Chapter 16
A Concatenated Error-Correction System Using the $|u|u+v|$ Code Construction

16.1 Introduction

There is a classical error-correcting code construction method where two good codes are combined together to form a new, longer code. It is a method first pioneered by Plotkin [1]. The Plotkin sum, also known as the $|u|u+v|$ construction method [3], consists of one or more codes having replicated codewords to which are added codewords from one or more other codes to form a concatenated code. This code construction may be exploited in the receiver with a decoder that first decodes one or more individual codewords prior to the Plotkin sum from a received vector. The detected codewords from this first decoding are used to undo the code concatenation within the received vector to allow the replicated codewords to be decoded. The output from the overall decoder of the concatenated code consists of the information symbols from the first decoder followed by the information symbols from the second stage decoder. Multiple codewords may be replicated and added to the codewords from other codes so that the concatenated code consists of several shorter codewords which are decoded first and the decoded codewords used to decode the remaining codewords. It is possible to utilise a recurrent construction whereby the replicated codewords are themselves concatenated codewords. It follows that the receiver has to use more than two stages of decoding.

With suitable modifications, any type of error-correction decoder may be utilised including iterative decoders, Viterbi decoders, list decoders, and ordered reliability decoders, and of particular importance the modified Dorsch decoder described in Chap. 15. It is well known that for a given code rate longer codes have better performance than shorter codes, but implementation of a maximum likelihood decoder is much more difficult for longer codes. The Plotkin sum code construction method provides a means whereby several decoders for short codes may be used together to implement a near maximum likelihood decoder for a long code.

M. Tomlinson et al., *Error-Correction Coding and Decoding*,
Signals and Communication Technology,
DOI 10.1007/978-3-319-51103-0_16

16.2 Description of the System

Figure 16.1 shows the generic structure of the transmitted signal in which the codeword of length n_1 from code u, denoted as $\mathscr{C}_u$ is followed by a codeword comprising the sum of the same codeword and another codeword from code v, denoted as $\mathscr{C}_v$ to form a codeword denoted as $\mathscr{C}_{cat}$ of length $2n_1$. This code construction is well known as the $|u|u+v|$ code construction [3]. The addition is carried out symbol by symbol using the arithmetic rules of the Galois Field being used, namely $GF(q)$. If code u is an (n_1, k_1, d_1) code with k_1 information symbols and Hamming distance d_1 and code v is an (n_1, k_2, d_2) code with k_2 information symbols and Hamming distance d_2, the concatenated code $\mathscr{C}_{cat}$ is an $(2n_1, k_1 + k_2, d_3)$ code with Hamming distance d_3 equal to the smaller of $2 \times d_1$ and d_2.

Prior to transmission, symbols from the concatenated codeword are mapped to signal constellation points in order to maximise the Euclidean distance between transmitted symbols in keeping with current best transmission practice. For example see the text book by Professor J. Proakis [4]. The mapped concatenated codeword is denoted as $\mathscr{X}_{cat}$ and is given by

$$\mathscr{X}_{cat} = |\mathscr{X}_u|\mathscr{X}_{u+v}| = |\mathscr{X}_u|\mathscr{X}_w|, \tag{16.1}$$

where $\mathscr{X}_w$ is used to represent $\mathscr{X}_{u+v}$.

$\mathscr{X}_{cat}$ consists of $2 \times n_1$ symbols and the first n_1 symbols are the n_1 symbols of $\mathscr{X}_u$ and the second n_1 symbols are the n_1 symbols resulting from mapping of the symbols resulting from the summation, symbol by symbol, of the n_1 symbols of $\mathscr{C}_u$, and the n_1 symbols of codeword $\mathscr{C}_v$.

