# Are Existing Iteratively Decoded QR Codes Suitable for Energy Efficient M2M Communication?

Stojan Z. Denic
Telecommunications Research Laboratory
Toshiba Research Europe Limited
Bristol, BS1 4ND UK
Email: stojan.denic@toshiba-trel.com

Mohamed R. Ismail
Telecommunications Research Laboratory
Toshiba Research Europe Limited
Bristol, BS1 4ND UK
Email: mohamed@toshiba-trel.com

***Abstract*—The application potential of iteratively decoded extended quadratic residue (EQR) codes for machine-to-machine (M2M) communication is explored. The M2M communication typically requires energy-efficient transmission of short packets making EQR codes good candidates for this emerging technology. Novel performance and decoding complexity results are presented for four EQR codes having codeword lengths previously uninvestigated under iterative decoding. The encoders and decoders are constructed for three different iterative decoding algorithms. Our study suggests limitations of current theoretical tools in short error-correction code design and possible directions for complexity reduction implying better energy efficiency.**

## I. Introduction

Automation, information and communication technologies have rapidly become part of many aspects of human activities ranging from industrial control and monitoring, traffic control and home management to energy management, environmental monitoring and security [1]. With the ever increasing interconnectedness of devices within such systems – often called machine-to-machine (M2M), a critical part is the communication subsystem, which needs to be: 1) energy efficient, 2) highly reliable and 3) delay intolerant. From the point of view of error-correction techniques, these criteria can be related to codeword length. In general, the shorter codeword length means a shorter delay in processing of a conveyed message. At the same time, corresponding decoders should be designed in such a manner so as to be energy efficient.

M2M communication can be categorised by at least two types of messages, control messages and state messages describing all network variables; both types are characterised by short length packets. Error correction coding systems designed for wireless LAN and cellular systems perform far from optimally at short packet lengths. Thus, there is a requirement for good coding schemes at short packet lengths and associated decoding schemes. From information theory, the channel capacity is approached when the code length is very large, as practically realised by low-density parity-check (LDPC) codes. In the past two decades or so, however, attention has turned to the study of shorter length codes, their efficient decoding and their achievable performance [2], [3], [4], [5], [6].

In [5], the imperfectness of a code, for a given code rate, information word size and target codeword error rate, was defined as the signal-to-noise ratio (SNR) penalty of the code with respect to the sphere-packing bound. It was found that according to the imperfectness criterion, there did not exist "nearly perfect" codes for information block sizes between 24 and 200 bits. For less than 24 bits, the authors marked extended quadratic residue (EQR) codes as nearly perfect and for greater than 200 bits turbo codes were shown as nearly perfect.

The focus of this paper is on the iterative belief propagation (BP) based decoding of EQR codes that belong to the previously mentioned gap - 24 to 200 information bit block sizes - when three different decoding algorithms are employed. These algorithms offer reduced complexity as compared to the maximum likelihood (ML) decoders and very good performance for information block sizes below and equal to 24. The goal of this paper is to: 1) design encoders and decoders for the aforementioned codeword lengths which is not a trivial task, and 2) present results on the code performance and complexity. The observed complexity is going to suggest the decoders' energy efficiency and if there is any space for the design improvement.

Previous work on the BP based decoding of algebraic codes (see [7], [8], [9] and [10]) considers a limited number of codes of various classes including Bose – Chaudhuri – Hocquenghem and quadratic residue (QR) codes, but does not carry out a systematic study of QR and EQR codes of various codeword lengths. Other related work can be found in [4], [11], [12] and [13].

### A. Paper Contribution and Organisation

The contributions of this paper are as follows:

- For all considered codes, new and optimized parity-check matrices with reduced number of short cycles are constructed, and a comparison of error correcting performance before and after cycle reduction is provided.
- It reports on the performance and decoding complexity of three iterative soft-decision decoding algorithms (RRD, modified RRD and MBBP) for EQR codes of codeword length 72, 74, 80 and 90 bits.
- For each of the studied codes, a count of the number of cycles of length four and six for the corresponding parity-check matrices is provided.

- The limitations of theoretical tools for the design of short error-correction codes are highlighted.

