Chapter 15
The Modified Dorsch Decoder

15.1 Introduction

In a relatively unknown paper published in 1974, Dorsch [4] described a decoder for linear binary block (n, k) codes using soft decisions quantised to J levels. The decoder is applicable to any linear block code and does not rely upon any particular features of the code, such as being a concatenated code or having a sparse parity-check matrix. In the Dorsch decoder, hard decisions are derived from the soft decisions using standard bit by bit detection, choosing the binary state closest to the received coordinate. The hard decisions are then ranked in terms of their likelihoods and candidate codewords are derived from a set of k, independent, most likely bits. This is done by producing a new parity-check matrix $\mathbf{H_I}$ obtained by reordering the columns of the original $\mathbf{H}$ matrix according to the likelihood of each coordinate, and reducing the resulting matrix to echelon canonical form by elementary row operations. After evaluation of several candidate codewords, the codeword with the minimum soft decision metric is output from the decoder. A decoder using a similar principle, but without soft decision quantisation, has been described by Fossorier [5, 6]. Other approaches, after ranking the reliability of the received bits, adopt various search strategies for finding likely codewords [11] or utilise a hard decision decoder in conjunction with a search for errors in the least likely bit positions [2, 15].

The power of the Dorsch decoder arises from the relatively unknown property that most codes, *on average*, can correct almost $n - k$ erasures [17], which is considerably more than the guaranteed number of correctable erasures of $d_{min} - 1$, or the guaranteed number of correctable hard decision errors of $\frac{d_{min}-1}{2}$, where d_{min} is the minimum Hamming distance of the code. In its operation, the Dorsch decoder needs to correct any combination of $n - k$ erasures which is impossible unless the code is an MDS code [12]. Dorsch did not discuss this problem, or potential solutions, in his original paper [4], although at least one solution is implied by the results he presented.

M. Tomlinson et al., *Error-Correction Coding and Decoding*,
Signals and Communication Technology,
DOI 10.1007/978-3-319-51103-0_15

In this chapter, a solution to the erasure correcting problem of being able to solve $n-k$ erasures for a non-MDS code is described. It is based on using alternative columns of the parity-check matrix without the need for column permutations. It is also shown that it is not necessary to keep recalculating each candidate codeword and its associated soft decision metric in order to find the most likely codeword. Instead, an incremental correlation approach is adopted which features low information weight codewords and a correlation function involving only a small number of coordinates of the received vector [17]. It is proven that maximum likelihood decoding is realised provided all codewords are evaluated up to a bounded information weight. This means that maximum likelihood decoding may be achieved for a high percentage of received vectors. The decoder lends itself to a low complexity, parallel implementation involving a concatenation of hard and soft decision decoding. It produces near maximum likelihood decoding for codes that can be as long as 1000 bits, provided the code rate is high enough. When implementing the decoder, it is shown that complexity may be traded-off against performance in a flexible manner. Decoding results, achieved by the decoder, are presented for some of the most powerful binary codes known and compared to Shannon's sphere packing bound [14].

The extension to non-binary codes is straightforward and this is described in Sect. 15.5.

15.2 The Incremental Correlation Dorsch Decoder

Codewords with binary coordinates having state 0 or 1, are denoted as:

$$\mathbf{x} = (x_0, x_1, x_2, \ldots, x_{n-1})$$

For transmission, bipolar transmission is used with coordinates having binary state 0 mapped to $+1$ and having state 1 mapped to -1. Transmitted codewords are denoted as

$$\mathbf{c} = (c_0, c_1, c_2, \ldots, c_{n-1})$$

The received vector $\mathbf{r}$ consists of n coordinates $(r_0, r_1, r_2, \ldots, r_{n-1})$ equal to the transmitted codeword plus Additive White Gaussian Noise with variance σ^2. The received vector processed by the decoder is assumed to have been matched filtered and free from distortion so that $\frac{1}{\sigma^2} = \frac{2E_b}{N_o}$, where E_b is the energy per information bit and N_o is the single sided noise power spectral density. Accordingly,

$$\sigma^2 = \frac{N_o}{2E_b}$$

The basic principle that is used is that the k most reliable bits of the received vector are initially taken as correct and the $n-k$ least reliable bits are treated as erasures. The parity-check equations of the code, as represented by $\mathbf{H}$, are used to solve for

these erased bits and a codeword $\hat{\mathbf{x}}$ is obtained. This codeword is either equal to the transmitted codeword or needs only small changes to produce a codeword equal to the transmitted codeword. One difficulty is that, depending on the code, the $n-k$ least reliable bits usually cannot all be solved as erasures. This depends on the positions of the erased coordinates and the power of the code. Only Maximum Distance Separable (MDS) codes [12] are capable of solving $n-k$ erasures regardless of the positions of the erasures in the received codeword. Unfortunately, there are no binary MDS codes apart from trivial examples. However, a set of $n-k$ erasures can always be solved from $n-k+s$ least reliable bit positions, and, depending on the code, s is usually a small integer. In order to obtain best performance it is important that the very least reliable bit positions are solved first, since the corollary that the $n-k$ least reliable bits usually cannot all be solved as erasures is that the k most reliable bits, used to derive codeword $\hat{\mathbf{x}}$, must include a small number of least reliable bits. However, for most received vectors, the difference in reliability between ranked bit k and ranked bit $k+s$ is usually small. For any received coordinate, the a priori log likelihood ratio of the bit being correct is proportional to $|r_i|$. The received vector $\mathbf{r}$ with coordinates ranked in order of most likely to be correct is defined as $(r_{\mu_0}, r_{\mu_1}, r_{\mu_2}, \ldots, r_{\mu_{n-1}})$, where $|r_{\mu_0}| > |r_{\mu_1}| > |r_{\mu_2}| > \cdots > |r_{\mu_{n-1}}|$.

The decoder is most straightforward for a binary MDS code. The codeword coordinates $(x_{\mu_0}, x_{\mu_1}, x_{\mu_2}, \ldots, x_{\mu_{k-1}})$ are formed directly from the received vector $\mathbf{r}$ using the bitwise decision rule $x_{\mu_i} = 1$ if $r_{\mu_i} < 0$ else $x_{\mu_i} = 0$. The $n-k$ coordinates $(x_{\mu_k}, x_{\mu_{k+1}}, x_{\mu_{k+2}}, \ldots, x_{\mu_{n-1}})$ are considered to be erased and derived from the k most reliable codeword coordinates $(x_{\mu_0}, x_{\mu_1}, x_{\mu_2}, \ldots, x_{\mu_{k-1}})$ using the parity-check equations.

For a non-MDS code, the $n-k$ coordinates cannot always be solved from the parity-check equations because the parity-check matrix is not a Cauchy or Vandermonde matrix [12]. To get around this problem a slightly different order is defined $(x_{\eta_0}, x_{\eta_1}, x_{\eta_2}, \ldots, x_{\eta_{n-1}})$.

The label of the last coordinate η_{n-1} is set equal to μ_{n-1} and $x_{\eta_{n-1}}$ solved first by flagging the first parity-check equation that contains $x_{\eta_{n-1}}$, and then subtracting this equation from all other parity-check equations containing $x_{\eta_{n-1}}$. Consequently, $x_{\eta_{n-1}}$ is now only contained in one equation, the first flagged equation.

