Low Complexity Non-Binary LDPC and Modulation Schemes Communicating over MIMO Channels

Feng Guo and Lajos Hanzo[1]

School of ECS, University of Southampton, SO17 1BJ, UK.
Tel: +44-23-8059-3125, Fax: +44-23-8059-4508
Email:lh[1]@ecs.soton.ac.uk, http://www-mobile.ecs.soton.ac.uk

Abstract – **In this contribution, Pearl's *belief propagation* (BP) algorithm is invoked for constructing a belief network, which is employed for developing a joint detection aided transmit diversity scheme and a non-binary LDPC decoder constructed over a finite field of GF(q). An exciting bit-by-bit detection scheme is further developed for creating the joint purly symbol-based LDPC/transmit diversity decoder. The performance of the proposed symbol-based system is benchmarked against the original bit-by-bit detection scheme, when communicating over an uncorrelated Rayleigh fading channel using two transmitters and two receivers. The associated detection complexities are also compared .**

1. INTRODUCTION AND SYSTEM SCHEMATIC

1.1. Motivation and State-of-the-Art

Since the invention of turbo codes by Berrou [1] *et. al*, the superior performance of iterative decoders has attracted substantial research interest. The family of Low Density Parity Check (LDPC) codes originally devised by Gallager as early as 1963 [2] has also been extensively studied during the 1990s [3, 4]. More recently, Mackay, McEliece and Cheng [5] pointed out that Gallager's *probabilistic LDPC decoding* algorithm [6], constitutes an instance of Pearl's belief propagation algorithm. Mackay and Neal demonstrated in [7] that despite their simple decoding structure, LDPC codes are also capable of operating near the channel capacity. Richardson [8] suggested the employment of a differential belief propagation decoding algorithm for binary LDPC codes using the Fast Fourier Transform (FFT) for reducing the decoding complexity imposed. In 1998, Davey and Mackay proposed a non-binary version of LDPC codes [9], which was potentially capable of outperforming binary LDPC codes. When using Richardson's FFT-based decoding algorithm [8], the complexity of non-binary LDPCs increases only linearly with respect to the size of the associated Galois field.

These non-binary LDPC codes may be conveniently combined with multilevel modulation and/or multiple antenna schemes, which are capable of supporting high data rate transmissions [10]. In this contribution, an LDPC-coded low-complexity two-transmitter and two-receiver scheme will be studied.

This contribution is structured as follows. In Section 2, a brief overview of Baysian networks [11] and Pearl's *belief propagation* algorithm [11] will be given. Section 3 introduces non-binary LDPC decoding over GF(q). Section 4 details the symbol-by-symbol joint detection algorithm advocated. Section 5 characterises the achievable performance of the proposed system in contrast to the corresponding bit-by-bit detection scheme proposed by Meshkat and Jafarkhani [12]. The proposed system will be characterized using 4QAM, 8PSK and 16QAM transmission schemes having a throughput of 2, 3 and 4 bits per symbol (BPS). Finally, Section 6 will provide a complexity comparison, while Section 7 offers our conclusions.

1.2. System Schematic

The overall system schematic is illustrated in Figure 1. The source bits are encoded by a non-binary LDPC encoder, and the LDPC encoded symbols are mapped to the corresponding QAM symbols, which are transmitted by the multiple antenna based transmit diversity system. The channel's output signal r is then fed into the demodulator where the soft-metric $\mathcal{M}_{r->v}(s_k)$ is calculated. More explicitly, the notation $\mathcal{M}_{r->v}(s_k)$ represents the soft metric passed from the demodulator to the LDPC decoder based on the symbol probability of the kth decoded symbol of the LDPC codeword. As seen in Figure 1, these soft-metrics are passed to the LDPC decoder, which carries out an LDPC iteration and produces the resultant *a posteriori probability* P_{post}. Based on the *a posteriori probability*, a tentative hard decision will be made and the resultant codeword will be check by the LPDC code's parity check matrix. If the resultant vector is an all-zero sequence, then a legitimate codeword has been found, and the hard-decision based sequence will be output. Otherwise, if the maximum number of iterations has not been reached, the *a posteriori probability* will be subtracted from the original soft metric denoted by $\mathcal{M}_{r->v}(s_k)$ and fed back to the demodulator for the next iteration, as seen in Figure 1. This process continues until the pre-defined maximum number of iterations has been encountered or a legitimate codeword has been found.
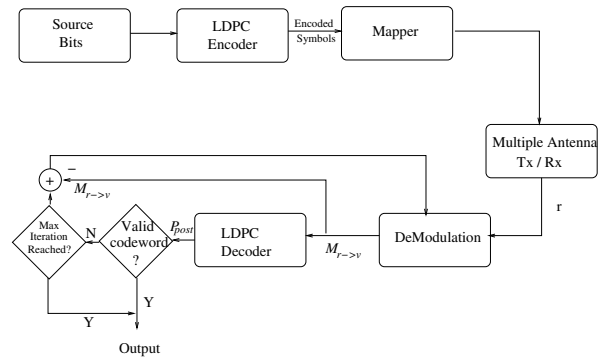