The encoding system to produce the concatenated codeword format shown in Fig. 16.1 is shown in Fig. 16.2. For each concatenated codeword, k_1 information symbols are input to the encoder for the (n_1, k_1, d_1) code and n_1 symbols are produced at the output of the encoder and are stored in the codeword buffer A as shown in Fig. 16.2. Additionally, for each concatenated codeword, k_2 information symbols are input to the encoder for the (n_1, k_2, d_2) code and n_1 symbols are produced at the output and are stored in the codeword buffer B as shown in Fig. 16.2. The encoded symbols

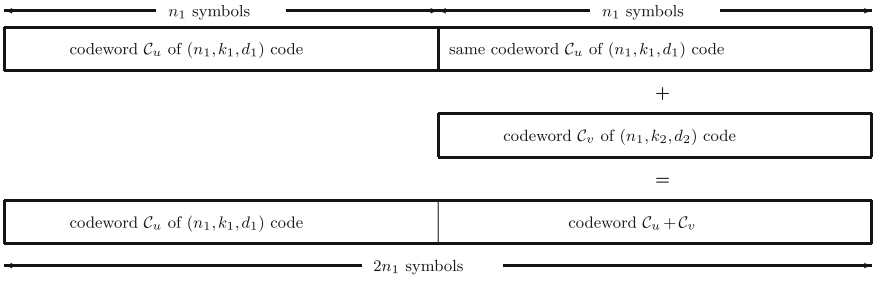

Fig. 16.1 Format of transmitted codeword consisting of two shorter codewords

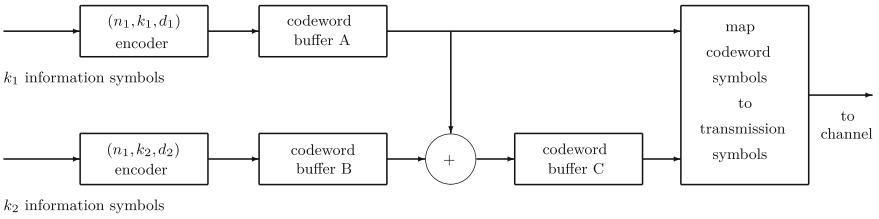

Fig. 16.2 Concatenated code encoder and mapping for transmission

output from the codeword buffer A are added symbol by symbol to the encoded symbols output from the codeword buffer B and the results are stored in codeword buffer C. The codeword stored in codeword buffer A is $\mathscr{C}_u$ as depicted in Fig. 16.1 and the codeword stored in codeword buffer C is $\mathscr{C}_u + \mathscr{C}_v$ as also depicted in Fig. 16.1. The encoded symbols output from the codeword buffer A are mapped to transmission symbols and transmitted to the channel, and these are followed sequentially by the symbols output from the codeword buffer C which are also mapped to transmission symbols and transmitted to the channel as shown in Fig. 16.2.

After transmission through the communications medium each concatenated mapped codeword is received as the received vector, denoted as $\mathscr{R}_{cat}$ and given by

$$\mathscr{R}_{cat} = |\mathscr{R}_u|\mathscr{R}_{u+v}| = |\mathscr{R}_u|\mathscr{R}_w|. \tag{16.2}$$

Codeword $\mathscr{C}_v$ is decoded first as shown in Fig. 16.3. It is possible by comparing the received samples $\mathscr{R}_u$ with the received samples $\mathscr{R}_{u+v}$ that the a priori log likelihoods of the symbols of $\mathscr{R}_v$ may be determined, since it is clear that the difference between the respective samples, in the absence of noise and distortion, is attributable to $\mathscr{C}_v$. This is done by the soft decision metric calculator shown in Fig. 16.3.

Binary codeword symbols are considered with values which are either 0 or 1. The i$^{\text{th}}$ transmitted sample, $X_{u_i} = (-1)^{C_{u_i}}$ and the n_1 + i$^{\text{th}}$ transmitted sample, $X_{u_i+v_i} = (-1)^{C_{u_i}} \times (-1)^{C_{v_i}}$. It is apparent that X_{v_i} and C_{v_i} may be derived from X_{u_i} and $X_{u_i+v_i}$.