The label of the next coordinate η_{n-2} is set equal to μ_{n-2} and an attempt is made to solve $x_{\eta_{n-2}}$ by finding an unflagged parity-check equation containing $x_{\eta_{n-2}}$. In the event that there is not an unflagged equation containing $x_{\eta_{n-2}}$, η_{n-2} is set equal to μ_{n-3} the label of the next most reliable bit, $x_{\mu_{n-3}}$ and the procedure repeated until an unflagged equation contains $x_{\eta_{n-2}}$. As before, this equation is flagged that it will be used to solve for $x_{\eta_{n-2}}$ and is subtracted from all other unflagged equations containing $x_{\eta_{n-2}}$. The procedure continues until all of the $n-k$ codeword coordinates $x_{\eta_{n-1}}, x_{\eta_{n-2}}, x_{\eta_{n-3}}, \ldots, x_{\eta_k}$ have been solved and all $n-k$ equations have been flagged. In effect, the least reliable coordinates are skipped if they cannot be solved. The remaining k ranked received coordinates are set equal to $(r_{\eta_0}, r_{\eta_1}, r_{\eta_2}, \ldots, r_{\eta_{k-1}})$ in most reliable order, where $|r_{\eta_0}| > |r_{\eta_1}| > |r_{\eta_2}| > \cdots > |r_{\eta_{n-1}}|$ and $(x_{\eta_0}, x_{\eta_1}, x_{\eta_2}, \ldots, x_{\eta_{k-1}})$ determined using the bit decision rule $x_{\eta_i} = 1$ if $r_{\eta_i} < 0$ else $x_{\eta_i} = 0$. The flagged parity-check equations are in upper triangular form and have to be solved in reverse order starting with the

last flagged equation. This equation gives the solution to x_{η_k} which is back substituted into the other equations and $x_{\eta_{k+1}}$ is solved next, back substituted, and so on, with coordinate $x_{\eta_{n-1}}$ solved last.

This codeword is denoted as $\hat{\mathbf{x}}$ and the mapped version of the codeword is denoted as $\hat{\mathbf{c}}$.

As is well-known [13], the codeword most likely to be transmitted is the codeword, denoted as $\check{\mathbf{x}}$, which has the smallest squared Euclidean distance, $D(\check{\mathbf{x}})$, between the mapped codeword, $\check{\mathbf{c}}$, and the received vector.

$$D(\check{\mathbf{x}}) = \sum_{j=0}^{n-1} (r_j - \check{c}_j)^2$$

$D(\check{\mathbf{x}}) < D(\mathbf{x})$ for all other codewords $\mathbf{x}$.

Equivalently $\check{\mathbf{x}}$ is the codeword, after mapping, which has the highest cross correlation

$$Y(\check{\mathbf{x}}) = \sum_{j=0}^{n-1} r_j \times \check{c}_j \tag{15.1}$$

$Y(\check{\mathbf{x}}) > Y(\mathbf{x})$ for all other codewords $\mathbf{x}$.

The decoder may be simplified if the cross correlation function is used to compare candidate codewords. The cross correlation is firstly determined for the codeword $\hat{\mathbf{x}}$

$$Y(\hat{\mathbf{x}}) = \sum_{j=0}^{n-1} r_j \times \hat{c}_j \tag{15.2}$$

It is interesting to make some observations about $Y(\hat{\mathbf{x}})$. Since the summation can be carried out in any order

$$Y(\hat{\mathbf{x}}) = \sum_{j=0}^{n-1} r_{\eta_j} \times \hat{c}_{\eta_j} \tag{15.3}$$

and

$$Y(\hat{\mathbf{x}}) = \sum_{j=0}^{k-1} r_{\eta_j} \times \hat{c}_{\eta_j} + \sum_{j=k}^{n-1} r_{\eta_j} \times \hat{c}_{\eta_j} \tag{15.4}$$

Considering the first term

$$\sum_{j=0}^{k-1} r_{\eta_j} \times \hat{c}_{\eta_j} = \sum_{j=0}^{k-1} |r_{\eta_j}| \tag{15.5}$$

This is because the sign of $\hat{c}_{\eta_j}$ equals the sign of $\hat{c}_{\eta_j}$ for $j < k$. Thus, this term is independent of the code and Eq. (15.4) becomes

$$Y(\hat{\mathbf{x}}) = \sum_{j=0}^{k-1} |r_{\eta_j}| + \sum_{j=k}^{n-1} r_{\eta_j} \times \hat{c}_{\eta_j} \tag{15.6}$$

Almost all of the k largest received coordinates (all of the k largest terms for an MDS code) are contained in the first term of Eq. (15.6) and this ensures that the codeword $\hat{\mathbf{x}}$, after mapping, has a high correlation with $\mathbf{r}$.

A binary, (hard decision), received vector $\mathbf{b}$ may be derived from the received vector $\mathbf{r}$ using the bitwise decision rule $b_j = 1$ if $r_j < 0$, else $b_j = 0$ for $j = 0$ to $n - 1$. It should be noted that in general the binary vector $\mathbf{b}$ is not a codeword.

It is useful to define a binary vector $\hat{\mathbf{z}}$ as

$$\hat{\mathbf{z}} = \mathbf{b} \oplus \hat{\mathbf{x}} \tag{15.7}$$

The maximum attainable correlation Y_{max} is given by

$$Y_{max} = \sum_{j=0}^{n-1} |r_{\eta_j}| \tag{15.8}$$

This correlation value occurs when there are no bit errors in transmission and provides an upper bound to the maximum achievable correlation for $\breve{\mathbf{x}}$. The correlation $Y(\hat{\mathbf{x}})$ may be expressed in terms of Y_{max} and $\hat{\mathbf{x}}$ for

$$Y(\hat{\mathbf{x}}) = Y_{max} - 2 \sum_{j=0}^{n-1} \hat{z}_{\eta_j} \times |r_{\eta_j}| \tag{15.9}$$

equivalently,

$$Y(\hat{\mathbf{x}}) = Y_{max} - Y_{\Delta}(\hat{\mathbf{x}}), \tag{15.10}$$

where $Y_{\Delta}(\hat{\mathbf{x}})$ is the shortfall from the maximum achievable correlation for the codeword $\hat{\mathbf{x}}$ and is evidently

$$Y_{\Delta}(\hat{\mathbf{x}}) = 2 \sum_{j=0}^{n-1} \hat{z}_{\eta_j} \times |r_{\eta_j}| \tag{15.11}$$

Some observations may be made about the binary vector $\hat{\mathbf{z}}$. The coordinates $\hat{z}_{\eta_j}$ for $j = 0$ to $(k - 1)$ are always equal to zero. The maximum possible weight of $\hat{\mathbf{z}}$ is thus $n - k$ and the average weight is $\frac{n-k}{2}$ at low $\frac{E_b}{N_o}$ values. At high $\frac{E_b}{N_o}$ values, the average weight of $\hat{\mathbf{z}}$ is small because there is a high chance that $\hat{\mathbf{x}}$ is equal to the transmitted

codeword. It may be seen from Eq. (15.11) that, in general, the lower the weight of $\hat{\mathbf{z}}$ the smaller will be $Y_{\Delta}(\hat{\mathbf{x}})$ and the larger will be the correlation value $Y(\hat{\mathbf{x}})$.

