

UNIVERSITY OF HAWAII LIBRARY  
ITERATIVE MULTISTAGE MAXIMUM LIKELIHOOD  
DECODING ALGORITHM FOR MULTILEVEL CODES  
AND ITS APPLICATIONS

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE  
UNIVERSITY OF HAWAII IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY  
IN  
ELECTRICAL ENGINEERING

AUGUST 2003

By  
Diana Stojanović

Dissertation Committee:

Shu Lin, Chairperson  
Marc P.C. Fossorier  
N. Thomas Gaarder  
Wesley W. Peterson  
Edward Bertram

## **Acknowledgement**

I would like to thank NASA and NSF for supporting my studying while at the Electrical Engineering Department of the University of Hawai'i at Mānoa. During the graduate program, I benefited from the professional guidance of my dissertation advisor, Dr. Shu Lin, to whom I owe much of my knowledge in Error Control Codes. In addition, I would like to thank the members of my Ph.D. committee, Dr. Marc Fossorier, Dr. Tom Gaarder, Dr. Wesley Peterson, and Dr. Edward Bertram for their teaching and advice. Many thanks go to Ms. Bernie Todd and Ms. Janice Oda-Ng of the Office of Electrical Engineering. It goes without saying that this dissertation could not have been completed without the loving support of my family and friends. For that I will always be grateful.

# Abstract

The role of channel coding in digital communication system is to provide reliability, that is, a successful information transmission in the presence of noise and interference, with as small an error rate as required.

Half a century of research in Information Theory and Communications resulted in construction of many good codes and classes of codes. In general, longer codes achieve better performance, but the required time, memory, and amount of computation needed for successful decoding of these codes may in practice be infeasible. Thus the search for an efficient decoding algorithm is as important as the search for a good code. A good trade-off between the performance, measured by the low error probability, and efficiency, measured by the low decoding complexity, is set as a criterion.

Multistage decoding is devised for decoding codes with multilevel structure to achieve an efficient trade-off between error performance and decoding complexity. Multilevel code structure is used to simplify decoding. Component codes are decoded level-by-level in series of decoding stages, with the decoded information passed between them. Optimal for the codes of small and medium lengths and number of decoding stages, this technique shows a significant drop in performance when applied to longer codes, thus sacrificing performance to achieve efficiency.

In this dissertation, we develop an efficient soft-decision iterative multistage decoding algorithm for decoding decomposable and multilevel concatenated codes. This

algorithm achieves maximum likelihood (ML) performance through iterations with optimality tests at each decoding stage. It is the first proposed multistage decoding algorithm that achieves ML performance, and at the same time has a significant reduction in average decoding complexity compared to other known ML decoding algorithms, such as Viterbi algorithm.

The application of the algorithm to two general classes of multilevel codes, decomposable linear block codes, on the example of Reed-Muller codes, and multilevel block coded modulation codes, is presented. The results show that this new algorithm achieves excellent performance-complexity trade-off.

# Table of Contents

<b>Acknowledgement .....</b>	<b>iii</b>
<b>Abstract .....</b>	<b>iv</b>
<b>List of Tables .....</b>	<b>ix</b>
<b>List of Figures.....</b>	<b>x</b>
<b>Chapter 1 Introduction .....</b>	<b>1</b>
<b>Chapter 2 Concatenated Codes.....</b>	<b>9</b>
2.1 Structure of concatenated codes .....	10
2.1.1. Parameters and properties .....	12
2.2 Encoding of concatenated codes .....	14
2.3 Decoding of concatenated codes .....	15
2.4 Types of concatenated schemes based on component codes .....	16
2.4.1 Orthogonal codes as inner codes .....	16
2.4.2 Block codes as inner codes.....	17
2.4.3 Convolutional codes as inner codes.....	18
<b>Chapter 3 Multilevel Concatenation.....</b>	<b>20</b>
3.1 Structure of multilevel concatenated codes.....	20
3.2 Multilevel Concatenated Codes.....	25
3.3 Encoding Algorithm and parameters of MCC .....	27
3.4 Multilevel Modulation Codes (MMC).....	28
3.5 Decomposition of RM codes as MCC .....	29

<b>Chapter 4 Multistage Decoding .....</b>	<b>34</b>
4.1 Multistage Decoding Algorithm for multilevel concatenated codes.....	36
4.2 Multistage Decoding Algorithm for multilevel modulation codes.....	45
4.3 Examples: Multistage decoding of RM codes.....	48
4.4 Performance comparison of MSD and ML decoding.....	57
4.5 List Multistage Decoding (LMSD) Algorithm.....	60
4.6 Examples: List Multistage Decoding of RM Codes.....	62
<b>Chapter 5 Iterative Multistage Maximum Likelihood Decoding Algorithm .....</b>	<b>67</b>
5.1 Review of the conventional MSD .....	68
5.2 Relations between metrics of estimates in MSD.....	71
5.3 IMS-MLD for two-level concatenated codes.....	79
5.3.1 Theorems for the two-stage iterative decoding algorithm .....	79
5.3.2 The Algorithm for two-stage decoding.....	81
5.4 IMS-MLD for $m$ -level concatenated codes.....	83
5.4.1 Theorems for $m$ -stage iterative decoding algorithm .....	83
5.4.2 The Algorithm for $m$ -stage decoding .....	84
5.5 Suboptimum versions .....	87
5.5.1 Limited list size decoding.....	87
5.5.2 Threshold decoding.....	87
<b>Chapter 6 Applications of IMS-MLD Algorithm.....</b>	<b>90</b>
6.1 Application of IMS-MLD to Decomposable RM codes.....	90
6.2 Application of IMS-MLD to BCM codes .....	96
6.2.1 Application of the maximum likelihood version.....	97

6.2.2 Application of suboptimum threshold version .....	100
<b>Chapter 7 Conclusion.....</b>	<b>103</b>
7.1 Work presented.....	103
7.2 Future work.....	106
<b>References.....</b>	<b>108</b>

## List of Tables

Table 3.1 Two-level Decompositions of Some Reed-Muller Codes .....	32
Table 3.2 Three-level Decompositions of Some Reed-Muller Codes .....	32
Table 6.1 Decoding Complexity for IMS-MLD of RM (3,7) = (128,64,16).....	93
Table 6.2 Decoding Complexity for Suboptimum IMS of RM (3,7)=(128,64,16).....	93
Table 6.3 Computational Complexity of various algorithms for (128,64,16) code.....	94
Table 6.4 Computational Complexity of various algorithms for (128,99,8) code.....	95