Section II describes the construction of QR and EQR codes. Section II-B explains automorphism groups of cyclic codes and provides brief descriptions of the three decoding algorithms studied. Section III provides: 1) definitions for measuring the *goodness* of a code, 2) results for cycle reduction of the parity check matrices and 3) packet error rate (PER) performance of the three decoding algorithms. Finally, section IV gives some recommendations for using iterative soft decoding algorithms with EQR codes.

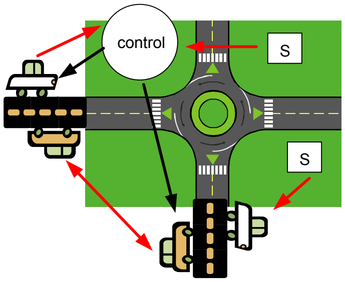

Fig. 1. M2M communication illustration.

## II. System Model and Short EQR Codes

An illustration of one possible scenario of the M2M communication is shown in Fig 1, for traffic improvement and control. There are three types of participants: controllers, sensors in the vehicles and sensors on the road. The participants can exchange two kinds of messages, control signals and state information messages depicted by black and red arrows, respectively. Of crucial importance is that the state information and control messages be delivered in a timely manner, so that vehicles execute appropriate commands at the appropriate time. Therefore, the delay in such networks has to be limited, and from the point of view of error-correction code design this means that the decoding process has to be fast, hence of a low complexity. Another feature of such control-communication networks is that control signals are described by a relatively low number of bits. In addition, an optimal solution has to be energy efficient since the majority of devices will not be connected to the mains. This can be provided by low decoding complexity.

The aforementioned requirements point to application of very short error-correction codes. The previous research [5] suggests that QR and EQR codes could be the best candidates for control-communication networks.

### *A. QR and EQR Codes*

QR codes are cyclic codes of length prime $p$ defined over a field $GF(l)$, where $l$ is another prime which is a quadratic residue mod $p$ [14]. For an odd prime $p$, the non-zero squares modulo $p$ are called the quadratic residues mod $p$. There are $(p-1)/2$ quadratic residues and $(p-1)/2$ non-residues.

Our analysis is focused on several binary quadratic residue codes where $l = 2$. For $l = 2$, it is proven that the prime $p$ has to be of the form $8m \pm 1$. Some examples of binary QR codes are the $[7, 4, 3]$ Hamming code and the $[23, 12, 7]$ Golay code.

There are several ways how the QR encoder can be realized. One way is to use a special type of polynomials, called an idempotent $E(x)$, belonging to the ring $R_n = GF(l)[x]/(x^n - 1)$. The ring $R_n$ consists of residue classes of $GF(l)[x]$ modulo $(x^n - 1)$, where $GF(l)[x]$ is the set of all polynomials in $x$ with coefficients from $GF(l)$. The *idempotent* $E(x)$ is any polynomial which satisfies $E(x) = E(x)^2 = E(x^2)$. For example, for $l = 2$ and $p = 4k-1$, the idempotents can take the following forms $E_q(x) = \sum_{r\in Q} x^r$, $F_q(x) = 1 + \sum_{n\in N} x^n$, $E_n(x) = \sum_{n\in N} x^n$ and $F_n(x) = 1 + \sum_{r\in Q} x^r$ which are used to generate the QR codes denoted by $\mathcal{L}$, $\bar{\mathcal{L}}$, $\mathcal{N}$ and $\bar{\mathcal{N}}$, respectively. Further, a generator matrix $G$ for one type of QR codes $\bar{\mathcal{L}}$ can be formed by taking the coefficients of $F_q(x)$ to represent one row of $G$, while the rest of the rows are obtained by circularly shifting the first row. The second type of QR code $\mathcal{L}$ is obtained by appending the all-one row to $G$ corresponding to $\bar{\mathcal{L}}$. Another important code class is that of extended QR codes which is obtained by appending a parity bit at the end of each codeword of a QR code. An interesting characteristic is that for $p = 4k - 1$, the extended QR code $\hat{\mathcal{L}}$ of $\mathcal{L}$ is self-dual.

The main characteristics of QR codes are that they have large minimum distance and large automorphism groups [14] which will be described in the following section.

### *B. Iterative Decoding Of EQR codes*

In this paper, three different iterative algorithms will be tested in an additive white Gaussian noise (AWGN) channel. All the three algorithms are based on the belief propagation algorithm. As the parity check matrices for EQR codes are dense, the probability of short cycles being present is higher than for LDPC codes. Hence, decoding of EQR codes using BP-based algorithms will be impaired. This is the reason the three considered algorithms use the existence of large automorphism groups of the EQR codes to modify a classical BP decoding approach.