Since there is no guarantee that the codeword $\hat{\mathbf{x}}$ is the transmitted codeword, the decoder has to evaluate additional codewords since one or more of these may produce a correlation higher than $\hat{\mathbf{x}}$. There are $2^k - 1$ other codewords which may be derived by considering all other $2^k - 1$ sign combinations of c_{η_j} for $j = 0$ to $k - 1$. For any of these codewords denoted as $\mathbf{c_i}$ the first term of the correlation given in Eq. (15.6) is bound to be smaller since

$$\sum_{j=0}^{k-1} r_{\eta_j} \times c_{i,\eta_j} < \sum_{j=0}^{k-1} |r_{\eta_j}| \tag{15.12}$$

This is because there has to be, by definition, at least one sign change of c_{i,η_j} compared to $\hat{c}_{\eta_j}$ for $j = 0$ to $k - 1$. In order for $Y(\mathbf{x_i})$ to be larger than $Y(\hat{\mathbf{x}})$ the second term of the correlation $\sum_{j=k}^{n-1} r_{\eta_j} \times c_{i,\eta_j}$ which uses the bits from the solved parity-check equations must be larger than $\sum_{j=k}^{n-1} r_{\eta_j} \times \hat{c}_{\eta_j}$ plus the negative contribution from the first term.

However, the first term has higher received magnitudes than the second term because the received coordinates are ordered. It follows that codewords likely to have a higher correlation than $\hat{\mathbf{x}}$ will have small number of differences in the coordinates x_{η_j} for $j = 0$ to $k - 1$. As the code is linear these differences will correspond to a codeword and codewords may be generated that have low weight in coordinates x_{η_j} for $j = 0$ to $k - 1$. These codewords are represented as $\tilde{\mathbf{x}}_\mathbf{i}$ and referred to as low information weight codewords since coordinates x_{η_j} for $j = 0$ to $k - 1$ form an information set. Thus, codewords $\mathbf{x_i}$ are given by

$$\mathbf{x_i} = \hat{\mathbf{x}} \oplus \tilde{\mathbf{x}}_\mathbf{i} \tag{15.13}$$

and $\tilde{\mathbf{x}}_\mathbf{i}$ are codewords chosen to have increasing weight in coordinates x_{η_j} for $j = 0$ to $k - 1$ as i is incremented. This means that for increasing i it will become less likely that a codeword will be found that has higher correlation than the correlation of a codeword already found.

The difference in the correlation value $Y_{\Delta}(\mathbf{x_i})$ as a function of $\tilde{\mathbf{x}}_\mathbf{i}$ may be derived. Firstly, the binary vector $\mathbf{z_i}$ is given by

$$\mathbf{z_i} = \mathbf{b} \oplus \hat{\mathbf{x}} \oplus \tilde{\mathbf{x}}_\mathbf{i} \tag{15.14}$$

which may be simplified to

$$\mathbf{z_i} = \hat{\mathbf{z}} \oplus \tilde{\mathbf{x}}_\mathbf{i} \tag{15.15}$$

The cross correlation $Y(\mathbf{x_i})$ is given by

$$Y(\mathbf{x_i}) = Y_{max} - 2\sum_{j=0}^{n-1} z_{i,\,\eta_j} \times |r_{\eta_j}| \tag{15.16}$$

equivalently

$$Y(\mathbf{x_i}) = Y_{max} - Y_{\Delta}(\mathbf{x_i}) \tag{15.17}$$

The shortfall from maximum correlation, $Y_{\Delta}(\mathbf{x_i})$, is evidently

$$Y_{\Delta}(\mathbf{x_i}) = 2\sum_{j=0}^{n-1} z_{i,\,\eta_j} \times |r_{\eta_j}| \tag{15.18}$$

Substituting for $\mathbf{z_i}$ gives $Y_{\Delta}(\mathbf{x_i})$ as a function of $\tilde{\mathbf{x}}_\mathbf{i}$.

$$Y_{\Delta}(\mathbf{x_i}) = 2\sum_{j=0}^{n-1} (\hat{z}_j \oplus \tilde{x}_{i\eta_j}) \times |r_{\eta_j}| \tag{15.19}$$

It is apparent that instead of the decoder determining $Y(\mathbf{x_i})$ for each codeword, $\mathbf{x_i}$, it is sufficient for the decoder to determine $Y_{\Delta}(\mathbf{x_i})$ for each codeword $\tilde{\mathbf{x}}_\mathbf{i}$ and compare the value with the smallest value obtained so far, denoted as $Y_{\Delta}(\mathbf{x_{min}})$, starting with $Y_{\Delta}(\hat{\mathbf{x}})$:

$$Y_{\Delta}(\mathbf{x_{min}}) = min\big(Y_{\Delta}(\mathbf{x})\big) \tag{15.20}$$

Thus it is more efficient for the decoder to compute the correlation (partial sum) of the $\tilde{\mathbf{x}}_\mathbf{i}$ instead of deriving $(\hat{\mathbf{x}} \oplus \tilde{\mathbf{x}}_\mathbf{i})$ by solving $\mathbf{H_I}$ and computing the squared Euclidean distance. Since codewords $\tilde{\mathbf{x}}_\mathbf{i}$ produce low weight in $\mathbf{z_i}$, the number of non-zero terms that need to be evaluated in Eq. (15.18) is typically $\frac{n-k}{2}$ rather than the $\frac{n}{2}$ terms of Eq. (15.1) which makes for an efficient, fast decoder. Before Eq. (15.19) is evaluated, the Hamming weight of $\mathbf{z_i}$ may be compared to a threshold and the correlation stage bypassed if the Hamming weight of $\mathbf{z_i}$ is high. There is an associated performance loss and results are presented in Sect. 15.4.

The maximum information weight $w_{inf\,max}$ necessary to achieve maximum likelihood decoding may be upper bounded from $Y_{\Delta}(\hat{\mathbf{x}})$ and $|r_{\eta_j}|$ initially, updated by $Y_{\Delta}(\mathbf{x_{min}})$ as decoding progresses, since

$$Y_{\Delta}(\mathbf{x_i}) \geq \sum_{m=0}^{w_{inf}} |r_{\eta_{k-m-1}}| \tag{15.21}$$

This is reasonably tight since there is a possibility of at least one codeword with information weight $w_{inf\,max}$, for which all of the coordinates of the binary vector $\mathbf{z_i}$ corresponding to the parity bits of $\tilde{\mathbf{x}}_\mathbf{i}$ are zero. Correspondingly, $w_{inf\,max}$ is the smallest integer such that

$$\sum_{m=0}^{w_{inf\,max}} |r_{\eta_{k-m-1}}| \geq Y_\Delta(\hat{\mathbf{x}}) \tag{15.22}$$

The codewords $\tilde{\mathbf{x}}_\mathbf{i}$ may be most efficiently derived from the $\mathbf{G}$ matrix corresponding to the solved $\mathbf{H}$ matrix because the maximum information weight given by Eq. (15.22) turns out to be small. Each row, i, of the solved $\mathbf{G}$ matrix is derived by setting $x_{\eta_j} = 0$ for $j = 0$ to $k-1$, $j \neq i$, and using the solved parity-check equations to determine x_{η_j} for $j = k$ to $n-1$. The maximum number of rows of the $\mathbf{G}$ matrix that need to be combined to produce $\tilde{\mathbf{x}}_\mathbf{i}$ is $w_{inf\,max}$.