# List of Figures

Figure 2.1 Concatenated Coding System .....	11
Figure 2.2 NASA standard concatenated system.....	12
Figure 2.3 Performance of NASA standard concatenated code .....	13
Figure 2.4 Encoding of a concatenated code.....	15
Figure 3.1 Example of a two-level concatenated code.....	21
Figure 3.2 Outer code encoding example for a two-level concatenated code .....	22
Figure 3.3 $m$ -level concatenated code .....	26
Figure 4.1 Multilevel concatenated scheme .....	36
Figure 4.2 Schematic representation of MSD.....	39
Figure 4.3 Stage- $i$ inner code decoding.....	43
Figure 4.4 Stage- $i$ outer code decoding.....	44
Figure 4.5 Performance of (32,6,16) RM code with 2-stage decoding .....	49
Figure 4.6 Performance of two different decompositions of (32,6,16) RM code.....	51
Figure 4.7 Performance of multistage decoding of (64,42,8) RM code.....	53
Figure 4.8 Performance of 3-stage decoding of (128,64,16) RM code .....	55
Figure 4.9 Performance of 3-stage decoding of (128,99,8) RM code .....	56
Figure 4.10 Performance of List MSD of (64, 42, 8) code.....	63
Figure 4.11 Performance of List MSD of (128,64,16) code.....	64
Figure 4.12 Performance of List MSD of (128,99,8) code.....	65
Figure 6.1 Performance of (128,64,16) over AWGN channel.....	94
Figure 6.2 Performance of (128,99,8) over AWGN channel .....	96

<b>Figure 6.3 Performance of the (32,6)(32,26)(32,31) BCM code.....</b>	<b>98</b>
<b>Figure 6.4 Performance of the (32,6)(32,6)(32,31) BCM code .....</b>	<b>99</b>
<b>Figure 6.5 Performance of BCM code with terminated convolutional outer codes.....</b>	<b>101</b>

# **Chapter 1**

## **Introduction**

The role of channel coding in digital communication system is to provide reliability, that is, a successful recovery of the transmitted information at the receiver with as small an error rate as required despite the presence of noise and interference.

Two theoretical questions arise in relation to this problem: under what conditions is reliable communication possible, and what are the characteristics of the code that can provide the required level of reliability. C. E. Shannon proved [1] that if the system transmission rate is smaller than the channel capacity, there exist a code that can provide transmission of data with arbitrarily small error rate. On the other hand, “the search for the perfect code” inspired the research in the areas of Information Theory and Communications for the past 55 years. During this time, many good codes and classes of codes were constructed.

In general, longer codes achieve better performance, but the required time, memory, and amount of computation needed for successful decoding of these codes may be infeasible for practical implementation. Thus the search for an efficient decoding algorithm is as important as the search for a good code. A good trade-off between the performance, measured by the low error probability, and efficient decoding, measured by the low complexity, is set as a criterion.

Soft-decision decoding algorithms have been investigated by many coding theorists over the past several decades. However, finding a computationally efficient soft-decision maximum likelihood (ML) decoding algorithm that can be implemented in practice is still an open and challenging problem, especially for long codes.

In [2], an elegant decoding method was presented, based on the trellis structure of the convolutional codes. This algorithm is later applied to block codes as well, as trellises for these are obtained based on their generator matrices [3]. The advantage of trellis-type decoders is modular implementation implied by the regular code structure. At this time, it is standard and inexpensive decoding method. However, trellises for long codes of rate close to  $R = 1/2$  become large (number of states grows with the length) and trellis-based soft-decision algorithms performed on the full trellis face the problem of unfeasible decoding complexity. Contributing to the complexity is the large number of states in the trellis, and the fact that the states are often densely connected. An example of a long code is the (128, 64, 16) Reed Muller (*RM*) code. *RM* codes can be easily decoded using hard-decision decoding, for instance, majority-logic decoding. To implement soft-decision Viterbi decoding, we need to process a trellis consisting of more than  $2^{15}$  states for the optimal, but possibly non-uniform sectionalization. Thus, for the codes for which simple algebraic decoders exist, the improved performance over hard decision algorithms is obtained at a very high price.

Multistage decoding (MSD) [4-6] was devised for decoding decomposable codes [7] and multilevel codes [8] to achieve an efficient trade-off between error performance and decoding complexity. In a conventional soft-decision MSD scheme, each stage uses

soft-decision decoding, but hard-decision decoded output is used to modify the soft-decision received sequence for the next decoding stage. Decoded information is passed down from one stage to the next until the last decoding stage. With this decoding, incorrect decoded information at one stage may result in error propagation through the subsequent decoding stages. Therefore, the decoding is not ML even if every decoding stage is ML – it is a suboptimum decoding scheme. This decoding works well for short to moderately long codes, say lengths up to 64. It provides excellent trade-off between error performance and decoding complexity. The performance degradation compared with that of MLD is very small, no more than .4 dB for decomposable codes of lengths 64 [9].

Performance analysis of MSD was addressed in [6, 9-11]. The performance degradation compared to one-level optimum (MLD) decoding is a result of increase of effective error coefficients. In [9], a bound on the effective error coefficients of two-stage decoding was derived. It was shown that it is directly proportional to the redundancy of the second level code, and thus the second level code should be kept as simple as possible. It was also argued that in order to prevent large increase in nearest neighbors, that is – large effective error coefficient, the number of decoding stages in multi-stage decoding should be as small as possible. Simulation results for some decomposed Reed-Muller codes showed that with proper choice of decomposition and using only two decoding stage, performance of MSD was within a fraction of dB from the optimum ML performance.

However, for long Reed-Muller codes, say of length 128 and up, the performance degradation becomes severe. Keeping the number of stages small, as suggested in [9], is

not possible for long codes of large lengths. More decoding stages are needed in order to keep the complexity of each stage amenable. Examples of three-stage decoding of Reed-Muller codes of length 128 were given in [12]. The performance degradation of the (128,64) Reed-Muller code with three-stage decoding was shown to be 1.5 dB at bit-error rate of  $10^{-5}$ . In [4], a multilevel method for combining binary block component codes with a signal set to form a block modulation code was introduced. This method suggested systematic construction of block modulation codes with large minimum squared Euclidean distance. These codes achieved good performance and were able to be decoded using multistage decoding algorithm.

Huber et al. also investigated multistage decoding for block coded modulation codes. In order to reduce the number of nearest neighbors, which is a result of multiple representations of binary symbols [10], they suggest that the multilevel code should be designed not by using the balanced distance (product of the outer code minimum distance and signal constellation squared Euclidean distance), but by using the information theoretic rule to determine the outer code rates. Codes in their examples designed based on the new criterion perform close to optimum, using multistage decoding.