An estimate of X_{v_i} and C_{v_i} may be derived from R_{u_i} and $R_{u_i+v_i}$. First:

$$X_{v_i} = X_{u_i} \times X_{u_i+v_i} = (-1)^{C_{u_i}} \times (-1)^{C_{u_i}} \times (-1)^{C_{v_i}} = (-1)^{C_{v_i}} \tag{16.3}$$

Second, in the absence of distortion and with Gaussian distributed additive noise with standard deviation σ, and normalised signal power, the log likelihood that $C_{v_i} = 0$, $L_{log}(C_{v_i} = 0)$ is given by

$$L_{\log}(C_{v_i} = 0) = \log\left[\cosh\left(\frac{R_{u_i} + R_{u_i+v_i}}{\sigma^2}\right)\right] - \log\left[\cosh\left(\frac{R_{u_i} - R_{u_i+v_i}}{\sigma^2}\right)\right]. \tag{16.4}$$

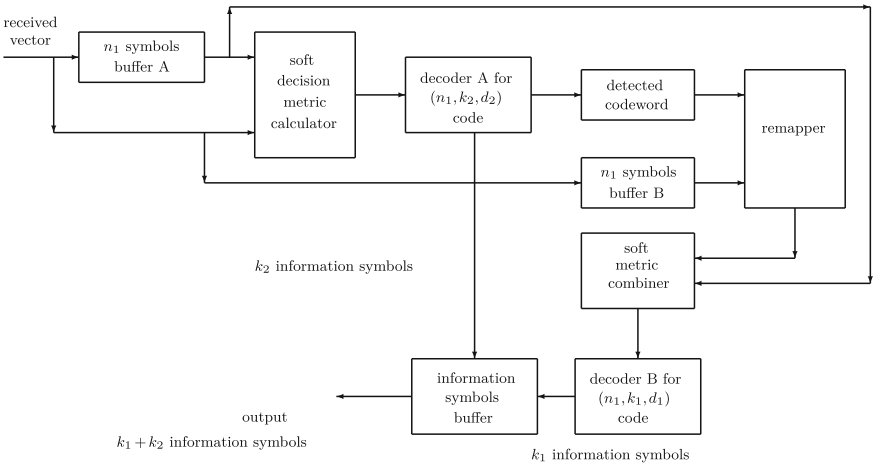

Fig. 16.3 Decoder for the concatenated code with the codeword format shown in Fig. 16.1

The soft decision metric calculator, shown in Fig. 16.3, calculates these log likelihoods according to Eq. (16.4) and these are input to the decoder A shown in Fig. 16.3. The decoder A determines the most likely codeword $\mathscr{C}_{\hat{v}}$ of the (n_1, k_2, d_2) code. With the knowledge of the detected codeword, $\mathscr{C}_{\hat{v}}$, the received samples $\mathscr{R}_{u+v}$, which are stored in the n_1 symbols buffer B, are remapped to form $\mathscr{R}_{\hat{u}}$ by multiplying $\mathscr{R}_{u+v}$ by $\mathscr{X}_{\hat{v}}$.

$$\mathscr{R}_{\hat{u}} = \mathscr{R}_{u+v} \times \mathscr{X}_{\hat{v}}. \tag{16.5}$$

This remapping function is provided by the remapper shown in Fig. 16.3. The output of the remapper is $\mathscr{R}_{\hat{u}}$. If the decoder's output is correct, $\mathscr{C}_{\hat{v}} = \mathscr{C}_v$ and there are now two independent received versions of the transmitted, mapped codeword $\mathscr{C}_u$, $\mathscr{R}_{\hat{u}}$ and the original received $\mathscr{R}_u$. Both of these are input to the soft metric combiner shown in Fig. 16.3, $\mathscr{R}_{\hat{u}}$ from the output of the remapper and $\mathscr{R}_u$ from the output of the n_1 symbols buffer A.

The soft metric combiner calculates the log likelihood of each bit of $\mathscr{C}_u$, C_{u_i} from the sum of the individual log likelihoods:

$$L_{\log}(C_{u_i} = 0) = \frac{2R_{u_i}}{\sigma^2} + \frac{2R_{\hat{u}_i}}{\sigma^2}. \tag{16.6}$$

These log likelihood values, $L_{\log}(C_{u_i} = 0)$, output from the soft metric combiner shown in Fig. 16.3 are input to the decoder B. The output of Decoder B is the k_1 information bits of the detected codeword $\mathscr{C}_{\hat{u}}$ of the (n_1, k_1, d_1) code, and these are input to the information symbols buffer shown in Fig. 16.3. The other input to the information symbols buffer is the k_2 information bits of the detected codeword