15.3 Number of Codewords that Need to Be Evaluated to Achieve Maximum Likelihood Decoding

For each received vector the decoder needs to evaluate the correlation shortfall for the codewords $\tilde{\mathbf{x}}_\mathbf{i}$ for information weights up to the maximum information weight of $w_{inf\,max}$ in order to achieve maximum likelihood decoding. The number of codewords that need to be evaluated is a function of the received vector. Not all of the codewords having information weight less than or equal to $w_{inf\,max}$ need be evaluated because lower bounds may be derived for $Y_\Delta(\mathbf{x_i})$ in terms of the coordinates of the information bits, their total weight and the magnitudes of selected coordinates of the received vector. For an information weight of w_{inf}, $Y_\Delta(\mathbf{x_i})$ is lower bounded by

$$Y_\Delta(\mathbf{x_i}) \geq |r_{\eta_j}| + \sum_{m=0}^{w_{inf}-1} |r_{\eta_{k-m-1}}| \quad 0 \leq j < k-m \tag{15.23}$$

and

$$|r_{\eta_{j_{min}(w_{inf})}}| \geq Y_\Delta(\mathbf{x_i}) - \sum_{m=0}^{w_{inf}-1} |r_{\eta_{k-m-1}}| \quad 0 \leq j < k-m \tag{15.24}$$

where $j_{min}(w_{inf})$ is defined as the lower limit for j to satisfy Eq. (15.24). The minimum number of codewords that need to be evaluated as a function of the received vector $N(\mathbf{r})$ is given by the total number of combinations

$$N(\mathbf{r}) = \sum_{m=0}^{w_{inf}} \binom{k - j_{min}(m) - 1}{m} \tag{15.25}$$

For many short codes the minimum number of codewords that need to be evaluated is surprisingly small in comparison to the total number of codewords.

15.4 Results for Some Powerful Binary Codes

The decoder can be used with any linear code and best results are obtained for codes which have the highest known d_{min} for a given codelength n and number of information symbols k. The best binary codes are tabulated up to length 257 in Marcus Grassl's on line data base [7]. Non-binary codes, for example, ternary codes of length up to 243 symbols and GF(4) codes of length up to 256 symbols are also tabulated.

A particularly good class of codes are the binary self-dual, double-circulant codes first highlighted in a classic paper by Karlin [8]. For example the (24, 12, 8) extended Golay code is included since it may be put in double-circulant form. There is also the (48, 24, 12) bordered double-circulant code, based on quadratic residues of the prime 47 and the (136, 68, 24) bordered double-circulant code based on quadratic residues of the prime 67. These codes are extremal [3] and are doubly even, only having codeword weights that are a multiple of 4, and in these cases it is necessary that the codelengths are a multiple of 8 [3]. For higher code rates of length greater than 256, the best codes are tabulated in [12], and some of these include cyclic codes and Goppa codes.

15.4.1 The (136, 68, 24) Double-Circulant Code

This code is a bordered double-circulant code based on the identity matrix and a matrix whose rows consist of all cyclic combinations, modulo $1 + x^{67}$, of the polynomial $b(x)$ defined by

$$\begin{aligned} b(x) = 1 &+ x + x^4 + x^6 + x^9 + x^{10} + x^{14} + x^{15} + x^{16} + x^{17} + x^{19} + x^{21} + x^{22} + x^{23} + x^{24} + x^{25} + x^{26} + x^{29} \\ &+ x^{33} + x^{35} + x^{36} + x^{37} + x^{39} + x^{40} + x^{47} + x^{49} + x^{54} + x^{55} + x^{56} + x^{59} + x^{60} + x^{62} + x^{64} + x^{65} \end{aligned} \tag{15.26}$$

The Frame Error Rate (FER) of this code using the extended Dorsch decoder with a maximum number of codewords limited to 3×10^6 is shown in Fig. 15.1. Also, shown in Fig. 15.1 is Shannon's [14] sphere packing bound offset by the loss for binary transmission [1], which is 0.19 dB for a code rate of $\frac{1}{2}$.

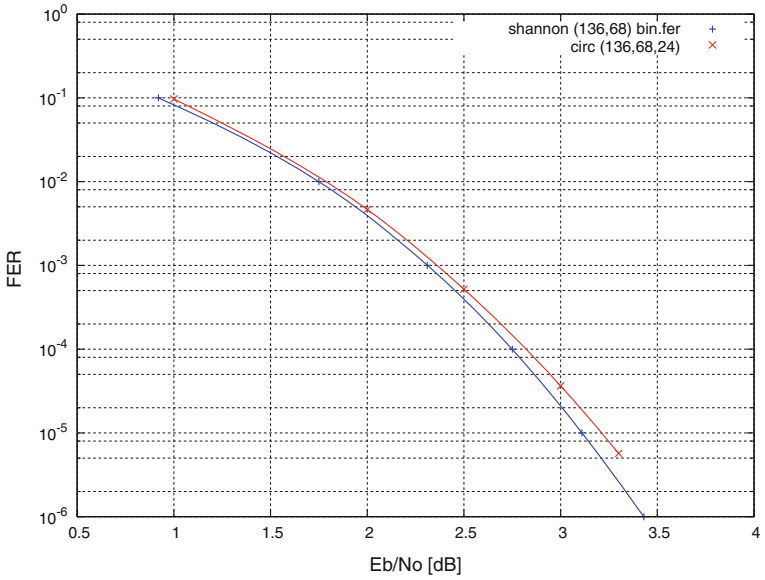

Fig. 15.1 FER as a function of $\frac{E_b}{N_o}$ for the double-circulant (136, 68, 24) code using incremental correlation decoding compared to the sphere packing bound, offset for binary transmission

It may be seen from Fig. 15.1 that the performance of the decoder in conjunction with the double-circulant code is within 0.2 dB of the best achievable performance for any (136, 68) code at 10^{-5} FER. Interestingly, there is a significant number of maximum likelihood codeword errors which have a Hamming distance of 36 or 40 from the transmitted codeword. This indicates that a bounded distance decoder would not perform very well for this code. At the typical practical operating point of $\frac{E_b}{N_o}$ equal to 3.5 dB, the probability of the decoder processing each received vector as a maximum likelihood decoder is shown plotted in Fig. 15.2 as a function of the number of codewords evaluated.