Figure 1: System schematic

1294

2. BAYSIAN NETWORKS AND BELIEF PROPAGATION

In this section, we provide a rudimentary introduction to both Baysian networks [11] and to the classic belief propagation algorithm [11].
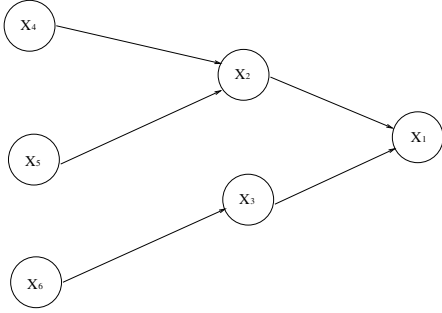

Figure 2: Baysian network having six nodes

Let us illustrate the associated concepts with the aid of an example. In Figure 2, we assume that X_4, X_5 and X_6 are the so-called evidence or observation nodes and we would like to infer the *a posteriori* probability of X_1. Hence by assuming that $P(X_4)$, $P(X_5)$ as well as $P(X_6)$ are known and given that the relationship of the various variables is characterized by the directed links seen in Figure 2, the *a posteriori* probability of X_1 can be represented as follows:

$$P(X_1) = P(X_1|X_2, X_3) \cdot P(X_2|X_4, X_5) \cdot P(X_3|X_6)$$
$$\cdot P(X_4) \cdot P(X_5) \cdot P(X_6). \quad (1)$$

If we use the notation $P_p(X_i)$ for representing the parent node set of node i, Equation 1 may be rewritten as:

$$P(X_1) = \prod_{i=1}^{N} P(x_i|P_p(x_i)). \quad (2)$$

In general a Baysian network consists of a set of random variables denoted by $X = \{X_1, X_2...X_n\}$ and these random variables are represented by the nodes of the network. Between the nodes, there are directed links representing the relationship of the so-called *parent nodes* with the so-called *child nodes*. Some of the nodes may correspond to random variables, whose values are encountered and hence observed, which are the previously mentioned "evidence nodes" or "observation nodes" [12]. When a specific set of variables corresponding to the evidence nodes is observed, the belief propagation algorithm may be invoked for inferring the *a posteriori* information corresponding to the rest of the nodes in the network, as it was demonstrated in the context of Figure 2.

Since LDPC codes may be conveniently characterized by the variable nodes, check nodes and their inter-connections [13], a corresponding Baysian network can be constructed. Furthermore, in this contribution, the multiple antenna aided transmit diversity scheme may also be represented as a Baysian network. Hence Section 4 will detail the process of developing a tri-partite Baysian network for the system proposed where the associated belief propagation algorithm invoked for this tri-partite Baysian network will also be described.

3. NON BINARY LDPC

Gallager's original binary LDPC codes are defined by a sparse parity check matrix (PCM) having a relatively low fraction of non-zero parity check entries in the matrix. A codeword is a legitimate one, if its product with the PCM using modulo-2 multiplications and additions is an all-zero vector. Davey and Mackay generalized the family of binary LDPC codes for a Galois Field of size q, i.e. for GF(q). These non-binary LDPC codes are defined by a similar sparse PCM, with the exception that the non-zero entries may now assume any integer value between 1 and $(q - 1)$. Thus the product of a legitimate codeword with the PCM calculated over GF(q) is an all zero vector, which is expressed as:

$$\sum_{j \in \mathcal{N}(m)} H_{mj}(x_j) = 0, \text{ over GF(q)}, \quad (3)$$

where $\mathcal{N}(m)$ is a set containing all the column indices of the non-zero entries in the mth row, and the variable H_{mj} represents the value of the non-zero entry in the mth row and j^{th} column of the PCM, which ranges from 1 to $(q - 1)$. Furthermore, x_j represents the surviving state of the jth symbol of the codeword, which is defined as the symbol state having the highest *a posteriori* probability among the q number of legitimate states. Since the non-zero entries in the PCM are elements of GF(q), they may also be represented as binary strings constituted by p bits, where $2^p = q$. In Equation 3 the variable x_j also assumes values ranging from 0 to $q - 1$, thus it may be represented by a binary bit string of size $p = log_2 q$. It has been shown [14] that an $M \times N$ non-binary PCM constructed over GF(q) has an equivalent binary PCM of size $Mp \times Np$ [14]. It has been shown in [15] that LDPC codes have to have a high column weight for the sake of achieving a good performance. However, a high column weight will introduce more short cycles in the PCM and this in turn degrades the achievable performance. The advantage of using non-binary LDPC codes is that the equivalent binary weight of the PCM is increased, while the number of short cycles may remain low [14].

As in binary LDPC codes, the non-zero entries of the PCM iteratively update the quantities R_{mn}^a and Q_{mn}^a, where R_{mn}^a represents the probability of the mth check being satisfied, when symbol n of the codeword is considered to be in state $a \in GF(q)$ and the other symbols have a separable distribution given by the probabilities $\{Q_{mn'}^b : n' \in \mathcal{N}(m)\backslash n, b \in GF(q)\}$. The calculation of probability R_{mn}^a is formulated as :

$$R_{m,n}^a = \sum_{\mathbf{x}:x_n=b} P(z_m = 0|\mathbf{x}) \prod_{k \in \mathcal{N}(m), k \neq n} Q_{m_k}^{x_k}. \quad (4)$$

More explicitly, this implies that the quantity R_{mn}^a represents the Probability Density Function (PDF) of the nth symbol being in any of the q number of legitimate states, given the knowledge of each individual PDF for the rest of the symbols participating in the mth check, and given that the mth check is satisfied.

Furthermore, the quantity Q_{mn}^a denotes the probability of the nth symbol being in the state $a \in GF(q)$ calculated from the probabilities $\{R_{m'n}^a : m' \in \mathcal{M}(n)\backslash m\}$ according to:

$$Q_{m,n}^a = \alpha_{m,n} f_n^a \prod_{k \in \mathcal{M}(n), k \neq m} R_{k,n}^a. \quad (5)$$

More explicitly, the quantity Q_{mn}^a is given by multiplying the *intrinsic probability* f_n^a by the product of probabilities R_{mn}^a provided by all check nodes except the mth, indicating that the nth symbol is in state a. Here $\mathcal{N}(i)$ and $\mathcal{M}(i)$ represented a set of column and row indices of the non-zero entries in the ith row and column of the PCM, respectively. The quantity R_{mn}^a and Q_{mn}^a are updated as follows.

After each iteration, the *a posteriori* probability P_n^a will be calculated based on the *intrinsic probability* f_n^a of the nth received sample and on the information updated as well as delivered by R_{mn}^a according to:

$$P_n^a = \alpha_n f_n^a \prod_{k \in \mathcal{M}(n)} R_{k,n}^a. \tag{6}$$

The quantity α in Equations 5 and 6 represents a normalization factor required for ensuring that the conditions $\sum_{a \in GF(q)} Q_{mn}^a = 1$ and $\sum_{a \in GF(q)} P_n^a = 1$ are satisfied. In Equation 4, the term $P(z_m = 0|\mathbf{x})$ is acting as a binary flag, which returns a value of 1, if the current codeword $\mathbf{x}$ satisfies the mth check, or 0 otherwise. More explicitly, this expression has to be evaluated for all legitimate manifestations of the codeword $\mathbf{x}$, which becomes computationally prohibitive, when the blocklength is high. However, since the operation of updating the quantity R_{mn}^a in Equation 4 corresponds to calculating the joint probability of all symbols within $\mathcal{N}(m)$ over GF(q) given in the form of the corresponding sum multiplied by their corresponding matrix entry, i.e. by the probability of

$$P(\sum_{\mathbf{x}:x_n=b} x_n \cdot H_{mn} = a), a \in GF(q), \tag{7}$$

thus a multi-dimensional FFT [14] over GF(q) may be invoked for the sake of reducing the complexity imposed.