The rule given in [9] that the redundancy of the second level code in the multilevel scheme should be small could also be interpreted as requiring the first level code to be as powerful as possible. Keeping the number of decoding levels small also agree with the requirement that the first level code is strong. In addition, it was shown in [12] that two-level and three-level decompositions of the same block code achieve quite similar performance if the first level codes are identical. In [13], similar observation was

made and a decoding scheme in which for all but the first-level code the hard decision decoding was suggested. Good performance of this scheme is due in part to usage of turbo decoding for the outer codes. However, the importance of the decoding of the first level is nevertheless emphasized.

To improve the performance of decoding methods that consist of several stages, list decoding is often used. List decoder simply provides more than one estimate at the time and puts them on the list. This list is delivered to the next decoder or to the destination. Investigation on list algorithms for concatenated codes showed [14] that with large list-sizes, error probability of the list decoding is very small.

List decoding algorithm can be used to improve the performance of MSD for long multilevel codes. Instead of only one, a list of  $L$  best decoded estimates is passed from one decoding stage to the next. At the last stage, the best codeword (largest correlation, or smallest Euclidean distance) is chosen as the decoded codeword. Examples in [12] showed that in some cases even using a list of small size achieves optimum, or close to optimum performance. However, for longer codes and larger number of decoding stages, the list size needs to be increased significantly. Of course, the increase of the decoding complexity of this list multistage decoding is multifold.

It was shown in [17] that list decoding with variable list size achieves large error exponents, that is, small error probabilities, while keeping the average complexity reasonably small. The average list size can be kept small by using a type of threshold decoding for putting the codewords on the list. The problem that exists with the variable

size list decoding is that the worst-case list size could be quite large, implying that significant buffering may be needed. However, once such buffer is provided, higher list-error exponents than the regular error exponent for the same code rate may be obtained, resulting in a much smaller error probability.

The algorithm proposed in [16] was an application of the Forney's [17] threshold list decoding to multilevel block modulation codes. The modification with respect to Conventional MSD (CMSD) consisted of passing an additional  $L$ -1 candidate from the first to the second stage, if the distance measure between the overall estimate based on the first candidate and the received sequence is larger than a threshold  $T$ . If the distance measure is Euclidean distance, the threshold should be larger than  $d_{\min}^2 / 4$  (where  $d_{\min}$  denotes the minimum Euclidean distance), since for  $d \leq d_{\min} / 2$  the estimate is surely optimum, and having  $T \leq d_{\min}^2 / 4$  would result in unnecessary operations when the estimate is optimum but not in the sphere of radius  $\sqrt{T}$ . It was reported that half of the gap between MLD and CMSD for a modulation code of length 64 was gained by using this algorithm with  $L = 2$  and a certain value of threshold. Both complexity and performance are functions of the threshold value.

For long decomposable codes, in order to maintain the trellis complexity of each component code in a practically implementable range, the number of decoding stages in MSD must go up. Investigation of list passing in MSD of RM codes [12] led to the conclusion that for longer codes and  $m$ -stage decoding ( $m > 2$ ) passing two estimates from first to the second stage gains only a fraction of a difference between MLD and

CMSD. Even increasing the size of the list to 5, gained only half (0.75 dB) of that difference for three stage decoding of Reed-Muller (128, 64, 16) code. As a result, the size of the list must increase to maintain small performance degradation compared to MLD. This results in a large increase of computational complexity. To overcome this problem, a list of variable size depending on the signal-to-noise ratio (SNR) can be used to reduce the average computational complexity.

The objective of this dissertation was to investigate MSD based decoding algorithms that can achieve better performance for long codes than the conventional MSD algorithm. The investigation lead to the iterative multistage maximum likelihood decoding (IMS-MLD) algorithm which achieves ML performance through decoding iterations. Each decoding iteration begins with the generation of a new estimate at a certain decoding stage. This new estimate is then passed down to the subsequent stages of decoding. During a decoding iteration, two simple optimality conditions are tested. If one of the conditions is satisfied, the decoding process is terminated and the best codeword found at the time is the most likely (ML) codeword. Decoding iteration continues until the ML codeword is found. In this IMS-MLD algorithm, a new estimate is generated only when it is needed, which reduces the decoding complexity enormously compared to other known MLD algorithms.

The organization of the dissertation is as follows. In chapter 2 we introduce concatenation as a method of forming powerful long codes. In chapter 3, the basics of multilevel concatenation are presented. Two methods for obtaining multilevel codes are discussed: decomposition of Reed-Muller [18] codes and construction of Block coded

modulation (BCM) [5, 19-31] codes. In chapter 4, we present details of Multistage Decoding (MSD) algorithm. The novel IMS-MLD algorithm is presented in chapter 5, and the application of the algorithm to decoding of decomposed RM codes and BCM codes, as well as some suboptimum versions of the algorithm in chapter 6. Chapter 7 contains concluding remarks and the guidelines for the future work.

## Chapter 2

# Concatenated Codes

It is well known [1] that small probability of error can be achieved using long codes. Longer codes of the same type are more powerful than the codes of smaller lengths, for both hard and soft decision decoding, thanks to the larger minimum distances between codewords. However, the gain in performance of long codes is paid, sometimes too dearly, by the increase in decoding complexity. The complexity grows mostly exponentially with the code length, and becomes a significant problem for implementation.

Concatenation is a good method of constructing long codes that can be decoded using algorithms with low decoding complexity. By concatenating two codes, such that the output of the first encoder is connected to the input of the second encoder, a long code (length of the resulting code is equal to the product of lengths of component codes) with large minimum distance is obtained. The advantage of a concatenated code over a code of similar length and minimum distance, however, one that does not have concatenated structure, is its suitability for decoding in two steps, each step having significantly smaller complexity than the non-concatenated structure.

The common characteristic of concatenated codes is that the minimum distance is lower bounded by the product of the minimum distances of the component codes, while the overall decoding complexity is proportional to the sum of their decoding

complexities. This offers a good trade-off between the power/performance and decoding complexity of the code. Furthermore, concatenation can be generalized to multilevel coding.

Concatenated codes differ with respect to the types of component codes and the number of levels of concatenation. In this chapter, we describe the structure, encoding and decoding algorithms for single-level concatenated codes. Multilevel concatenation is presented in chapter 3.

