Department in 1994. In July of 1998 he received a Bachelor's a Degree in Electrical Engineering. He continued his graduate studies by joining the Iraqi Commission for Computers and Informatics where he received a Higher Diploma in Information Systems in 2001. He enrolled at Diyala University as an instructor in electrical power, computer and physics departments. In 2009 he was granted a scholarship to complete his master degree in electrical engineering at the University of Arkansas at Little Rock, USA. He got his MSc in Wireless Communications from UALR, USA, in 2012. While completing his graduate degree, his research effort has been in the area of antennas and microwave material characterization. His current research areas include photonic band gap (PBG) structures, metamaterial, GPS, implantable wireless systems, and nanoscale microwave devices. Know, he is one of the a Member of IEEE and reviewer in PIER Journals.

Dr.Hussain Al-Rizzo received his B.Sc. in Electronics and Communications (1979) (High Honors), Postgraduate Diploma in Electronics and Communications (1981) (High Honors) and M.Sc. in Microwave Communication Systems (1983) (High Honors) from the University of Mosul, Mosul, Iraq. From May 1983 to October 1987 he was working with the Electromagnetic Wave Propagation Department, Space and Astronomy Research Center, Scientific Research Council, Baghdad, Iraq Baghdad, Iraq.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 8, February 2014

On December, 1987, he joined the Radiating Systems Research Laboratory, Electrical and Computer Engineering Department, University of New Brunswick, Fredericton, NB, Canada where he obtained his Ph.D. (1992) in Computational Electromagnetic, Wireless Communications, and the Global Positioning System. For his various academic achievements he won the nomination by the University of New Brunswick as the best doctoral Graduate in science and engineering. Since 2000, he joined the Systems Engineering Department, University Arkansas at Little Rock where he is currently a tenured Professor. He has published over 40 peer-reviewed journal papers, 70 conference presentations, and several patents. His research areas include implantable antennas and wireless systems, smart antennas, WLAN deployment and load balancing, electromagnetic wave scattering by complex objects, design, modeling and testing of high-power microwave applicators, design and analysis of micro strip antennas for mobile radio systems, precipitation effects on terrestrial and satellite frequency re-use communication systems, field operation of NAVSTAR GPS receivers, data processing, and accuracy assessment, effects of the ionosphere, troposphere and multipath on code and carrier-beat phase GPS observations and the development of novel hybrid Cartesian/cylindrical FD-TD models for passive microwave components.