Let $\mathcal{C}$ be a block code of length $n$, then the permutation group of the code $\mathcal{C}$, $Per(\mathcal{C})$, is defined as the set of permutations of coordinate places which send $\mathcal{C}$ onto itself, also referred to as the *automorphism group* in [15]. Let $p$ be a prime of the form $8m \pm 1$, then the *projective special linear group* $PSL_2(p)$ is generated by the set of permutations of $\{0, 1, 2, \ldots, p-1, \infty\}$ by $S : i \to i + 1$, $V : i \to \rho^2 i$ and $T : i \to -\frac{1}{i}$, where $\rho$ is a primitive element of $GF(p)$ [14].

Presented below is a brief description of the three iterative decoding algorithms highlighting their salient features.

*1) Random Redundant Decoding (RRD):* The inner loop of the RRD algorithm uses a BP decoding algorithm with the addition of a damping factor, $\alpha$ [8]. An outer loop iterates over different values of $\alpha$. For each iteration of the inner loop, a random group element (permutation) is chosen from the automorphism group of the code and applied to the soft output of the BP decoder. The applied permutations are accumulated over the iterations with the inverse of the accumulated permutation being applied at the end of the decoding procedure.

*2) Multiple Basis Belief Propagation (MBBP):* The MBBP algorithm operates multiple BP decoders in parallel, each one using a different parity-check matrix representation of the dual code to decode the received codeword. For a $[n, k, d]$ linear code where $k$ is the information word size, $n$ codeword length and $d$ minimum distance $d$, at least the $(n-k)$ linearly independent or more linearly dependent codewords of the dual code $\mathcal{C}^{\perp}$ can be used to form a parity-check matrix of $\mathcal{C}$. However, as noted in [7], not all matrices formed from such codewords are suitable for decoding using BP, rather only minimum-weight codewords of the dual code should, wherever possible, be used. The reason for this is that parity-check matrices for decoding should be as sparse as possible. For the decoders which have converged to a valid codeword the output is passed to a decision unit, the least metric selector (LMS). The LMS unit selects the codeword which best satisfies the decision rule $\hat{c} = \underset{s \in S}{\arg\max} \ \Pr\{y|\hat{c}_s\}$, where $y$ is the received codeword, $S$ is the set of of decoders producing a valid codeword, $\hat{c}_s$ is a decoded codeword by the decoder $s \in S$, and $\hat{c}$ is the best choice for the decoded codeword according to LMS.

*3) Modified Random Redundant Decoding (mRRD):* In [8] Halford made the important observation that decoding with an initial parity-check matrix $H$ which has been permuted by $\beta$, where $\beta \in Per(\mathcal{C})$, is equivalent to decoding with the initial parity-check matrix and a permuted codeword, $\beta^{-1}y$. The mRRD algorithm [10] combines the RRD and MBBP algorithms by permuting the received codeword in an identical manner to the RRD algorithm whilst having multiple BP decoders all decoding with the same parity-check matrix. The mRRD algorithm is, in essence, a parallelization of the RRD algorithm. One important distinguishing feature of the mRRD algorithm is the absence of the damping factor used in the RRD algorithm.

## III. Case Study of Four EQR Codes

### A. Methodology

To analyse EQR code performance under the modified BP decoding algorithms, the three criteria are employed: 1) The code *imperfectness*, 2) The number of cycles of the parity-check matrices, and 3) The stopping sets.

We first discuss the code imperfectness. It is introduced in [5] which explores and reformulates Shannon's sphere-packing bound which is a useful performance benchmark for comparing all codes $[n, k, d]$ of a given code rate $R = k/n$.

Table I shows required minimal SNRs according to Shannon's sphere-packing bound for code rate $R = 1/2$ versus $k$ to achieve a codeword error rate of $P_w = 10^{-4}$. (The SNR is defined as $E_b/N_0$, where $E_b$ is the energy per transmitted information bit, while $N_0$ is the so-called double-sided power spectral density of the noise.) It can be observed that the required SNR decreases with the information word size $k$. Thus, theoretically, the longer the codeword the smaller required minimum SNR. On the other hand, longer codes will require more complex decoders to achieve the theoretically predicted performance, as shown later.