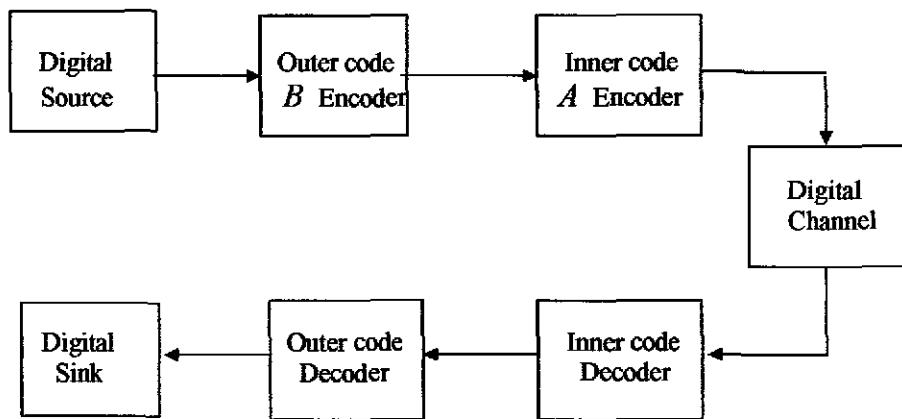
## 2.1 Structure of concatenated codes

Concatenated coding scheme, as originally proposed by Forney [15] is shown in Figure 2.1. The information sequence is transformed using two codes for two consecutive encodings. First, the information sequence is encoded by the outer code  $B$ , which is usually a non-binary code over  $GF(q)$ . The codeword produced by the outer code is then encoded, symbol-by-symbol, using the binary inner code  $A$ .

The encoding block thus consists of two encoders connected back to back, outer code encoder followed by the inner code encoder, and decoding block – of two decoders, inner code decoder followed by the outer code decoder. The two encoders are sometimes regarded as a *super encoder* and the two decoders - as a *super decoder*, each corresponding to the overall concatenated code  $C$ . Another way of looking at the encoder of the concatenated code is to group inner encoder, channel and inner decoder into *super channel*. Inner coder/decoder pair reduces the transmission probability

introduced by the channel and pass the partially corrected codeword to the outer code decoder.

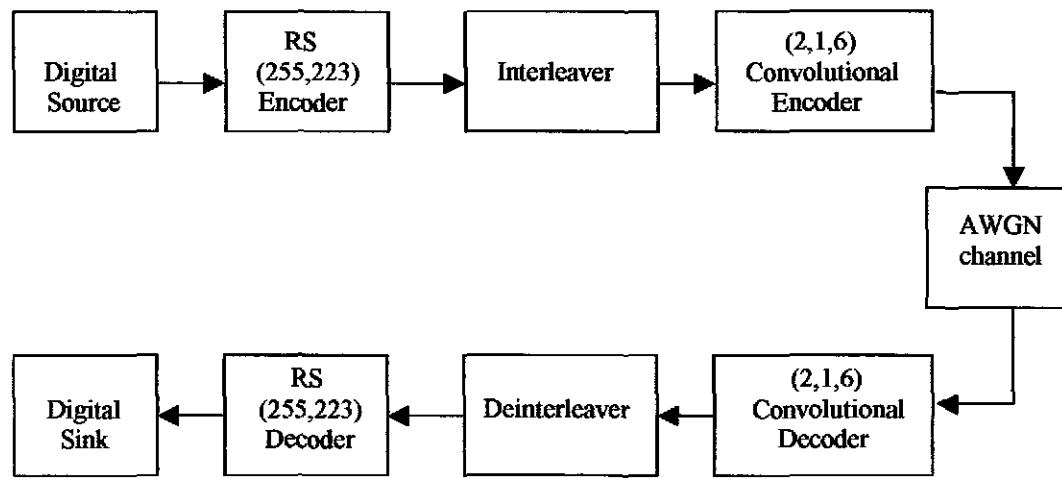
Let the dimension (number of information bits) of the outer and inner code be denoted by  $K$  and  $k$ , respectively, and the lengths by  $N$  and  $n$  respectively. Binary representations of symbols of the outer code codewords are considered information sequences for the binary inner code, so the number of symbols,  $q$ , in the alphabet of outer code  $B$  must be equal to the number of information sequences for the code  $A$ , that is,  $q = 2^k$ .



**Figure 2.1 Concatenated Coding System**

An example of a good concatenated system, used by NASA as one of the standard schemes for space communication, is given in Figure 2.2. The scheme combines Reed-

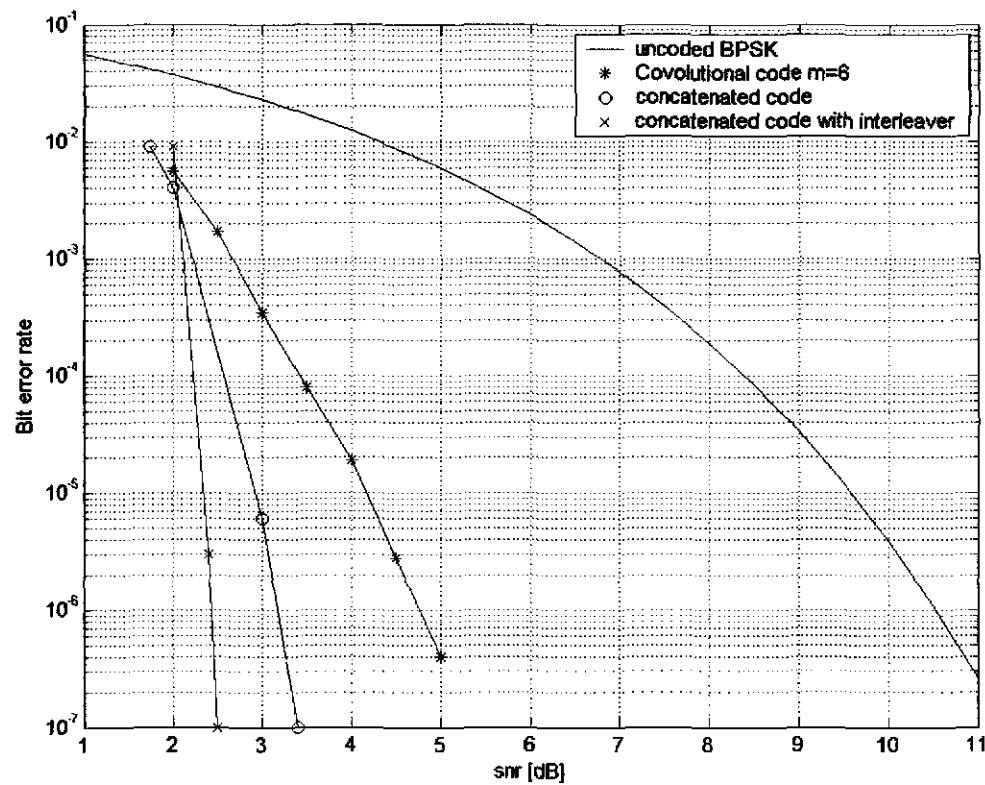
Solomon (255,223) code with the rate  $\frac{1}{2}$  convolutional code of memory order  $M=6$  and minimum distance  $d_{\min} = 10$ . Interleaver is used to enhance the performance. This system achieves a powerful performance with probability of error as low as  $P_e = 10^{-7}$  at signal-to-noise ratio  $\text{SNR}=2.5\text{dB}$  over AWGN channel, as shown in Figure 2.3.