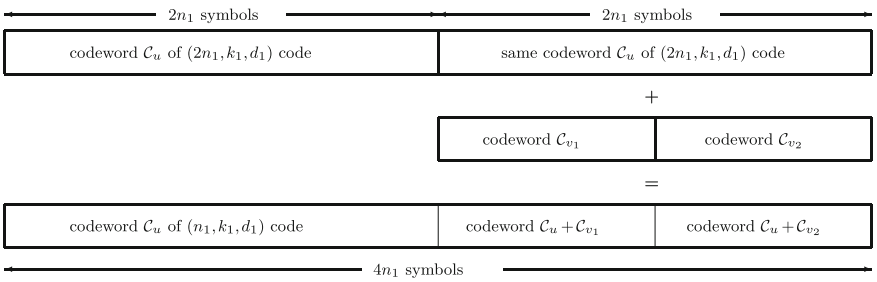

Fig. 16.4 Format of transmitted codeword consisting of three shorter codewords

$\mathscr{C}_{\hat{v}}$ of the (n_1, k_2, d_2) code, provided at the output of decoder A. The output of the information symbols buffer, for each received vector, is the $k_1 + k_2$ information bits which were originally encoded, provided both decoders' outputs, A and B, are correct.

In similar fashion to previous constructions, Fig. 16.4 shows the format of a concatenated codeword of length $4 \times n_1$ symbols consisting of three shorter codewords. The codeword of length $2 \times n_1$ from a $(2n_1, k_1, d_1)$, code u, denoted as $\mathscr{C}_u$ is replicated as shown in Fig. 16.4. The first half of the replicated codeword, $\mathscr{C}_u$, is added to the codeword $\mathscr{C}_{v1}$ and the second half of the replicated codeword, $\mathscr{C}_u$, is added to the codeword $\mathscr{C}_{v2}$, as shown in Fig. 16.4. Each codeword $\mathscr{C}_{v1}$ and $\mathscr{C}_{v2}$ is the result of encoding k_2 information symbols using code v, a (n_1, k_2, d_2) code. The concatenated codeword that results, $\mathscr{C}_{cat}$, is from a $(4n_1, k_1 + 2k_2, d_3)$ concatenated code where d_3 is the smaller of $2d_1$ or d_2.

The decoder for the concatenated code with codeword format shown in Fig. 16.4 is similar to the decoder shown in Fig. 16.3 except that following soft decision metric calculation each of the two codewords $\mathscr{C}_{v_1}$ and $\mathscr{C}_{v_2}$ are decoded independently. With the knowledge of the detected codewords, $\mathscr{C}_{\hat{v}_1}$ and $\mathscr{C}_{\hat{v}_2}$, the received samples $\mathscr{R}_{u+v_1}$, which are buffered, are remapped to form the first n_1 symbols of $\mathscr{R}_{\hat{u}}$ by multiplying $\mathscr{R}_{u+v_1}$ by $\mathscr{X}_{\hat{v}_1}$ and the second n_1 symbols of $\mathscr{R}_{\hat{u}}$ are obtained by multiplying $\mathscr{R}_{u+v_2}$ by $\mathscr{X}_{\hat{v}_2}$.The two independent received versions of the transmitted, mapped codeword $\mathscr{C}_u$, $\mathscr{R}_{\hat{u}}$ and the original received $\mathscr{R}_u$ are input to a soft metric combiner prior to decoding the codeword $\mathscr{C}_{\hat{u}}$.

In another code arrangement, Fig. 16.5 shows the format of a concatenated codeword of length $3 \times n_1$ symbols. The concatenated codeword is the result of three layers of concatenation. A codeword of length n_1 from a (n_1, k_1, d_1), code u, denoted as $\mathscr{C}_u$ is replicated twice, as shown in Fig. 16.5. A second codeword of length n_1 from a (n_1, k_2, d_2), code v, denoted as $\mathscr{C}_v$ is replicated and each of these two codewords is added to the two replicated codewords $\mathscr{C}_u$, as shown in Fig. 16.5. A third codeword of length n_1 from a (n_1, k_3, d_3), code w, denoted as $\mathscr{C}_w$ is added to the codeword summation $\mathscr{C}_u + \mathscr{C}_v$, as shown in Fig. 16.5. The concatenated codeword that results, $\mathscr{C}_{cat}$, is from a $(3n_1, k_1 + k_2 + k_3, d_4)$ concatenated code where d_4 is the smallest of $3d_1$ or $2d_2$ or d_3.