Of course to guarantee maximum likelihood decoding, all $2^{68} = 2.95 \times 10^{20}$ codewords need to be evaluated by the decoder. Equation (15.21) has been evaluated for the double-circulant (136, 68, 24) code in computer simulations, at an $\frac{E_b}{N_o}$ of 3.5 dB, for each received vector and the cumulative distribution derived. Figure 15.2 shows that by evaluating 10^7 codewords per received vector, 65% of received vectors are guaranteed to be maximum likelihood decoded. For the remaining 35% of received vectors, although maximum likelihood decoding is not guaranteed, the probability is very small that the codeword with the highest correlation is not the transmitted codeword or a codeword closer to the received vector than the transmitted codeword. This last point is illustrated by Fig. 15.3 which shows the FER performance of the decoder as a function of the maximum number of evaluated codewords.

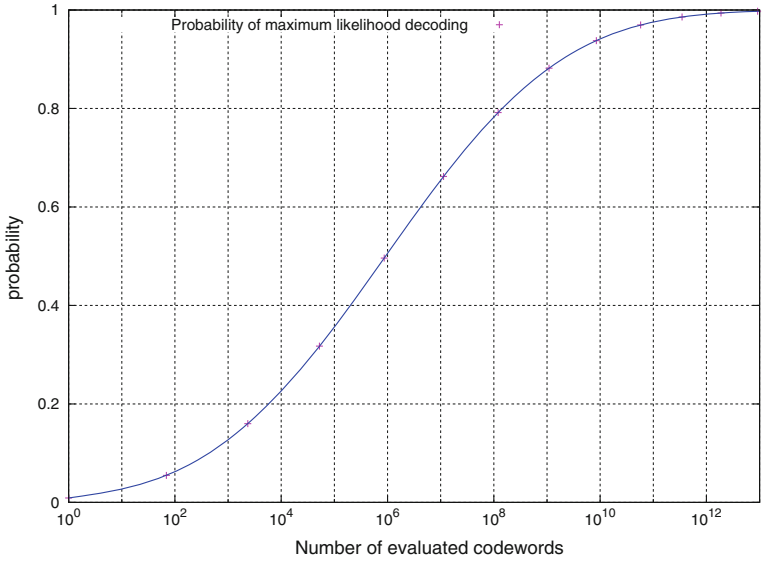

Fig. 15.2 Probability of a received vector being maximum likelihood decoded as a function of number of evaluated codewords for the (136, 68, 24) code at $\frac{E_b}{N_o} = 3.5$ dB

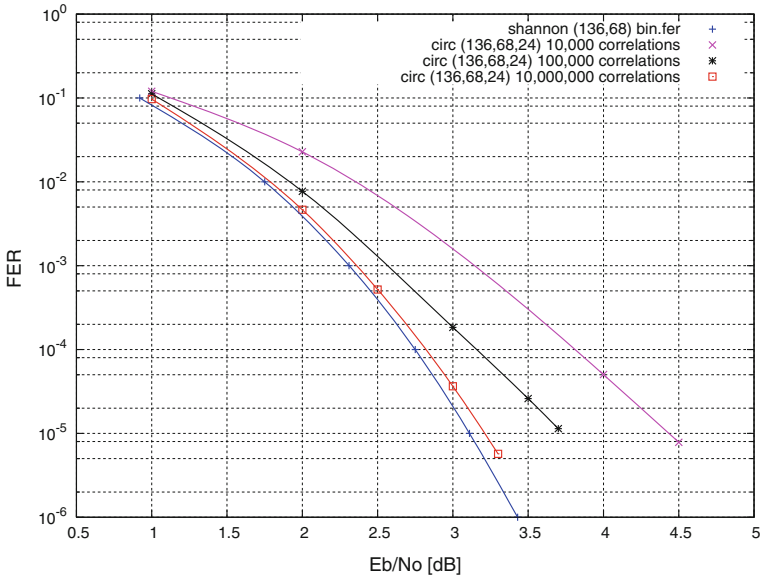

Fig. 15.3 FER performance of the (136, 68, 24) code as a function of number of evaluated codewords

Fig. 15.4 An example of received coordinate magnitudes in their solved order for the (136, 68, 24) code at $\frac{E_b}{N_o} = 2.5$ dB for a single received vector

The detailed operation of the decoder may be seen by considering an example of a received vector at $\frac{E_b}{N_o}$ of 2.5 dB. The magnitudes of the received coordinates, ordered in their solved order, is shown in Fig. 15.4. In this particular example, it is not possible to solve for ordered coordinates 67 and 68 (in their order prior to solving of the parity-check matrix) and so these coordinates are skipped and become coordinates 68 and 69, respectively, in the solved order. The transmitted bits are normalised with magnitudes 1 and the σ of the noise is $\approx$1.07. The shift in position of coordinate 69 (in original position) to 67 (in solved order) is evident in Fig. 15.4. The positions of the bits received in error in the same solved order is shown in Fig. 15.5. It may be noted that the received bit errors are concentrated in the least reliable bit positions. There are a total of 16 received bit errors and only two of these errors correspond to the (data) bit coordinates 11 and 34 of the solved $\mathbf{G}$ matrix. Evaluation of 10^7 codewords indicates that the minimum value of $Y_{\Delta}(\mathbf{x_{min}})$ is $\approx$13.8, and this occurs for the 640th codeword producing a maximum correlation of $\approx$126.2 with $Y_{max} \approx 140$. The weight of $\mathbf{z_{min}}$ is 16 corresponding to the 16 received bit errors.

In practice, it is not necessary for $Y_{\Delta}(\mathbf{x_i})$ given by the partial sum equation (15.18) to be evaluated for each codeword. In most cases, the weight of the binary vector $\mathbf{z_i}$ is sufficiently high to indicate that this codeword is not the most likely codeword. Shown in Fig. 15.6 are the cumulative probability distributions for the weight of $\mathbf{z_i}$ for the case where $\mathbf{x_i}$ is equal to the transmitted codeword, and the case where it is not equal to the transmitted codeword. Two operating values for $\frac{E_b}{N_o}$ are shown: 3.5 dB and 4 dB. Considering the decoding rule that a weight 29 or more for $\mathbf{z_i}$ is unlikely to be produced by the transmitted codeword means that 95.4% of candidate codewords

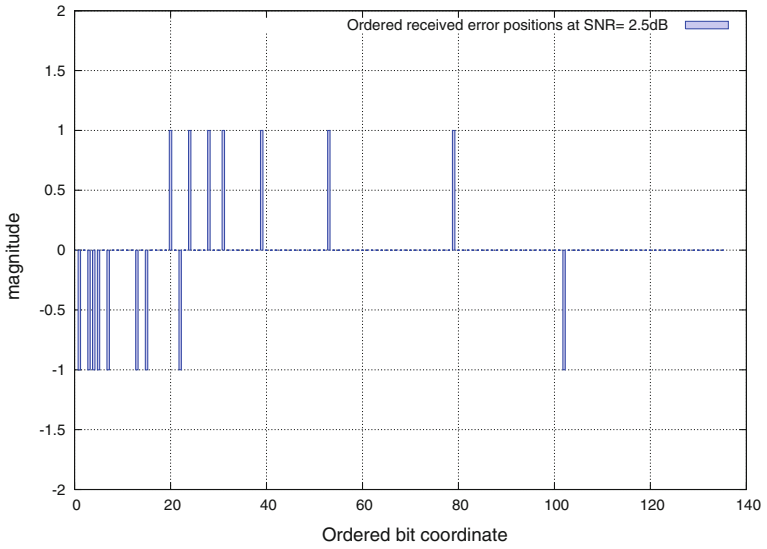

Fig. 15.5 Received bits showing bit error positions for the same received vector and same order as that shown in Fig. 15.4