4. JOINT DETECTION SYSTEM

In this section, we will invoke the Baysian network of Section 2 for characterising the LDPC decoder amalgamated with a transmit diversity scheme originally proposed by Meshkat and Jafarkhani, which was detected using a bit-by-bit decoding algorithm [12].
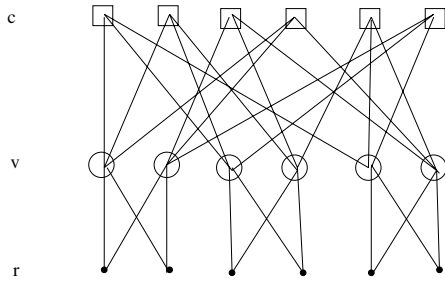

Figure 3: Tri-partite graph of the jointly detected transmit diversity system using LDPC coding

In Figure 3 the squares represent the check nodes of the LDPC code, while the circles correspond to its variable nodes. The black dots represents the channel outputs. The connections between the squares and the circles represent the non-zero entries within the PCM. After the LDPC encoding process, each variable node contains an encoded LDPC symbol constituted by $p = log_2 q$ bits. Each pair of symbols is transmitted by the two transmitters and received by the two receiver antennas through four propagation paths. Thus the lines connecting the circles and black dots represent the two-transmitter, two-receiver system. In the butterfly-shaped section of Figure 3 each pair of the channel outputs is correlated, since they originate from the same pair of modulated symbols, which are transmitted through four different propagation paths by the two-antenna system. The message passing scheme of the joint decoder may be described as follows. In the associated equations, we will use r, v and c for representing the received channel output, the variable nodes and the check nodes, respectively.

Following the approach of [12], initially the jth soft channel output is passed from the black dots in Figure 3 to the variable nodes

(circles) of the LDPC decoder and the associated confidence measures are calculated by comparing the soft channel output values to all legitimate transmitted symbols according to [12] :

$$\mathcal{M}_{r->v}(b_k) = \sum_{all\ B_i} \frac{1}{(\sqrt{2\pi}\sigma_n)^{n_r}} e^{-\frac{|\mathbf{r}-\mathbf{HS}(B_i,b_k)|^2}{2\sigma_n^2}} \cdot$$
$$\prod_{b_j \in B_i} \mathcal{M}_{v->r}(b_j). \tag{8}$$

In Equation 8, σ_n represents the standard deviation of the Gaussian noise, $\mathbf{H}$ is an $n_r \times n_t$ matrix containing the complex valued fading coefficients of each transmission path, where n_r and n_t are the number of receiver and transmitter antennas, respectively. Using bps representing the number of bits per symbol for the corresponding modulation scheme, B_i represents the set of $(n_t \times bps) - 1$ transmitted bits, but excludes the kth bit b_k, which contributes to the value of the received vector $\mathbf{r}$ at the n_r number of receivers, while $\mathbf{S}(B_i, b_k)$ is a vector of n_t components containing the symbols corresponding to the bit set of B_i and b_k. More explicitly, let us consider Figure 4, for example. When using QPSK modulation, four bits are mapped into two QPSK symbols and transmitted by the two transmitters to the two receivers. If $\mathcal{M}_{r->v}(b_2)$ is under consideration, then we have $B_i = \{b_0, b_1, b_3\}$ and $(B_i, b_2) = \{b_0, b_1, b_2, b_3\}$. The vector $\mathbf{r}$ will contain elements of $\{r_0, r_1\}$. For a particular set of $B_i = \{b_0 = 1, b_1 = 1, b_3 = 0\}$, since QPSK modulation is employed in Figure 4, thus the notation $\mathbf{S}(B_i, b_2)$ represents $\{11, 00\}$ or $\{11, 10\}$, depending on the specific value of b_2 concerned.
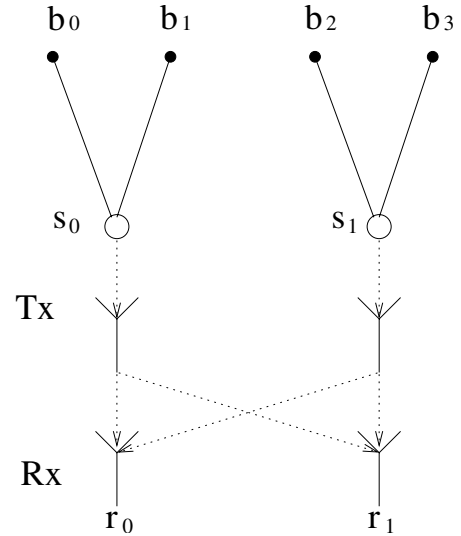