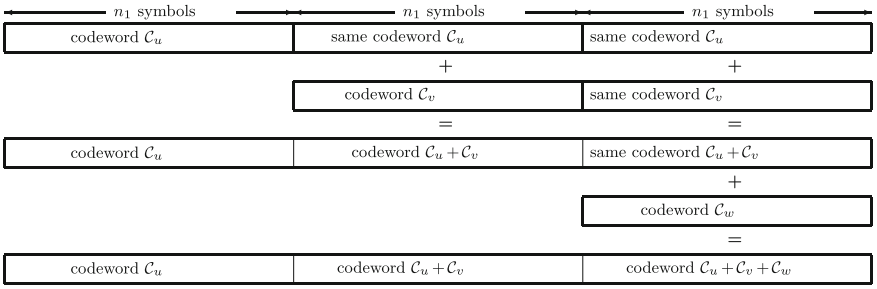

Fig. 16.5 Format of transmitted codeword consisting of two levels of concatenation and three shorter codewords

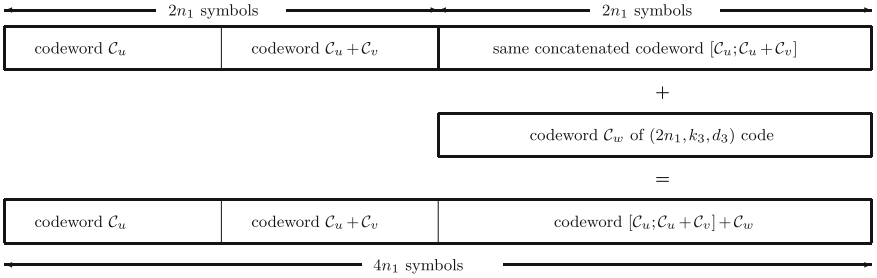

Fig. 16.6 Format of transmitted codeword after two stages of concatenation

The decoder for the three layered concatenated code with codeword format shown in Fig. 16.5 uses similar signal processing to the decoder shown in Fig. 16.3 with changes corresponding to the three layers of concatenation. The codeword $\mathscr{C}_{\hat{w}}$ is decoded first following soft decision metric calculation using the$\mathscr{R}_{u+v}$ and $\mathscr{R}_{u+v+w}$ sections of the received vector. The detected codeword $\mathscr{C}_{\hat{w}}$ is used to obtain two independent received versions of the transmitted, mapped result of the two codewords summation $\mathscr{C}_{u+v}$, $\mathscr{R}_{u+\hat{v}}$ and the original received $\mathscr{R}_{u+v}$. These are input to a soft metric combiner and the output is input to the soft decision metric calculation together with $\mathscr{R}_u$, prior to decoding of codeword $\mathscr{C}_{\hat{v}}$. With the knowledge of codeword $\mathscr{C}_{\hat{v}}$, remapping and soft metric combining is carried out prior to the decoding of codeword $\mathscr{C}_{\hat{u}}$.

Figure 16.6 shows the format of a concatenated codeword of length $4 \times n_1$ symbols. The concatenated codeword is the result of three layers of concatenation. A concatenated codeword with the format shown in Fig. 16.1 is replicated and added to a codeword, $\mathscr{C}_{\hat{w}}$, of length $2n_1$ symbols from a $(2n_1, k_3, d_3)$ code to form a codeword of an overall concatenated code having parameters $(4n_1, k_1 + k_2 + k_3, d_4)$, where d_4 is equal to the smallest of $4d_1$, $2d_2$ or d_3.