TABLE I
Minimum SNR to achieve $P_w = 10^{-4}$ as a function of information word length, $k$, $R = 1/2$.

| $k$ | 10 | 50 | 100 | 200 | 1000 | 10000 |
|---|---|---|---|---|---|---|
| SNR (dB) | 4.8 | 3 | 2.4 | 1.8 | 0.8 | 0.2 |

*Definition 3.1:* For a rate $R$ and information word length $k$, the *imperfectness* of a code is the difference between the SNR required to achieve a given codeword error rate, $P_w$, and the minimum SNR provided by Shannon's sphere packing bound for the same $R$, $k$ and $P_w$. A code is considered *nearly perfect* if it has an imperfectness less than 1 dB.

In [5], it is shown that EQR codes $[8, 4, 4]$, $[24, 12, 8]$ and $[48, 24, 12]$ have an imperfectness of 0.5. The current work explores the imperfectness of longer EQR codes and their decoding complexity for the range of $k$ between 30 and 50 bits.

The second criteria relevant for the analysis of BP type decoders is the number of cycles of length four and six and sparseness of the Tanner graph corresponding to a decoder parity check matrix. Since EQR codes have high-density parity-check matrices, the likelihood of having short cycles is much greater compared to LDPC codes. Thus, it is necessary to design parity check matrices in a way that minimizes the number of non-zero entries. In [7], a construction method is proposed where each parity check matrix consists of rows corresponding to cyclic shifts of one minimum-weight codeword of the dual code. For all EQR codes treated further, the algorithm found in [16] is used to find the minimum-weight codewords of a dual EQR code.

To further improve error correcting performance of the codes, a cycle reduction algorithm of [9] transforms previously designed parity check matrices. The cycle reduction algorithm is a brute-force algorithm which performs exhaustive linear transformations of a parity check matrix in order to reduce the number of cycles of length four and six.

### B. Decoding Performance and Discussion

At the transmitter, the information words are randomly generated, encoded using the generator matrix for one of chosen EQR codes, modulated by a binary phase-shift keying and then passed through an AWGN channel. At the receiver, the log-likelihood ratio (LLR) of the received codeword is calculated and passed to the decoding algorithm under test. We examine EQR codes $[72, 36, 12]$, $[74, 37, 14]$, $[80, 40, 16]$ and $[90, 45, 18]$.

For the RRD and mRRD decoding algorithms, which decode with a single parity check matrix, the set of matrices generated from the minimum-weight codewords of the dual

TABLE II
THE NUMBER OF MINIMUM-WEIGHT CODEWORDS FOUND BY THE ALGORITHM IN [16]

.

| $k$ | 36 | 37 | 40 | 45 |
|---|---|---|---|---|
| No. of codewords | 35 | 90 | 634 | 2423 |

TABLE III
THE MBBP DECODER PARAMETERS.

| $n$ | No. iterations/matrices | No. decoders. | Total no. of iter. |
|---|---|---|---|
| 72,72CR | 100 | 35 | 3500 |
| 74,74CR | 100 | 90 | 9000 |
| 80,80CR | 20 | 634 | 12860 |
| 90 | 10 | 2423 | 24230 |
| 90CR | 60 | 300 | 18000 |

code [7] is searched to find the one with the lowest cycle count. This parity check matrix is then used for the decoding process. For the MBBP decoding algorithm, the whole set or a subset of the decoding matrices is employed depending on a desired complexity and/or performance.

Table II shows the number of minimum-weight codewords of the dual EQR codes found by the algorithm proposed in [16]. For each of the codewords, a corresponding parity check matrix is constructed and used by the decoding algorithms.

Fig. 2 shows the average number of cycles of length $4$ and $6$ for the parity-check matrices of the EQR codes given in Table II. It can be observed that the number of cycles grows quickly with code length and even after an application of the cycle reduction algorithm the number of short cycles is still large.