**Figure 2.2 NASA standard concatenated system**

### 2.1.1. Parameters and properties

- 1) The length of the concatenated code  $C$  is equal to the product of the lengths of the outer and the inner code  $n_C = N \cdot n$ .
- 2) The dimension of the concatenated code  $C$  is equal to the product of the dimensions of the outer and inner code  $k_C = K \cdot k$ .
- 3) Code  $C$  is linear if its component codes, outer code  $B$  and inner code  $A$ , are linear.



**Figure 2.3 Performance of NASA standard concatenated code**

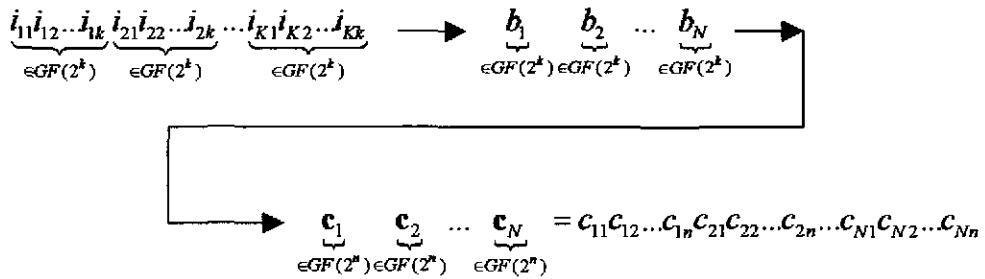
Important property of a linear code is that the minimum distance between the codewords is equal to the minimum non-zero weight of a codeword in  $C$ . Using this property, it is easily shown that:

- 1) The minimum distance of a concatenated code  $C$  is bounded below by the product of the minimum distance of outer code,  $d_{\min_B}$ , and the minimum distance of inner code,  $d_{\min_A}$ .

*Proof:* Assume the outer code  $B = (N, K)$  and the inner code  $A = (n, k)$  are both linear codes. Then, the minimum weight codewords in  $A$  and  $B$  have weights  $d_{\min_A}$  and  $d_{\min_B}$ , respectively. A non-zero codeword in  $C$  will have at least  $d_{\min_B}$  non-zero blocks (minimum weight in  $B$ ), each of which will have weight at least  $d_{\min_A}$  (blocks are codewords in  $A$ ). Thus the minimum weight of a codeword in  $C$ ,  $w_{\min}$ , is at least  $d_{\min_A} \cdot d_{\min_B}$ . The equality holds if, for at least one minimum weight codeword in  $C$ , all the non-binary symbols in that codeword are mapped by the inner encoder into minimum weight codewords in  $A$ . By linearity property of the concatenated code, the minimum distance  $d_{\min_C}$  is equal to the weight of the minimum weight codeword in  $C$ , therefore  $d_{\min_C} = w_{\min} \geq d_{\min_A} \cdot d_{\min_B}$ .

## 2.2 Encoding of concatenated codes

The information sequence of the concatenated code  $C$  is  $k \cdot K$  bits long, where  $k$  is the dimension of the inner code, and  $K$  the dimension of the outer code. During outer code encoding, consecutive groups of  $k$  bits are regarded as symbols in  $GF(2^k)$ . There are  $K$  such groups. Thus, the information sequence consists of  $K$  symbols in  $GF(2^k)$ . These  $K$  information symbols are encoded using outer code  $B$  into a codeword of length  $N$ , consisting of symbols over  $GF(2^k)$ . Next, each of the  $N$  symbols is regarded as a binary sequence of length  $k$ , and encoded using the inner code  $A$  into a binary codeword of length  $n$ . Overall, we obtain, as shown in Figure 2.4, a sequence of  $n \cdot N$  bits, which represents a codeword in concatenated code  $C$ .



**Figure 2.4 Encoding of a concatenated code**

### 2.3 Decoding of concatenated codes

The structure of a concatenated coding scheme makes it possible to perform decoding of a long code  $C$  in two steps, using decoding of codes of smaller lengths,  $n$  and  $N$  respectively, for inner and outer code. Each of these decodings is much simpler than the decoding of a code of length  $n \cdot N$  that does not have concatenated structure. Thus, the concatenated structure provides a significant reduction of decoding complexity.

Decoding of the received sequence is performed in two steps. The symbols of the received sequence are grouped into  $N$  blocks of length  $n$ . Each block is then decoded using inner code decoder to obtain  $k$  decoded bits per block. Sequences of  $k$  decoded bits are considered a symbol over  $GF(2^k)$  and the outer code decoding is performed on the words of length  $N$ .

Large minimum distance of the concatenated scheme will ensure a good asymptotic error performance. Especially good results are obtained for transmission over

compound channels, where first the scattered random errors are corrected by the inner code, and then the bursts are taken care of by the non-binary outer code.

## 2.4 Types of concatenated schemes based on component codes

The outer codes in Forney's scheme are Reed-Solomon codes. Hard decision decoding of these codes<sup>1</sup> is relatively simple to implement and achieve good error performance, as long as the inner code provides estimates with a reasonably small probability of error ( $<10^{-2}$ ). For this reason, Reed-Solomon codes are used in many concatenated coding schemes.

While outer codes are usually non-binary codes, the type of inner codes can vary from orthogonal codes implemented as modulation schemes to block and convolutional codes. In the following, we give a brief description for each of the three types.

### 2.4.1 Orthogonal codes as inner codes

This type of scheme uses simplex, orthogonal, or biorthogonal codes implemented as binary block codes or M-ary modulation scheme (such as  $2^k$ -ary FSK) as inner codes. Application is restricted to cases where constraints on bandwidth are not imposed, since the rates are usually very low, and therefore the bandwidth expansion is high. For more detailed description of these types of codes see [32].

---

<sup>1</sup> Some soft decision decoding methods give even better results with a comparable increase in decoding complexity.

### 2.4.2 Block codes as inner codes

In this type of concatenated scheme, the inner code is a (linear) block code  $(n, k)$  of length  $n$  and dimension  $k$ . Each coded symbol of the outer code codeword is regarded as a  $k$  bit long binary information sequence, which is then encoded using  $(n, k)$  binary (linear) inner code.

The correspondence between the dimension of the inner code,  $k$ , and the size of Galois field over which the outer code is constructed,  $q = 2^k$ , restricts the application of concatenation to block codes of small dimensions, as otherwise the “symbol size” (size of the finite field) would become too large. A slight modification of the original encoding procedure allows for much larger dimensions of the inner block code. Namely, if several, say  $l$ , outer code symbols are regarded as an information sequence for the inner code, then the size of the finite field  $q = 2^{k/l}$  can be kept small even for large dimensions of the inner code. Thus, more powerful inner codes can be used. Naturally,  $l$  – the number of consecutive symbols to be considered together – needs to be a divisor of  $k$ .