Figure 4: Schematic of a two-transmitter two-receiver system using QPSK modulation

The metric $\mathcal{M}_{r->v}(b_k)$ is then passed to the variable nodes, and the following message is calculated at the variable nodes [12]:

$$\mathcal{M}_{v_k->c_i}(b_k) = \alpha_k \mathcal{M}_{r->v}(b_k) \prod_{j \in \mathcal{M}(k), j \neq i} \mathcal{M}_{c_j->v_k}(b_k), \tag{9}$$

which is then passed onto the check nodes. The structure of Equation 9 is similar to the update formula of Q_{mn}^a seen in Equation 5, with the exception that the constant *intrinsic information* f_n^a in Equation 5 has been replaced by $\mathcal{M}_{r->v}(b_k)$, as formulated in Equation 8.

The metric $\mathcal{M}_{c_j->v_k}(b_k)$ of Equation 9 corresponds to R_{kn}^a in Equation 5, which is then passed from the check nodes to the variable nodes, and updated according to Equation 4. Then the message $\mathcal{M}_{v->r}(b_k)$ has to be passed from the variable node back to the channel output, which is formulated as:

$$\mathcal{M}_{v->r}(b_k) = \alpha \prod_{j \in \mathcal{M}(k)} \mathcal{M}_{c_j->v_k}(b_k). \quad (10)$$

This operation is similar to Equation 6, however the *instrinsic information* f_n^a seen in Equation 6 is omitted from Equation 10 and the metric $\mathcal{M}_{v->r}(b_k)$ is fed back to the channel output nodes from the variable nodes. More explicitly, the reason that the *intrinsic information* is omitted in this case is, because this information, which was denoted by $\mathcal{M}_{r->v}(b_k)$ in Equation 9 has been used during the calculation of the metric $\mathcal{M}_{v_k->c_i}(b_k)$ and thus it should be excluded, when providing *extrinsic* information for the channel output nodes.

Again, the above-mentioned procedure was proposed by Meshkat and Jafarkhani in [12] for a binary system. In Equation 8, the *a priori information* is calculated on a bit-by-bit basis, assuming that the bits are independent of each other. However, this assumption is only approximately valid in Gray-coded non-binary modulation schemes. Hence it is beneficial to combine non-binary QAM schemes with matching non-binary LDPC codes, since this process requires no symbol to bit probability conversion. Furthermore, we will show in Section 6 that upon using a purly symbol based joint decoding technique, the associated decoding complexity may be significantly reduced. Hence the bit-by-bit based metric update formula of Equation 8 is converted to its symbol-based counterpart as:

$$\mathcal{M}_{r->v}(s_k) = \sum_{\text{all } S_i} \frac{1}{(\sqrt{2\pi}\sigma_n)^{n_r}} e^{-\frac{|\mathbf{r}-\mathbf{HS}(S_i,s_k)|^2}{2\sigma_n^2}} \cdot \prod_{s_j \in S_i} \mathcal{M}_{v->r}(s_j). \quad (11)$$

In Equation 11, S_i now represents a set of $(n_t - 1)$ number of symbols rather than $(n_t \times bps - 1)$ number of bits, including all symbols, except for s_k, which contributes to the value of the received vector $\mathbf{r}$ at the n_r receivers, and $\mathbf{S}(S_i, s_k)$ is a vector of size n_t containing the symbols including s_0 and s_1, as in Figure 4. More explicitly, rather than using the bits representing the symbols in Figure 4 for the calculation of the *a priori information*, the symbols are employed directly in this scheme. Hence, if symbol s_1 of Figure 4 is concerned, the vector S_i will have only one element of $\{s_0\}$. The *a priori information* $\mathcal{M}_{v->r}(s_j)$ represents the probability of the jth symbol, rather than that of the bits in Equation 8.

5. SIMULATION RESULTS

In this section, the achievable performance of the proposed system is characterized. All simulation parameters are listed in Table 1. We used 4QAM, 8PSK and 16QAM transmission for the sake of increasing the throughput of the system. Correspondingly, non-binary LDPC codes operating over GF(4), GF(8) and GF(16) were used. Since the number of transmitters was set to two and the LDPC code had a coding rate of 1/2, the effective throughput was 2, 3 and 4 bits/symbol, for the corresponding configurations. More explicitly, the effective throughput of the modems was not reduced by the 1/2-rate LDPC codec, because the doubled number of bits was conveyed by two transmit antennas. A coded blocklength of 1500 bits was used.