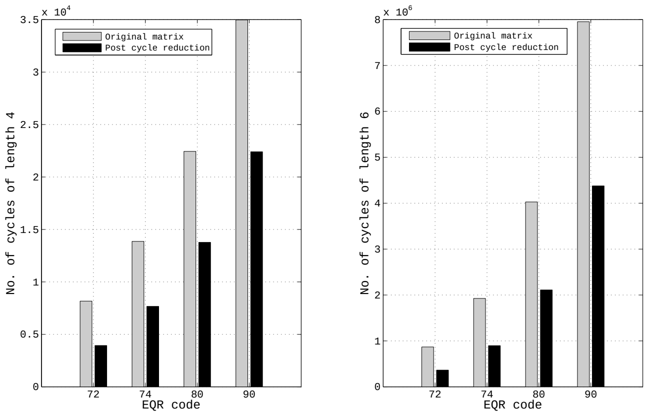

Fig. 2. Cycle count for parity-check matrices of the EQR codes pre and post cycle reduction.

For the MBBP algorithm, Table III presents the number of decoding matrices and decoding iterations used for the different codes. It should be noted that the choice of parameters is found empirically, where the goal is to achieve the PER of $10^{-4}$ with an SNR in the region between $4.5$ and $6$ dB. The notation CR refers to the case when a decoding parity-check matrix has a reduced number of cycles. E.g., for the code $[72, 36, 12]$, the number of iterations per decoder is $100$, while the number of the decoders is $35$, for both decoding algorithm versions, with and without cycle reduction. One should note that these parameters are chosen to observe theoretical limitations of the codes; it is understood that practical applications would possibly not permit such a big number of decoder iterations and number of decoders.

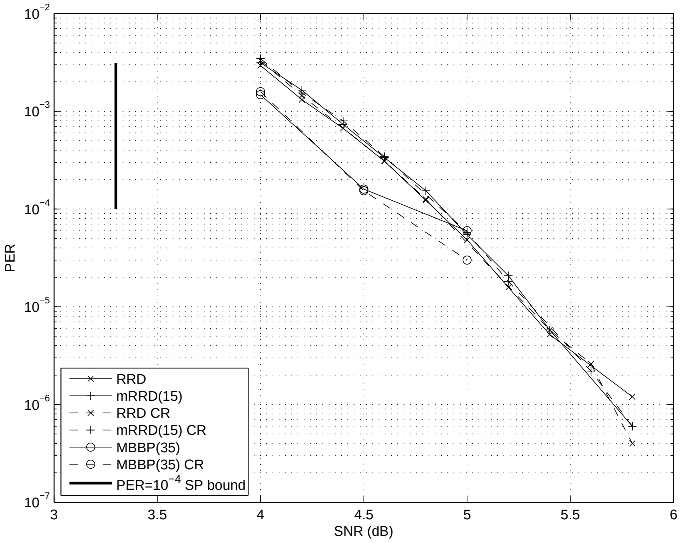

Fig. 3. [72,36,12] Extended QR code PER.

For the RRD algorithm, all codes are run with a maximum of $30$ iterations of the BP algorithm, $30$ iterations of the outer loop corresponding to modification of the damping factor and $30$ iterations of the loop corresponding to selection of a random permutation. For the mRRD algorithm a maximum of $30$ iterations of the BP algorithm and $30$ iterations of the outer loop corresponding to selection of a random permutation were used for each of the $15$ decoders used in decoding with the same parity check matrix.

Fig. 3-6 present the PER results for the MBBP, RRD and mRRD algorithms. Two sets of plots are shown for each code and decoding algorithm, with and without application of the cycle reduction algorithm to the parity check matrix for the respective code. Those curves marked CR refer to the cycle reduced results. Also shown on the graphs is the SNR required to achieve a PER of $10^{-4}$ as given by the Shannon's sphere packing bound.

Although Fig. 2 indicates a significant reduction in the number of cycles of length four and six of the decoding parity-check matrices, this however has a little impact on the PER for the four EQR codes. We suspect this happens because the number of short cycles remains large even after application of the cycle reduction algorithm.