A disadvantage of the modified scheme is that an incorrect decoding of the inner code affects  $l$  consecutive outer code symbols, making these symbol errors correlated. To avoid this problem, an interleaver can be used between the outer and inner encoder with a corresponding deinterleaver between the inner and outer decoder.

### **2.4.3 Convolutional codes as inner codes**

In this type of concatenated scheme, the inner code is a convolutional code. The input to the inner code is a sequence of bits obtained by considering the binary representation of symbols of the outer code. Binary sequences at the output of the convolutional code is transmitted over the channel and at the receiver, the sequence is first decoded using convolutional code decoder, and then the estimated information bits are grouped into symbols and decoded using outer code decoder.

The errors from the Viterbi decoder of a convolutional code appear in bursts [2, 33]. Since Reed-Solomon codes are good burst error correcting codes [34], they represent a good choice for outer code when inner convolutional codes are used. However, very long error bursts at the output of the convolutional code decoder might cause correlated errors in successive symbols of RS code. To avoid correlated errors, an interleaver/deinterleaver pair is used in the scheme (as shown in Figure 2.2). Interleaving is performed in such a way that no two symbols within the same window size of Viterbi decoder belong to the same codeword of the RS outer code.

The combination of RS outer code with convolutional inner code represents a powerful scheme, which achieves very low probability of error for reasonably small signal-to-noise ratios.

In this chapter we discussed a powerful code designing technique – concatenation. In the next chapter, we describe how this concept can be generalized to multilevel coding schemes, in which each level is a concatenated code. The codes with such structure can be decoded easily using reduced complexity decoding method of multistage decoding [5].

# Chapter 3

## Multilevel Concatenation

The idea of concatenation presented in chapter 2 can be generalized to an  $m$ -level scheme ( $m \geq 2$ ). The first attempt to do so was done by Blokh and Zyablov in 1976. In [8], they presented a three level scheme in which each level consisted of the inner and the outer code. All the constituent codes were linear. The following paper by Zinoviev [35] further generalized Blokh-Zyablov codes to a non-linear scheme. Zyablov et al. call this scheme Generalized Concatenated codes since they are obtained as a generalization of Forney's concatenated codes. Throughout this Dissertation, we use the term *multilevel codes* to include multilevel concatenated codes and the multilevel coded modulation codes.

We first give some examples, and then describe general construction for two types of multilevel concatenated codes. Later in the chapter, we discuss the encoding and parameters of multilevel concatenated codes, while a well-known algorithm for decoding of multilevel codes, *multistage decoding (MSD) algorithm*, is described in chapter 4.

### 3.1 Structure of multilevel concatenated codes

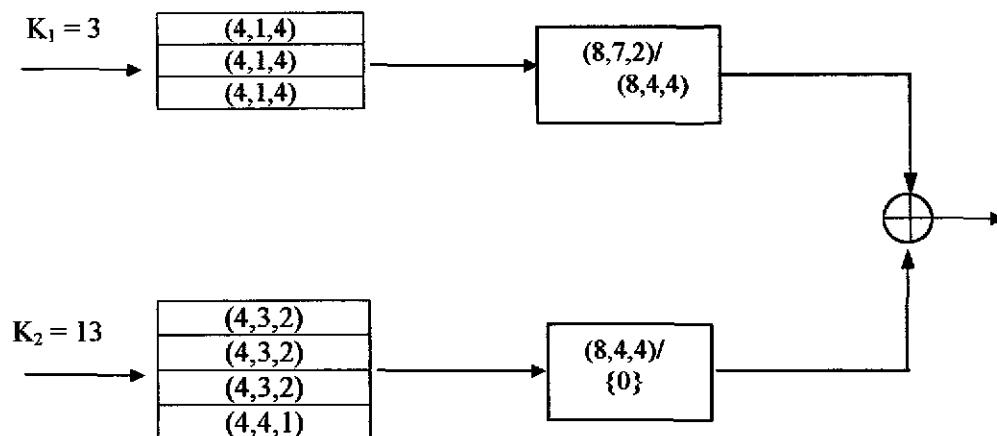
Generalized concatenated code  $C$  consists of  $m$  levels of concatenation with one outer and one inner component code in each level. The outputs of these  $m$  concatenated codes are then combined to obtain a codeword in  $C$ . Different levels of concatenation are not,

however, independently chosen. The set of inner codes in levels 1 through  $m$  are based on the partition chain of a block code, or a signal constellation. We therefore distinguish two types of multilevel concatenation: 1) Multilevel concatenated codes (MCC), and 2) Multilevel coded modulation codes.

### Example 3.1

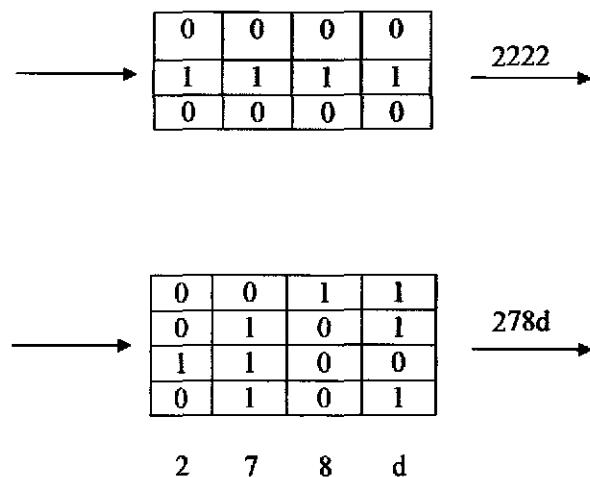
Let us consider an example of a multilevel concatenated code with the encoder given in Figure 3.1. It is a 2-level code consisting of concatenation of one inner and one outer code at each level.

The first outer code is a length  $N = 4$ , dimension  $K_1 = 3$ , and minimum distance  $d_{\min B_1} = 4$  code, obtained by combining three repetition codes of length four to get a non-binary code over  $GF(2^3)$ .



**Figure 3.1 Example of a two-level concatenated code**

The second outer code is also a non-binary code, with parameters  $N = 4$ ,  $K_2 = 13$ , and  $d_{\min B_2} = 1$ . To obtain this non-binary code with symbols in  $GF(2^4)$ , three  $(4,3,2)$  parity check codes are combined with a universal code  $(4,4,1)$  of length 4. The encoding using 0100010101100101 as input is shown in Figure 3.2. Symbols are represented in hexadecimal form ( $d_{16} = 13_{10}$ ), and information bits are given in white fields.