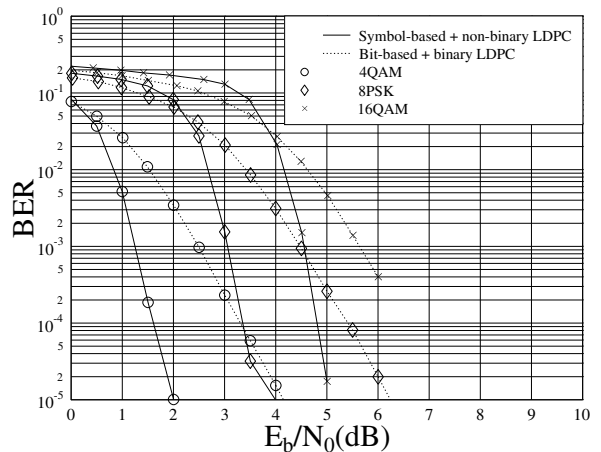| No. of Transmitters	2
No. of Receivers	2
Max No. of Joint Decoding Iterations	5
Channel	Uncorrelated Rayleigh Fading
Average LDPC column weight	2.5
LDPC coding rate	0.5
LDPC decoding field	GF(4), GF(8), GF(16)
Modulation schemes	4QAM, 8PSK, 16QAM
System throughput	2, 3, 4 bits/symbol
Coded Blocklength	1500 bits

Table 1: System simulation parameters

Figure 5: Performance of the symbol-based joint decoding aided system using non-binary LDPC codes for communicating over uncorrelated Rayleigh fading channels. The performance of the bit-based joint decoding system of [12] using a 1/2-rate binary LDPC code was also plotted as a benchmarker.

In Figure 5 it may be observed that by using the proposed symbol based joint decoding algorithm, a better bit error ratio (BER) performance is achieved than that of the original bit-based algorithm. An approximately 2dB gain was achieved at a BER of 10^{-5} for all of the three modulation schemes used.

6. COMPLEXITY

In this section the complexity of the proposed system is estimated. When the number of bits per symbol is increased, calculating the *a priori probability* using the bit-by-bit approach of Equation 8 will become quite complex. Using non-binary LDPCs will also increase the decoding complexity, however, this increase is only linearly dependent on the number of bits, if the multi-dimensional FFT based LDPC decoder of [14] is used.

As seen from Equation 8, the decoding operations require the evaluation of all possible input symbol configurations containing the kth bit of the original codeword, as well as the calculation of the *a priori probability* provided by the neighbouring bits. Hence, for a

*	Bit-based	Symbol-based
4QAM	179	91
8PSK	739	198
16QAM	3338	518
+	Bit-based	Symbol-based
4QAM	27	28
8PSK	75	61
16QAM	267	144

Table 2: Complexity comparison between the bit-based benchmarker algorithm of [12] and the proposed symbol-based algorithm using the simulation parameters listed in Table 1.
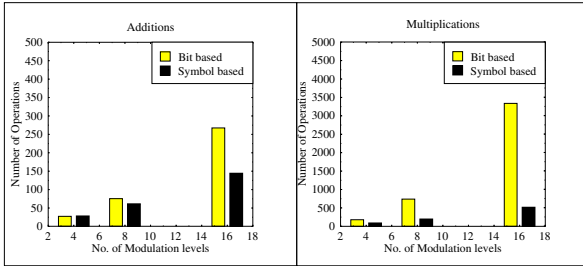

Figure 6: Complexity comparison between the bit-based benchmarker algorithm of [12] and the proposed symbol-based algorithm using the simulation parameters listed in Table 1.

system having n_t transmitters and bps number of bits per symbol, the total number of metric evaluations using Equation 8 will be $2^{bps \cdot n_r}$. Each metric evaluation requires $n_t \times n_r$ number of multiplications for determining $\mathbf{HS}(B_i, b_k)$ in Equation 8, one multiplication for evaluating the square, and one for carrying out the required division. One substraction is needed for finding the Euclidean distance between the received sample and each of the constellation points. Furthermore, for each metric evaluation, $bps \times n_t - 1$ number of multiplications are needed for calculating the $a\ priori\ probability$. Thus, for each decoded bit, the required number of multiplications becomes $((bps \times n_t - 1) + (n_t \times n_r + 2)) \times 2^{bps \times n_t} = (n_t \times (bps + n_r) + 1) \times 2^{bps \times n_t}$. The required number of additions is $2^{bps \times n_t}$. By contrast, for the proposed non-binary system using Equation 11, the number of multiplications needed for the $a\ priori\ probability$ calculation is reduced to $(n_t - 1)$, since we are directly determining the symbol probability. Thus the overall number of multiplications per bit for the symbol based system is $(n_t \times (n_r + 1) + 1) \times 2^{bps \times n_t}/bps$ and the number of additions is $2^{bps \times n_t}/bps$ per bit.