Analysis of results from decoding with the MBBP algorithm

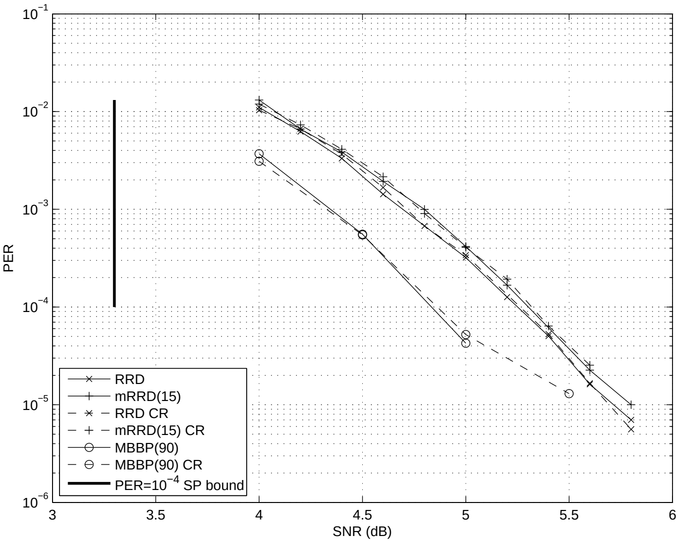

Fig. 4. [74,37,14] Extended QR code PER.

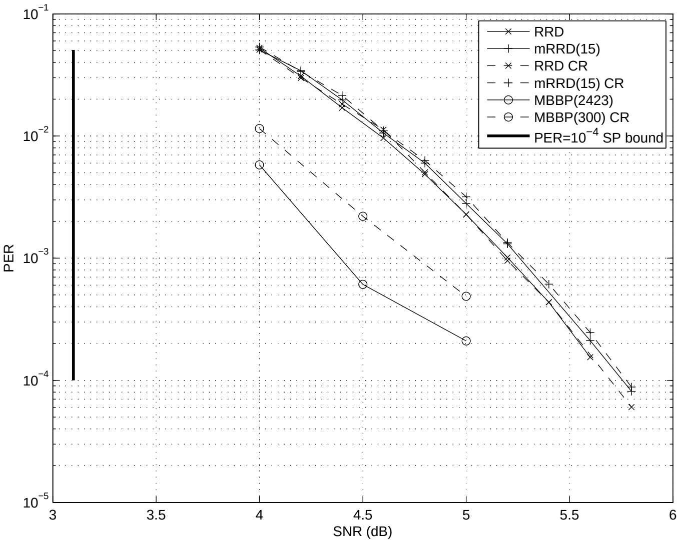

Fig. 6. [90,45,18] Extended QR code PER.

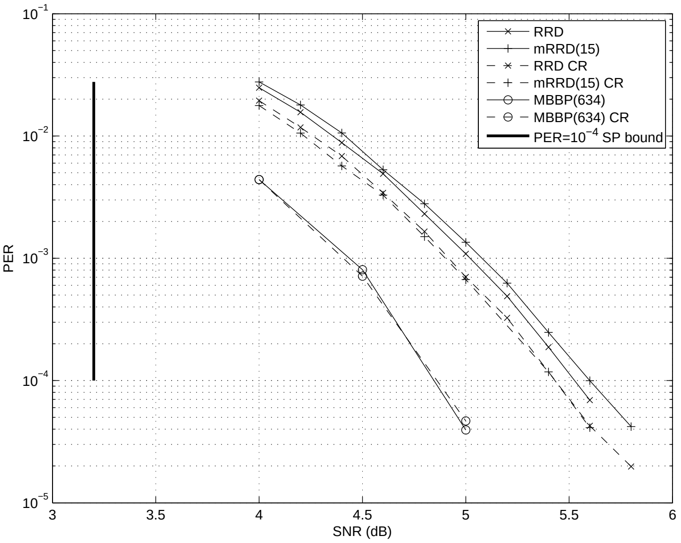

Fig. 5. [80,40,16] Extended QR code PER.

shows that the PER curves for the first three codes intersect the $10^{-4}$ line between $4.5$ and $5$ dB. The fourth code $[90, 45, 18]$ requires a larger SNR to reach a PER of $10^{-4}$. For the latter code, since the number of potentially good parity-check matrices found is large (see Table II), and the cycle reduction algorithm is time consuming, a subset of 300 out of 2423 parity-check matrices has been chosen. To keep the overall number of decoding iterations comparable with the case when there is no cycle reduction of the parity-check matrix, the number of decoding iterations per matrix is increased to 60. However, the simulation results for this code with cycle reduction, show that the present choice of parameters for the decoder – 60 decoder iterations and 300 decoders – is not enough to match the PER performance of the same code with a larger number of the decoder matrices and without cycle reduction. From Definition 3.1 and for the targeted PER of $10^{-4}$, the resulting imperfectness with the MBBP decoder for the codes $[72, 36, 12]$, $[74, 37, 14]$, $[80, 40, 16]$ and $[90, 45, 18]$ are 1.3, 1.5, 1.6 and $\geq$ 1.9 dB, respectively. Thus, it can be said that only the first code with a chosen set of the MBBP decoder parameters is close to being nearly perfect. If possible, improving the performance of these codes would require a larger number of BP iterations and/or decoding matrices which raises the question of practicality of such coding schemes. Comparing the imperfectness to other codes found in [5], at least the first three EQR codes outperform terminated convolutional codes of corresponding codeword lengths and constraint-length-7 whose imperfectness is 1.7, 1.72 and 1.8 dB, respectively. However, this is achieved at the expense of larger complexity.