The decoder for the three layered concatenated code with codeword format shown in Fig. 16.6 is similar to the decoder described above. Codeword $\mathscr{C}_{\hat{w}}$ is detected first following soft decision metric calculation using $\mathscr{R}_u$ and $\mathscr{R}_{u+v}$ sections of the

received vector as one input and the $\mathscr{R}_{u+w}$ and $\mathscr{R}_{u+v+w}$ sections of the received vector as the other input. The detected codeword $\mathscr{C}_{\hat{w}}$ is used to obtain two independent received versions of the concatenated codeword of length $2n_1$ symbols with format equal to that of Fig. 16.1. Accordingly, following soft metric combining of the two independent received versions of the concatenated codeword of length $2n_1$ symbols, a vector of length equal to $2n_1$ symbols is obtained which may be input to the concatenated code decoder shown in Fig. 16.3. This decoder provides at its output the $k_1 + k_2$ detected information symbols which together with the k_3 information symbols already detected provide the complete detected output of the overall three layer concatenated code.

Any type of code, binary or non-binary, LDPC, Turbo or algebraically constructed code, may be used. Any corresponding type of decoder, for example an iterative decoder or a list decoder may be used. As an illustration of this, Decoder A and Decoder B, shown in Fig. 16.3, do not have to be the same type of decoder.

There are particular advantages in using the modified Dorsch decoder, described in Chap. 15, because the Dorsch decoder may realise close to maximum likelihood decoding, with reasonable complexity of the decoder. The complexity increases exponentially with codelength. Using modified Dorsch decoders. Both decoder A and decoder B shown in Fig. 16.3 operate on n_1 received samples and may realise close to maximum likelihood decoding with reasonable complexity even though the concatenated codelength is $2 \times n_1$ symbols and the total number of received samples is $2 \times n_1$ samples. Using a single modified Dorsch decoder to decode the $2 \times n_1$ samples of the concatenated code directly will usually result in non-maximum likelihood performance unless the list of codewords evaluated for each received vector is very long. For example, a modified Dorsch decoder with moderate complexity, typically will process 100,000 codewords for each received vector and realise near maximum likelihood performance. Doubling the codelength will require typically in excess of 100,000,000 codewords to be processed for each received vector if near maximum likelihood performance is to be maintained.

An example of the performance that may be achieved is shown in Fig. 16.7 for the concatenated codeword format shown in Fig. 16.1. The encoder used is the same as that shown in Fig. 16.2 and the concatenated code decoder is the same as that shown in Fig. 16.3. The results were obtained by computer simulation using Quaternary Phase Shift Keying (QPSK) modulation and featuring the Additive White Gaussian Noise (AWGN) channel. The decoder error rate, the ratio of the number of incorrect codewords output by the decoder to the total number of codewords output by the decoder, is denoted by the Frame Error Rate (FER) and this is plotted against $\frac{E_b}{N_o}$, the ratio of the energy per information bit to the noise power spectral density. Binary codes are used and the length of the concatenated code is 256 bits. For best results, it is important to use outstanding codes for the constituent codes, particularly for code v which is decoded first. In this example, code u is the (128,92,12) extended Bose Chaudhuri Hocquenghem (BCH) code. Code v is the (128,36,36) extended cyclic code, an optimum code described in [5] by D. Schoemaker and M. Wirtz. The (128,36,36) extended cyclic code is not an extended BCH code as it has roots {1, 3, 5, 7, 9, 11, 13, 19, 21, 27, 43, 47, 63}. The minimum Hamming distance