On the other hand, upon employing non-binary LDPC codes, the message passing between the system components becomes more complex owing to the increased GF size. The number of multiplications and additions required for each decoding bit can be represented as $7tq/\log_2(q)$ and $2tq$, where t and q are the LDPC code's average column weight and GF size, respectively [14]. Hence the overall complexity required by the two systems in each of the modulation schemes is listed in Table 2, and plotted in Figure 6.

From Table 2 and Figure 6, it can be observed that by using the proposed symbol-by-symbol joint decoding algorithm, the complexity may be significantly reduced. For example, in case of 16QAM, a complexity reduction in excess of 80 percent was achieved. This complexity reduction becomes even higher for 64QAM. For the sake of simplicity in this comparison, the complexity increase imposed by the extra finite field multiplication during the codeword validation was ignored.

7. CONCLUSION

In this contribution, a novel symbol based joint detection algorithm was proposed for non-binary LDPC-coded transmit diversity-aided transmissions. The proposed purely symbol-based system was capable of achieving a gain of approximately 2dB for QPSK, 8PSK and 16QAM transmissions in comparison to the identical-throughput bit-by-bit benchmarker algorithm. The scheme advocated was also shown to be less compolex than its binary benchmarker.

8. REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes," in *Proceedings of the IEEE International Confrence on Communications*, pp. 1064–1070, 1993.

[2] R. Gallager, "Low Density Parity Check Codes," *IEEE Transaction on Information Theory*, vol. 8, pp. 21–28, Jan. 1962.

[3] S. T. Brink, G. Framer, A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Transaction on Communications*, vol. 52, pp. 670–678, April 2004.

[4] B. Lu, G Yue and X. Wang, "Performance analysis and design optimization of LDPC-coded MIMO OFDM systems," *IEEE Transaction on Signal Processing*, vol. 52, pp. 348 – 361, Feb. 2004.

[5] R. J. McEliece, D. J. C. MacKay, J. F. Cheng, "Turbo Decoding as an Instance of Pearl's Belief Propagation Algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 140–152, Feb. 1998.

[6] R. Gallager, "Low Density Parity Check Codes," *Ph.D thesis,M.I.T,USA*, 1963.

[7] D. J. C MacKay, and R. M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes," *Electronics Letters*, vol. 33, pp. 457–458, 13 March 1997.

[8] T. Richardson, R. Urbanke, "The Capacity of Low-Density Parity Check Codes Under Message-Passing Decoding," *IEEE Transaction on Information Theory*, vol. 47, pp. 599–618, Feb. 2001.

[9] M. C. Davey and D. J. C MacKay, "Low Density Parity Check Codes over GF(q)," *IEEE Communications Letters*, vol. 2, pp. 165–167, June 1998.

[10] L. Hanzo, T. H. Liew, B. L. Yeap, S. X. Ng, *Turbo Coding, Turbo Equalisation and Space-Time Coding for transmission over fading channels*, ch. 9, pp. 317–390. Wiley & IEEE, 2002.

[11] J. Pearl, "Probabilistic Reasoning in Intelligent Systems," *San Mateo, CA:Morgan Kaufmann*, 1988.

[12] P. Meshkat and H. Jafarkhani, "Space-Time Low-Density Parity-Check Codes," vol. 2, (Pacific Grove, Monterey, CA, USA), pp. 1117 –1121, 3-6, Nov 2002.

[13] M. R. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Transactions on Information Theory*, vol. 27, September 1981.

[14] M.C. Davey, "Error-Correction using Low Density Parity Check Codes," *Ph.D thesis, University of Cambridge,UK*, 1999.

[15] D. J. C. MacKay and R. M. Neal, "Good Error-Correction Codes Based on Very Sparse Matrices," *IEEE Transactions on Information Theory*, vol. 45, pp. 399–431, March 1999.