For the RRD and mRRD algorithms, the SNR required to achieve a PER of $10^{-4}$ ranges from $4.75$ to $5.75$ dB. With the exception of the $[80, 40, 16]$ code, application of the cycle reduction algorithm makes little difference to PER performance. For the $[80, 40, 16]$ code, the cycle reduction gives a small improvement for both the RRD and mRRD algorithms between $0.1$ and $0.2$ dB.

Overall, the MBBP decoder has better performance than the other decoders for all four codes. We assume that this is due to the greater diversity afforded by multiple parity check matrices. Our simulations show similar variation in behaviour for the mRRD algorithm when decoding with more or less parity check matrices. Table IV shows the SNR required at the PER of $10^{-3}$ for the MBBP and RRD algorithms, the $\Delta$SNR column shows the difference between the SNR values. The final column gives the number of decoding matrices used by the MBBP algorithm. As the number of decoding matrices used by the MBBP decoder increases, the difference in the required SNR to achieve the PER of $10^{-3}$ increases, highlighting the benefit of having more decoding matrices.

TABLE IV
COMPARISON BETWEEN MBBP AND RRD DECODERS AT PER=$10^{-3}$.

| $n$ | MBBP SNR | RRD SNR | ΔSNR | MBBP mat. |
|---|---|---|---|---|
| 72 | 4.09 | 4.33 | -0.24 | 35 |
| 74 | 4.34 | 4.76 | -0.42 | 90 |
| 80 | 4.44 | 5.04 | -0.60 | 634 |
| 90 | 4.39 | 5.20 | -0.81 | 2423 |

From Table III the maximum number of decoding iterations used by the MBBP algorithm for the four codes is seen in the fourth column. The RRD algorithm allows for a maximum of $30 \times 30 \times 30 = 27000$ decoding iterations. Whereas, the mRRD algorithm allows for a maximum of $30 \times 30 \times 15 = 13500$ decoding iterations. For implementation purposes, the RRD and mRRD algorithms have the advantage of using only one parity check matrix, thereby greatly reducing the amount of storage required compared to the MBBP algorithm. The permutation step required by the RRD and mRRD algorithms is easily implemented in hardware by appropriate addressing of the parity check matrix memory.

Another factor affecting the error correcting performance of iterative decoders is that of stopping sets [17]. Counting the stopping sets is time consuming; we ran a stopping set counting algorithm for the $[72, 36, 12]$ and $[74, 37, 14]$ codes and found that all 35 and 90 parity-check matrices, respectively, did not contain stopping sets of length smaller than 8. So, we were not able to apply parity check matrix optimization based on a criterion suggested in [7]. From this point of view all the parity check matrices were equivalent. It seems that at least in this case, the stopping set criterion is of little help. Other criteria based on the trapping sets and pseudo-codewords [18] could be more helpful for AWGN channels.

### C. Energy Efficiency and Decoding Complexity

The previous analysis demonstrates that design of error-correction schemes for M2M communications is a challenging problem. It affects all three important features of the M2M network, reliability, energy efficiency and delay. Improved reliability and performance require larger decoding complexity in terms of number of iterations and used decoding matrices (in the case of MBBP decoders), while this implies increased energy consumption. Moreover, as the codeword length becomes larger, the decoders are more complex and energy hungry. Therefore, future research should address this tradeoff among reliability, performance and energy efficiency for moderate codeword lengths.