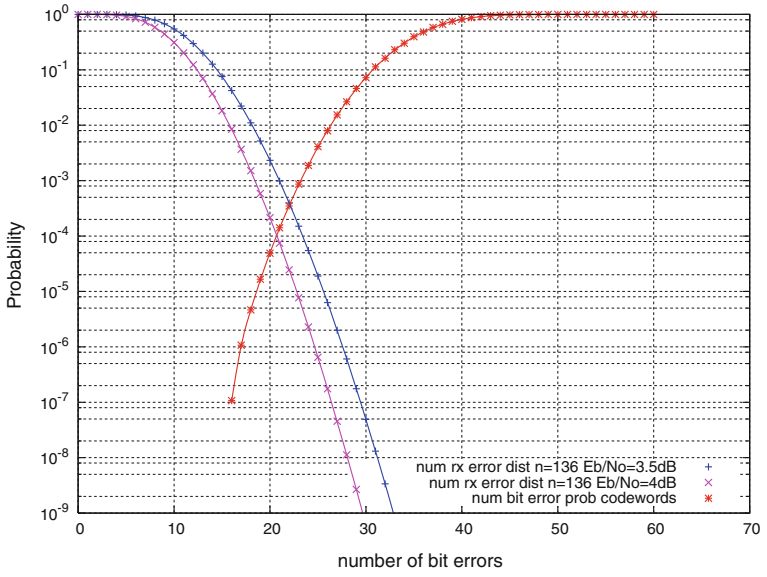

Fig. 15.6 Cumulative probability distributions for the number of bit errors for the transmitted codeword and non-transmitted, evaluated codewords for the (136, 68, 24) code

may be rejected at this point, and that the partial sum equation (15.18) need only be evaluated for 4.6% of the candidate codewords. In reducing the decoder complexity in this way, the degradation to the FER performance as a result of rejection of a transmitted codeword corresponds to ≈3% increase in the FER and is not significant.

15.4.2 The (255, 175, 17) Euclidean Geometry (EG) Code

This code is an EG code originally used in hard decision, one-step majority-logic decoding by Lin and Costello, Jr. [10]. Finite geometry codes also have applications as LDPC codes using iterative decoding with the belief propagation algorithm [9]. The (255, 175, 17) code is a cyclic code and its parity-check polynomial $p(x)$ may conveniently be generated from the cyclotomic idempotents as described in Chap. 12. The parity-check polynomial is

$$p(x) = 1 + x + x^3 + x^7 + x^{15} + x^{26} + x^{31} + x^{53} + x^{63} + x^{98} \tag{15.27}$$

$$+ x^{107} + x^{127} + x^{140} + x^{176} + x^{197} + x^{215} \tag{15.28}$$

The FER performance of the code is shown in Fig. 15.7 and was obtained using the incremental correlation decoder and is shown in comparison to using the iterative decoder. Also shown in Fig. 15.7 is the sphere packing bound offset by the binary transmission loss.

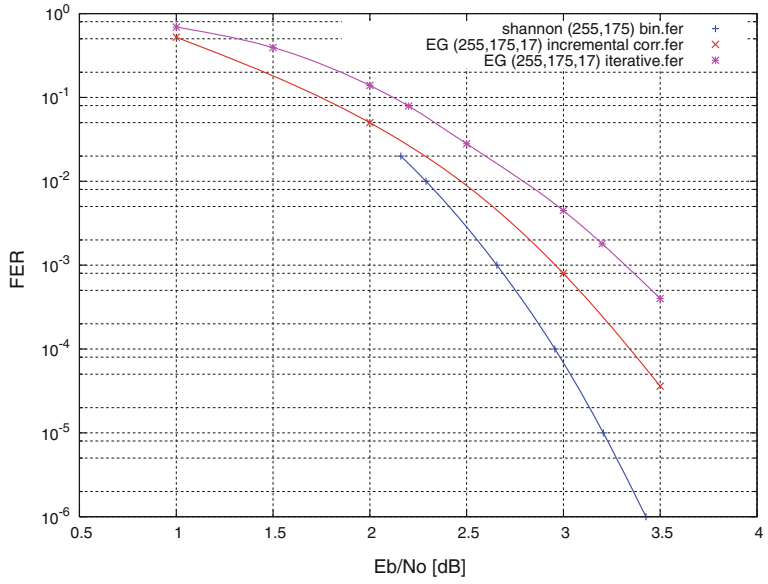

Fig. 15.7 FER performance of the (255, 175, 17) EG code using belief propagation, iterative decoding, compared to incremental correlation decoding

Although this EG code performs well with iterative decoding it is apparent that the incremental correlation decoder is able to improve the performance of the code for the AWGN channel by 0.45 dB at 10^{-3} FER.

15.4.3 The (513, 467, 12) Extended Binary Goppa Code

Goppa codes are frequently better than the corresponding BCH codes because there is an additional information bit and the Goppa code is only one bit longer than the BCH code. For example, the (512, 467, 11) binary Goppa Code has one more information bit than the (511, 466, 11) BCH code and may be generated by the irreducible Goppa polynomial $1 + x^2 + x^5$, whose roots have order 31 which is relatively prime to 511. The d_{min} of the binary Goppa code [12] is equal to twice the degree of the irreducible polynomial plus 1 and is the same as the (511, 466, 11) BCH code. The Goppa code may be extended by adding an overall parity check, increasing the d_{min} to 12.

The FER performance of the extended Goppa code is shown in Fig. 15.8 and was obtained using the incremental correlation decoder. Also shown in Fig. 15.8 is the sphere packing bound offset by the binary transmission loss. It can be seen that the realised performance of the decoder is within 0.3 dB at 10^{-4}.

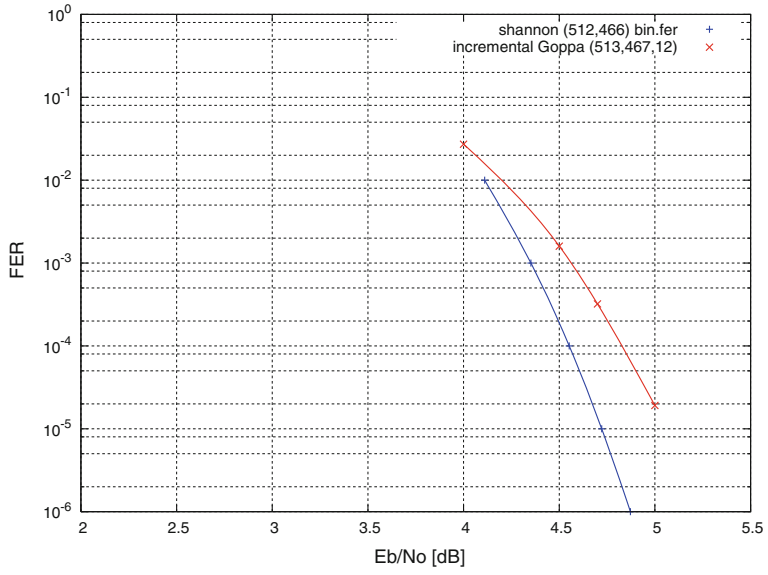

Fig. 15.8 FER performance of the (513, 467, 12) binary Goppa code using incremental correlation decoding