**Figure 3.2 Outer code encoding example for a two-level concatenated code**

The first level inner code is formed using partitions of a linear parity check code of length 8,  $(8,7,2)$ , by the Reed-Muller  $(8,4,4)$  code, which is a subcode of  $(8,7,2)$ . There are  $2^7 = 128$  codewords in  $(8,7,2)$  code, and  $2^4 = 16$  codewords in  $(8,4,4)$  code. Therefore, the number of cosets in this partition is equal to  $2^7 / 2^4 = 8$ . Each codeword in the coset is a sum of the coset representative and a codeword from the subcode  $(8,4,4)$ .

The first inner code,  $A^{(1)} / A^{(2)}$ , maps the symbols of the outer code alphabet into cosets of the partition  $(8, 7, 2)/(8, 4, 4)$ . Each symbol corresponds to a particular coset in the partition. Since each coset can be specified by a coset representative, the symbols of the outer code  $B^{(1)}$  are mapped to the coset representatives of the partition. Therefore, a codeword of length  $N = 4$  is mapped onto a sequence of four coset representatives of  $(8, 7, 2)/(8, 4, 4)$ , giving as the output of the first concatenation level a  $N \times n = 4 \times 8 = 32$  bit long sequence  $\mathbf{c}^{(1)}$  (superscript denotes the level of concatenation).

Similarly, we obtain a 32-bit long codeword  $\mathbf{c}^{(2)}$  at the output of the second concatenation level. The overall codeword is then obtained as a sum of codewords in level-1 and level-2, i.e.,  $\mathbf{c} = \mathbf{c}^{(1)} + \mathbf{c}^{(2)}$ . Note that the first level codeword  $\mathbf{c}^{(1)}$  is a sequence of 4 coset representatives in  $(8, 7, 2)/(8, 4, 4)$ ,  $\mathbf{c}^{(1)} = \mathbf{a}_1^{(1)} \mathbf{a}_2^{(1)} \mathbf{a}_3^{(1)} \mathbf{a}_4^{(1)}$ ,  $\mathbf{a}_j^{(1)} \in A^{(1)}$ , for  $j = 1, 2, 3, 4$ , while  $\mathbf{c}^{(2)}$  is a sequence of 4 codewords in  $(8, 4, 4)$ ,  $\mathbf{c}^{(2)} = \mathbf{a}_1^{(2)} \mathbf{a}_2^{(2)} \mathbf{a}_3^{(2)} \mathbf{a}_4^{(2)}$ ,  $\mathbf{a}_j^{(2)} \in A^{(2)}$ , for  $j = 1, 2, 3, 4$ . A sum of a coset representative from  $\{A^{(1)} / A^{(2)}\}$  and a codeword in  $A^{(2)}$ , is a codeword in  $A^{(1)}$ . Therefore, the sum of  $\mathbf{c}^{(1)}$  and  $\mathbf{c}^{(2)}$  is a sequence of 4 codewords in  $(8, 7, 2)$ .

Described two-level concatenated code has  $k_C = K_1 + K_2 = 3 + 13 = 16$  information bits, and is of length  $N \times n = 4 \times 8 = 32$ . Thus, it is a  $(32, 16)$  code. It can be shown that with proper mapping of  $GF(2^4)$  symbols onto coset representatives, the minimum distance of this code is equal to  $d_{\min B_1} \times d_{\min P_1} = d_{\min B_2} \times d_{\min P_2} = 8$ . In fact, the code in this example is the second order Reed-Muller code  $RM(2, 5)$ .

The 2-level concatenated scheme in Example 3.1 employs non-binary outer codes  $B^{(1)}$  and  $B^{(2)}$  constructed by interleaving several binary codes. Non-binary symbol at position- $i$  is a function of the bits at position- $i$  in each code. The outer code can also be a non-binary code constructed directly over  $GF(8)$ .

### **Example 3.2**

Consider the concatenated code  $C$  from Example 3.1. Let us replace the inner code  $(8, 7, 2)/(8, 4, 4)$  with a partition of a signal constellation by its subconstellation. Since the number of codewords in  $A^{(1)} = (8, 7, 2)$  and  $A^{(2)} = (8, 4, 4)$  is 128 and 16, respectively, the code  $A^{(1)}$  is replaced by a signal constellation consisting of 128 points, and the code  $A^{(2)}$  by a sub-constellation consisting of 16 points.

Based on the output of the first outer code from Example 3.1,  $\mathbf{b}^{(1)} = (2, 2, 2, 2)$ , subconstellation-2 would be chosen for each of the four symbols of the outer code codeword  $\mathbf{b}^{(1)}$ , so the output of the first concatenation level would be  $\mathbf{c}^{(1)} = 2222$ .

Next, based on the output of the second outer code  $B^{(2)}$ ,  $\mathbf{b}^{(2)} = (2, 7, 8, d)$ , a signal point would be chosen for each symbol in  $\mathbf{b}^{(2)}$  from the appropriate sub-constellation determined by  $\mathbf{c}^{(1)}$ : 2<sup>nd</sup> signal point in sub-constellation 2, followed by the 7<sup>th</sup>, 8<sup>th</sup>, and 13<sup>th</sup> ( $13_{16} = d_{16}$ ) point, all in subconstellation-2.

The code given in Example 3.1 belongs to a group of multilevel concatenated codes whose inner codes are formed from partitions of a linear block code. The code

given in Example 3.2 to a group of multilevel concatenated modulation codes whose inner codes are formed from partitions of a signal constellation. We now describe these two general classes of multilevel codes.