## IV. CONCLUSION

Since M2M communication requires a short packet transmission, this paper analyses a number of issues which are related to the design of short EQR error correction codes subject to iterative BP type decoding. Three different decoding algorithms were examined for four EQR codes having codeword lengths from 72 to 90. We demonstrated practical difficulties encountered in the decoder design for mentioned codeword lengths such as minimum-weight codeword exploration, cycle and stopping set counting and decoder complexity. We found that some of the classical theoretical tools for the decoder design – e.g. cycle counting and stopping sets – could be of little help when moderate to larger codeword lengths are considered. It was observed that the decoder complexity increases rapidly with codeword length, and a true cost of complexity can only be understood when analysing a hardware implementation.

## REFERENCES

[1] S.-Y. Lien, K.-C. Chen, and Y. Lin, “Toward ubiquitous massive accesses in 3gpp machine-to-machine communications,” *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 66–74, 2011.

[2] I. S. Reed, X. Yin, and T.-K. Truong, “Algebraic decoding of the (32, 16, 8) quadratic residue code,” *Information Theory, IEEE Transactions on*, vol. 36, no. 4, pp. 876–880, 1990.

[3] Y.-H. Chen, T.-K. Truong, Y. Chang, C.-D. Lee, and S.-H. Chen, “Algebraic decoding of quadratic residue codes using berlekamp-massey algorithm,” *Journal of information science and engineering*, vol. 23, no. 1, pp. 127–145, 2007.

[4] R. Lucas, M. Bossert, and M. Breitbach, “On iterative soft-decision decoding of linear binary block codes and product codes,” *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 2, pp. 276–296, 1998.

[5] S. Dolinar, D. Divsalar, and F. Pollara, “Code performance as a function of block size,” *Telecommunications and Mission Operations Progress Report*, vol. 133, pp. 1–23, Jan. 1998.

[6] Y. Polyanskiy, H. V. Poor, and S. Verdú, “Channel coding rate in the finite blocklength regime,” *Information Theory, IEEE Transactions on*, vol. 56, no. 5, pp. 2307–2359, 2010.

[7] T. Hehn, J. Huber, O. Milenkovic, and S. Laendner, “Multiple-bases belief-propagation decoding of high-density cyclic codes,” *Communications, IEEE Transactions on*, vol. 58, no. 1, pp. 1–8, 2010.

[8] T. Halford and K. Chugg, “Random redundant soft-in soft-out decoding of linear block codes,” in *Information Theory, 2006 IEEE International Symposium on*, 2006, pp. 2230–2234.

[9] ——, “Transactions letters - random redundant iterative soft-in soft-out decoding,” *Communications, IEEE Transactions on*, vol. 56, no. 4, pp. 513–517, 2008.

[10] I. Dimnik and Y. Be’ery, “Improved random redundant iterative HDPC decoding,” *Communications, IEEE Transactions on*, vol. 57, no. 7, pp. 1982–1985, 2009.

[11] J. Jiang and K. Narayanan, “Iterative soft decoding of Reed-Solomon codes,” *Communications Letters, IEEE*, vol. 8, no. 4, pp. 244–246, 2004.

[12] ——, “Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix,” *Information Theory, IEEE Transactions on*, vol. 52, no. 8, pp. 3746–3756, 2006.

[13] A. Kothiyal and O. Takeshita, “A comparison of adaptive belief propagation and the best graph algorithm for the decoding of linear block codes,” in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, 2005, pp. 724–728.

[14] F. MacWilliams and N. Sloane, *The Theory of Error Correcting Codes*, ser. North-Holland Mathematical Library. North-Holland Publishing Company, 1977, no. v. 1-2. [Online]. Available: http://books.google.co.uk/books?id=aB8IAQAAIAAJ

[15] J. MacWilliams, *Permutation Decoding of Systematic Codes*, ser. Bell telephone system technical publications. Bell Telephone Laboratories, 1964. [Online]. Available: http://books.google.co.uk/books?id=Ms_1mgEACAAJ

[16] J. S. Leon, “A probabilistic algorithm for computing minimum weights of large error-correcting codes,” *Information Theory, IEEE Transactions on*, vol. 34, no. 5, pp. 1354–1359, 1988.

[17] T. Richardson and R. L. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

[18] C. A. Kelley and D. Sridhara, “Pseudocodewords of tanner graphs,” *Information Theory, IEEE Transactions on*, vol. 53, no. 11, pp. 4013–4038, 2007.