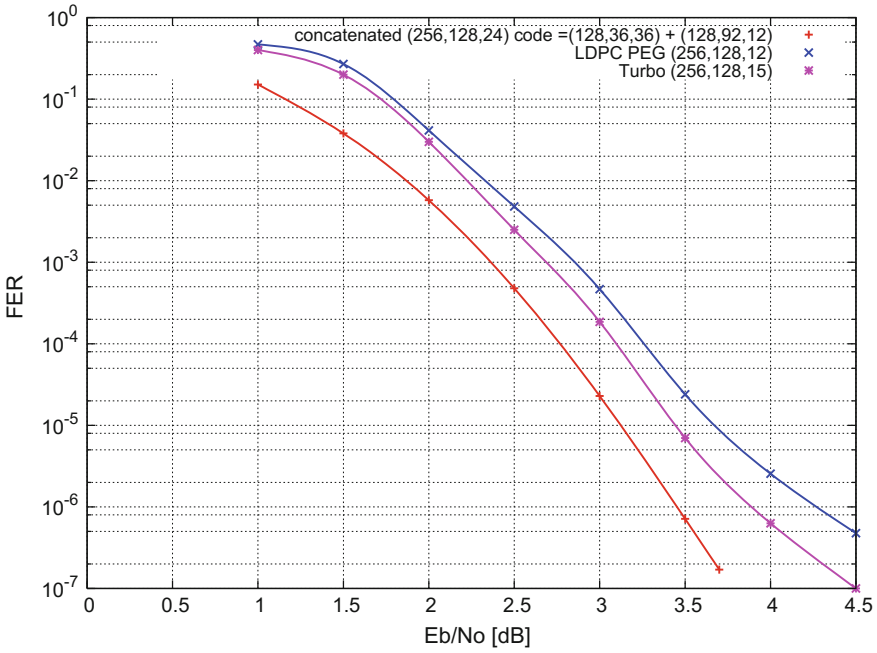

Fig. 16.7 The error rate performance for a (256,128,24) concatenated code compared to iterative decoding of a (256,128,15) Turbo code and a (256,128,12) LDPC code

of the concatenated code is $2d_1 = 24$. Both decoder A and decoder B, as shown in Fig. 16.3, are a modified Dorsch decoder and for both code u and code v, near maximum likelihood performance is obtained with moderate decoder complexity. For each point plotted in Fig. 16.7, the number of codewords transmitted was chosen such that were at least 100 codewords decoded in error.

Also shown in Fig. 16.7 is the performance of codes and decoders designed according to the currently known state of the art in error-correction coding that is Low Density Parity Check (LDPC) codes using Belief Propagation (BP) iterative decoding, and Turbo codes with BCJR iterative decoding. Featured in Fig. 16.7 is the performance of an optimised Low Density Parity Check (LDPC) (256,128,12) code using BP, iterative decoding and an optimised (256,128,15) Turbo code with iterative decoding. As shown in Fig. 16.7 both the (256,128,15) Turbo code and the (256,128,12) LDPC code suffer from an error floor for $\frac{E_b}{N_o}$ values higher than 3.5dB whilst the concatenated code features a FER performance with no error floor. This is attributable to the significantly higher minimum Hamming distance of the concatenated code which is equal to 24 in comparison to 15 for the Turbo code and 12 for the LDPC code. Throughout the entire range of $\frac{E_b}{N_o}$ values the concatenated code can be seen to outperform the other codes and decoders.

For (512,256) codes, using the concatenated code arrangement, the performance achievable is shown in Fig. 16.8. The concatenated code arrangement uses the

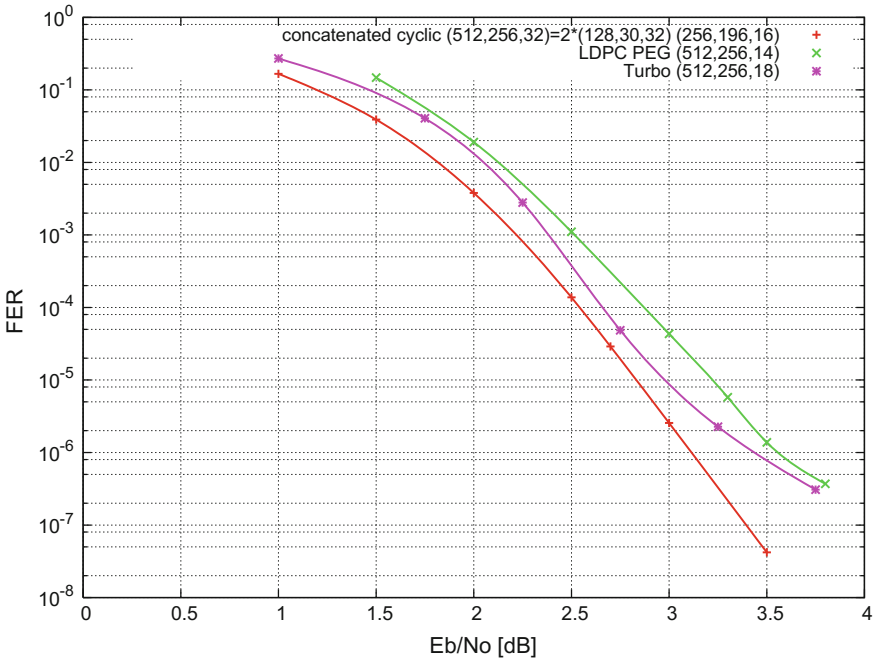

Fig. 16.8 Comparison of the error rate performance for a (512,256,32) concatenated code compared to iterative decoding of a (512,256,18) Turbo code and a (512,256,14) LDPC code