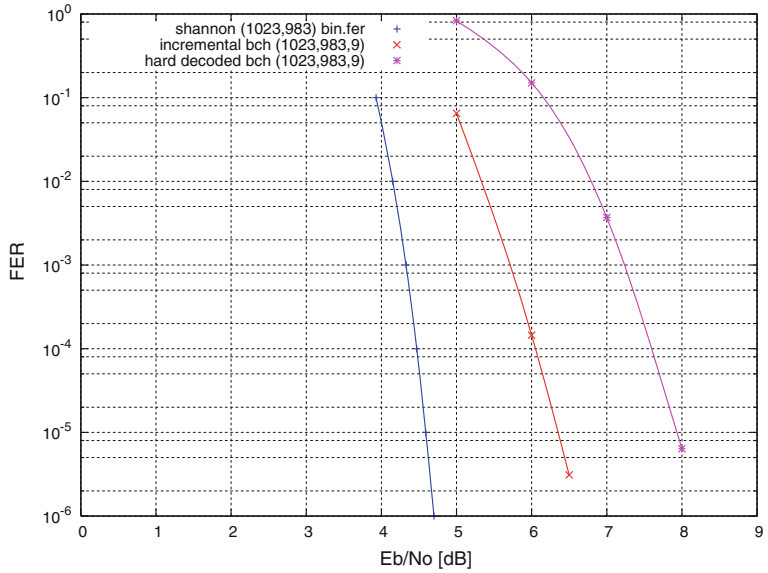

Fig. 15.9 FER performance of the (1023, 983, 9) binary BCH code using incremental correlation decoding compared to hard decision decoding

15.4.4 The (1023, 983, 9) BCH Code

This code is a standard BCH code that may be found in reference text book tables such as by Lin and Costello, Jr. [10]. This example is considered here in order to show that the decoder can produce near maximum likelihood performance for relatively long codes. The performance obtained is shown in Fig. 15.9 with evaluation of candidate codewords limited to 10^6 codewords. At 10^{-5} FER, the degradation from the sphere packing bound, offset for binary transmission, is 1.8 dB. Although this may seem excessive, the degradation of hard decision decoding is 3.6 dB as may also be seen from Fig. 15.9.

15.5 Extension to Non-binary Codes

The extension of the decoder to non-binary codes is relatively straightforward, and for simplicity binary transmission of the components of each non-binary symbol is assumed. Codewords are denoted as before by $\mathbf{x_i}$ but redefined with coefficients, $\gamma_{j\,i}$ from $GF(2^m)$

$$\mathbf{x_i} = (\gamma_{0\,i}\, x_0, \gamma_{1\,i}\, x_1, \gamma_{2\,i}\, x_2, \ldots, \gamma_{n-1\,i}\, x_{n-1}) \tag{15.29}$$

The received vector $\mathbf{r}$ with coordinates ranked in order of those most likely to be correct is redefined as

$$\mathbf{r} = \sum_{l=0}^{m-1} (r_{l\,\mu_0}, r_{l\,\mu_1}, r_{l\,\mu_2}, \ldots, r_{l\,\mu_{n-1}}) \tag{15.30}$$

so that the received vector consists of n symbols, each with m values. The maximum attainable correlation Y_{max} is straightforward and is given by

$$Y_{max} = \sum_{j=0}^{n-1} \sum_{l=0}^{m-1} |r_{l\,j}| \tag{15.31}$$

The hard decided received vector $\mathbf{r}$, is redefined as

$$\mathbf{b} = \sum_{j=0}^{n-1} \theta_j \, x^j \tag{15.32}$$

where θ_j is the $GF(2^m)$ symbol corresponding to $sign(r_{l\,j})$ for $l = 0$ to $m - 1$.

Decoding follows in a similar manner to the binary case. The received symbols are ordered in terms of their symbol magnitudes $|r_{\mu_j}|_S$ where each symbol magnitude is defined as

$$|r_{\eta_j}|_S = \sum_{l=0}^{m-1} |r_{l\,\eta_j}| \tag{15.33}$$

The codeword $\hat{\mathbf{x}}$ is derived from the k coordinates x_{η_j} whose coefficients v_{η_j} are the $GF(2^m)$ symbols corresponding to $sign(r_{l\,\eta_j})$ for $l = 0$ to $m - 1$; for $j = 0$ to $k - 1$ and then using the solved parity-check equations for the remaining $n - k$ coordinates.

The vector $\mathbf{z_i}$ is given by

$$\mathbf{z_i} = \mathbf{b} \oplus \hat{\mathbf{x}} \oplus \tilde{\mathbf{x}}_\mathbf{i} \mod GF(2^m) \tag{15.34}$$

which may be simplified as before to

$$\mathbf{z_i} = \hat{\mathbf{z}} \oplus \tilde{\mathbf{x}}_\mathbf{i} \mod GF(2^m) \tag{15.35}$$

Denoting the n binary vectors $\rho_{i\,l\,j}$ corresponding to the n $GF(2^m)$ coefficients of $\mathbf{z_i}$

$$Y(\mathbf{x_i}) = Y_{max} - Y_\Delta(\mathbf{x_i}) \tag{15.36}$$

where $Y_{\Delta}(\mathbf{x_i})$, the shortfall from maximum correlation is given by

$$Y_{\Delta}(\mathbf{x_i}) = 2 \sum_{j=0}^{n-1} \sum_{l=0}^{m-1} \rho_{ilj} \times |r_{lj}| \tag{15.37}$$

In the implementation of the decoder, as in the binary case, the Hamming weight of the vector $\mathbf{z_i}$ may be used to decide whether it is necessary to evaluate the soft decision metric given by Eq. (15.37) for each candidate codeword.