concatenated codeword format which is shown in Fig. 16.4. As before, the FER results were obtained by computer simulation using QPSK modulation and the AWGN channel. Both codes v_1 and v_2 are the same and equal to the outstanding (128,30,38) best-known code [6]. Code u is equal to a (256,196,16) extended cyclic code. Featured in Fig. 16.8 is the performance of an optimised Low Density Parity Check (LDPC) (512,256,14) code using BP iterative decoding and an optimised (512,256,18) Turbo code with iterative decoding. For each point plotted in Fig. 16.8, the number of codewords transmitted was chosen such that were at least 100 codewords decoded in error. As shown in Fig. 16.8 both the (512,256,18) Turbo code and the (512,256,14) LDPC code suffer from an error floor for $\frac{E_b}{N_o}$ values higher than 3.4 dB whilst the concatenated code features a FER performance with no error floor. As before this is attributable to the significantly higher minimum Hamming distance of the concatenated code which is equal to 32 in comparison to 18 for the Turbo code and 14 for the LDPC code. Throughout the entire range of $\frac{E_b}{N_o}$ values, the concatenated code system can be seen to outperform the other coding arrangements for (512,256) codes.

16.3 Concatenated Coding and Modulation Formats

With the $|u|u + v|$ code construction and binary transmission, the received vector for the codeword of code v suffers the full interference from the codeword of code u because it is transmitted as $u + v$. The interference is removed by differential detection using the first version of the codeword of code u. However, although the effects of code u are removed, differential detection introduces additional noise power due to noise times noise components. One possible solution to reduce this effect is to use multi-level modulation such as 8-PSK. Code u is transmitted as 4-PSK and code v modulates the 4-PSK constellation by ± 22.5 degrees. Now there is less direct interference between code u and code v. Initial investigations show that this approach is promising, particularly for higher rate systems.

16.4 Summary

Concatenation of good codes is a classic method of constructing longer codes which are good. As codes are increased in length, it becomes progressively harder to realise a near maximum likelihood decoder. This chapter presented a novel concatenated code arrangement featuring multiple near maximum likelihood decoders for an optimised matching of codes and decoders. It was demonstrated that by using some outstanding codes as constituent codes, the concatenated coding arrangement is able to outperform the best LDPC and Turbo coding systems with the same code parameters. The performance of a net (256,128) code achieved with the concatenated arrangement is compared to a best (256,128) LDPC code and a best (256,128) Turbo code. Similarly, the performance of a (512,256) net concatenated code is compared to a best (512,256) LDPC code and a best (512,256) Turbo code. In both cases, the new system was shown to outperform the LDPC and Turbo systems. To date, for the AWGN channel and net, half rate codes no other codes or coding arrangement is known that will outperform the system presented in this chapter for codes of lengths 256 and 512 bits.

References

1. Plotkin, M.: Binary codes with specified minimum distances. IEEE Trans. Inf. Theory **6**, 445–450 (1960)
2. Tomlinson, M., Tjhai, C.J., Ambroze, M.: Extending the Dorsch decoder towards achieving maximum-likelihood decoding for linear codes. IET Proc. Commun. **1**(3), 479–488 (2007)
3. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland (1977)
4. Proakis, J.: Digital Communications, 4th edn. McGraw-Hill, New York (2001)

5. Schomaker, D., Wirtz, M.: On binary cyclic codes of odd lengths from 101 to 127. IEEE Trans. Inf. Theory **38**(2), 516–518 (1992)
6. Grassl, M.: Code Tables: Bounds on the Parameters of Various Types of Codes. http://www.codetables.de