15.5.1 Results for the (63, 36, 13) GF(4) BCH Code

This is a non-binary BCH code with the generator polynomial $g(x)$ defined by roots

$$\{\alpha^1, \alpha^4, \alpha^{16}, \alpha^2, \alpha^8, \alpha^{32}, \alpha^3, \alpha^{12}, \alpha^{48}, \alpha^5, \alpha^{20}, \alpha^{17}, \alpha^6, \alpha^{24}, \alpha^{33},$$
$$\alpha^7, \alpha^{28}, \alpha^{29}, \alpha^9, \alpha^{36}, \alpha^{18}, \alpha^{10}, \alpha^{40}, \alpha^{34}, \alpha^{11}, \alpha^{44}, \alpha^{50}\}$$

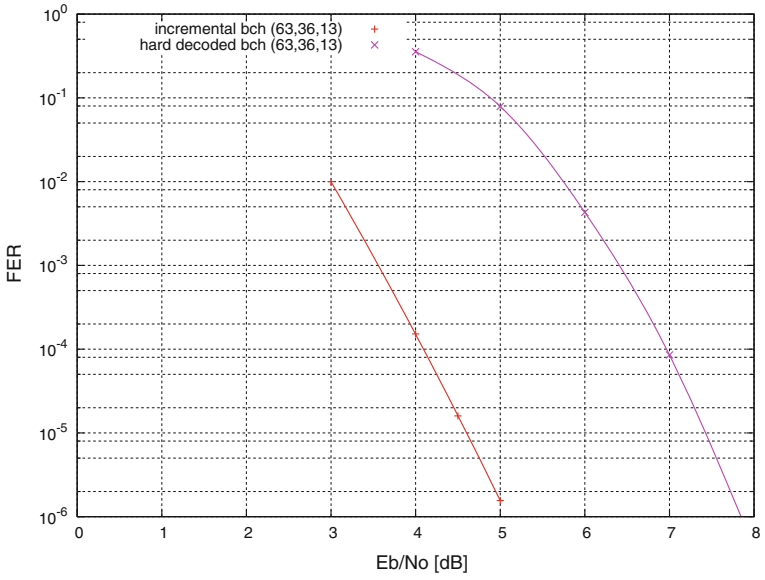

Fig. 15.10 FER performance of the (63, 36, 13) GF(4) BCH code using incremental correlation decoding compared to hard decision decoding

The benefit of having $GF(4)$ coefficients is that $g(x)$ does not need to contain the roots

$$\{\alpha^{14}, \alpha^{56}, \alpha^{35}, \alpha^{22}, \alpha^{25}, \alpha^{37}\}$$

which are necessary to constrain $g(x)$ to binary coefficients [12]. Correspondingly, the binary version of this BCH code is the lower rate (63, 30, 13) code with 6 less information symbols (bits).

The performance of the (63, 36, 13) $GF(4)$ BCH Code is shown in Fig. 15.10 for the AWGN channel using Quadrature Amplitude Modulation (QAM). Also shown in Fig. 15.10 is the performance of the code with hard decision decoding. It may be seen that at 10^{-4} FER the performance of the incremental correlation decoder is 2.9 dB better than the performance of the hard decision decoder.

15.6 Conclusions

It has been shown that the extended Dorsch decoder may approach maximum likelihood decoding by an incremental correlation approach in which for each received vector a partial summation metric is evaluated as a function of low information weight codewords. Furthermore, the number of information weight codewords that need to be evaluated to achieve maximum likelihood decoding may be calculated as an upper bound for each received vector. Consequently, for each received vector it is known whether the decoder has achieved maximum likelihood decoding. An efficient decoder structure consisting of a combination of hard decision threshold decoding followed by partial sum correlation was also described, which enables practical decoders to trade-off performance against complexity.

The decoder for non-binary codes was shown to be straightforward for the AWGN channel and an example was described for a GF(4) (63, 36, 13) BCH code using QAM to transmit each GF(4) symbol. It is readily possible to extend the decoder to other modulation formats by extensions to the incremental correlation of Eq. (15.37) although this inevitably involves an increase in complexity. It is hoped that there will sufficient interest from the coding community to address this research area.

Another interesting conclusion is just how well some codes in Brouwer's table perform with maximum likelihood decoding. In particular, the (136, 68, 24) double-circulant, extremal, self-dual code is shown to be an outstanding code.

It seems that the implementation of this type of decoder coupled with the availability of powerful processors will eventually herald a new era in the application of error control coding with the re-establishment of the importance of the optimality of codes rather than the ease of decoding. Certainly, this type of decoder is more complex than an iterative decoder, but the demonstrable performance, which is achievable for short codes, can approach theoretical limits for error-correction coding performance such as the sphere packing bound.

15.7 Summary

The current day, unobtainable goal of a practical realisation of the maximum likelihood decoder that can be used with any error-correcting code has been partially addressed with the description of the modified Dorsch decoder presented in this chapter. A decoder based on enhancements to the original Dorsch decoder has been described which achieves near maximum likelihood performance for all codes whose codelength is not too long. It is a practical decoder for half rate codes having a codelength less than about 180 bits or so using current digital processors. The performance achieved by the decoder when using different examples of outstanding binary codes has been evaluated and the results presented in this chapter. A description of the decoder suitable for use with non-binary codes has also been given. An example showing the results obtained by the decoder using a (63, 36, 13) GF(4) non-binary code for the AWGN channel has also been presented.

References

1. Butman, S., McEliece, R.J.: The ultimate limits of binary coding for a wideband Gaussian channel. JPL Deep Space Netw. Prog. Rep. **42–22**, 78–80 (1974)
2. Chase, D.: A class of algorithms for decoding block codes with channel measurement information. IEEE Trans. Inf. Theory IT **18**, 170–182 (1972)
3. Conway, J.H., Sloane, N.J.A.: A new upper bound on the minimum distance of self-dual codes. IEEE Trans. Inf. Theory **36**(6), 1319–1333 (1990)
4. Dorsch, B.G.: A decoding algorithm for binary block codes and J-ary output channels. IEEE Trans. Inf. Theory **20**, 391–394 (1974)
5. Fossorier, M., Lin, S.: Soft-decision decoding of linear block codes based on ordered statistics. IEEE Trans. Inf. Theory **41**(5), 1379–1396 (1995)
6. Fossorier, M., Lin, S.: Computationally efficient soft-decision decoding of linear block codes based upon ordered statistics. IEEE Trans. Inf. Theory **42**, 738–750 (1996)
7. Grassl, M.: Code Tables: Bounds on the parameters of various types of codes. http://www.codetables.de (2007)
8. Karlin, M.: New binary coding results by circulants. IEEE Trans. Inf. Theory **15**(1), 81–92 (1969)
9. Kou, Y., Lin, S., Fossorier, M.: Low-density parity-check codes based on finite geometries: a rediscovery and new results. IEEE Trans. Inf. Theory **47**(7), 2711–2736 (2001)
10. Lin, S., Costello Jr., D.J.: Error Control Coding: Fundamentals and Applications, 2nd edn. Pearson Education, Inc, Englewood Cliffs (2004)
11. Lous, N.J.C., Bours, P.A.H., van Tilborg, H.C.A.: On maximum likelihood soft-decision decoding of binary linear codes. IEEE Trans. Inf. Theory **39**, 197–203 (1993)
12. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
13. Proakis, J.: Digital Communications, 4th edn. McGraw-Hill (2001)
14. Shannon, C.E.: Probability of error for optimal codes in a Gaussian channel. Bell Syst. Tech. J. **38**(3), 611–656 (1959)
15. Snyders, J.: Reduced lists of error patterns for maximum likelihood soft decision decoding. IEEE Trans. Inf. Theory **37**, 1194–1200 (1991)

16. Tjhai, C.J., Tomlinson, M., Ambroze, M., Ahmed, M.: Cyclotomic idempotent-based binary cyclic codes. Electron. Lett. **41**(3), 341–343 (2005)
17. Tomlinson, M., Tjhai, C.J., Ambroze, M., Ahmed, M.: Improved error correction decoder using ordered symbol reliabilities. UK Patent Application GB0637114.3 (2005)