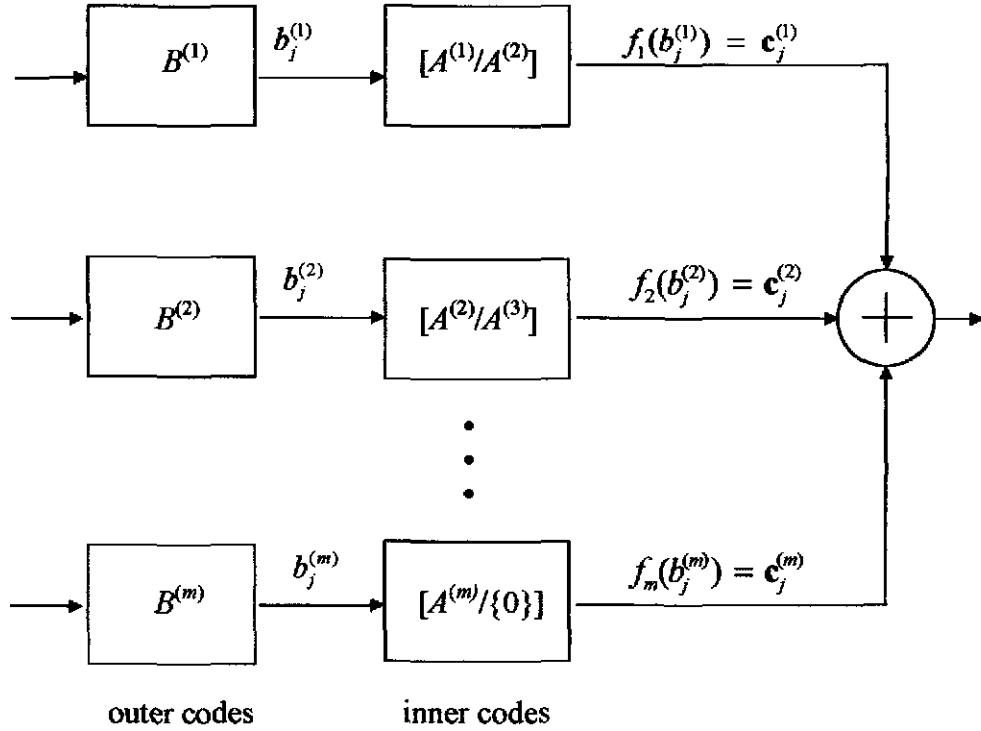
### 3.2 Multilevel Concatenated Codes

An  $m$ -level concatenated code [9] is formed from a set of  $m$  inner codes and a set of  $m$  outer codes as shown on Figure 3.3. The  $m$  inner codes are coset codes formed from a binary linear block code  $A^{(1)}$  of length  $n$  and a sequence of  $m$  linear subcodes of  $A^{(1)}$ , denoted by  $A^{(2)}, A^{(3)}, \dots, A^{(m+1)} = \{\mathbf{0}\}$ , with  $A^{(l+1)} \subset A^{(l)}$  for  $1 \leq l \leq m$ . The inner code  $[A^{(l)} / A^{(l+1)}]$  at level- $l$  is simply a specific set of **representatives** of the cosets of  $A^{(l)}$  modulo  $A^{(l+1)}$ , denoted by  $A^{(l)} / A^{(l+1)}$ . Each codeword in  $A^{(l)}$  is a sum of  $m$  coset representatives of  $[A^{(l)} / A^{(2)}], [A^{(l)} / A^{(3)}], \dots, [A^{(l)} / \{0\}]$ , respectively. For  $1 \leq l \leq m$ , let  $q_l \triangleq |[A^{(l)} / A^{(l+1)}]|$ , and let  $[A^{(l)} / A^{(l+1)}] = \{\mathbf{a}_j^{(l)} : 1 \leq j \leq q_l\}$ . Then a coset in  $A^{(l)} / A^{(l+1)}$  is given by

$$\{\mathbf{a}_j^{(l)} + \mathbf{a} : \mathbf{a} \in A^{(l+1)}\}. \quad (3.1)$$

For  $1 \leq l \leq m$ , the outer code at level- $l$ , denoted by  $B^{(l)}$ , is an  $(N, K_l)$  linear block code over  $GF(q_l)$ . Let  $f_l(\cdot)$  be a one-to-one mapping from  $GF(q_l)$  onto  $[A^{(l)} / A^{(l+1)}]$ . Let  $\mathbf{b}^{(l)} = (b_1^{(l)}, b_2^{(l)}, \dots, b_N^{(l)})$  be a codeword in the level- $l$  outer code  $B^{(l)}$ . During the encoding of level- $l$ ,  $\mathbf{b}^{(l)}$  is encoded into the following sequence,

$$\mathbf{c}^{(l)} = (\mathbf{c}_1^{(l)}, \mathbf{c}_2^{(l)}, \dots, \mathbf{c}_N^{(l)}) = (f_l(b_1^{(l)}), f_l(b_2^{(l)}), \dots, f_l(b_N^{(l)})) \quad (3.2)$$



**Figure 3.3  $m$ -level concatenated code**

where  $\mathbf{c}_j^{(l)} = f_l(b_j^{(l)})$  is a coset representative in  $[A^{(l)}/A^{(l+1)}]$ . Therefore,  $\mathbf{c}^{(l)}$  is a sequence of coset representatives from  $[A^{(l)}/A^{(l+1)}]$  and is a codeword in the  $l^{\text{th}}$  level concatenated code, denoted by  $C^{(l)} \triangleq B^{(l)} \circ [A^{(l)}/A^{(l+1)}]$ . The direct sum

$$C \triangleq B^{(1)} \circ [A^{(1)}/A^{(2)}] \oplus B^{(2)} \circ [A^{(2)}/A^{(3)}] \oplus \dots \oplus B^{(m)} \circ [A^{(m)}/A^{(m+1)}] \quad (3.3)$$

forms an  $m$ -level concatenated code. For simplicity, we denote this  $m$ -level code with  $C \triangleq \{B^{(1)}, B^{(2)}, \dots, B^{(m)}\} \circ \{A^{(1)}, A^{(2)}, \dots, A^{(m)}\}$ . Every codeword  $\mathbf{c} \in C$  is a sum of  $m$  codewords  $\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \dots, \mathbf{c}^{(m)}$  from the  $m$  component concatenated codes  $C^{(1)}, C^{(2)}, \dots, C^{(m)}$  respectively, i.e.,  $\mathbf{c} = \mathbf{c}^{(1)} + \mathbf{c}^{(2)} + \dots + \mathbf{c}^{(m)}$ .

### 3.3 Encoding Algorithm and parameters of MCC

The information block is first divided into  $m$  parts of size  $K_l$ ,  $1 \leq l \leq m$ , and outer code encoding at each level is performed using the outer code  $B^{(l)}$ . Next, an array of size  $m \times N$  is formed in which the  $l^{\text{th}}$  row represents the codeword  $\mathbf{b}^{(l)}$ .

Then each column of this array is mapped into a codeword in  $A^{(l)}$  using the mappings  $f_l(\cdot)$ . As a result, we obtain a sequence of  $N$  codewords in  $A^{(l)}$ , denoted  $(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N)$ , where

$$\mathbf{c}_j = \sum_{l=1}^m f_l(b_j^{(l)}).$$

#### Parameters of MCC

It was shown in [35] that the minimum Hamming distance of a generalized concatenated code  $C$  is lower bounded by the smallest minimum distance among the component codes, i.e.,

$$d_{\min} \geq \min_{1 \leq l \leq m} \{d_{\min A^{(l)}} d_{\min B^{(l)}}\}.$$

The multilevel concatenated code  $C$  is, therefore, a binary code of length  $n_C = N \cdot n$ , dimension  $k_C = \sum_{l=1}^m K_l$ , and minimum distance  $d_{\min} \geq \min_{1 \leq l \leq m} \{d_{\min A^{(l)}} d_{\min B^{(l)}}\}$ .

### 3.4 Multilevel Modulation Codes (MMC)

Multilevel modulation codes [28,36] can be constructed in a similar way to multilevel concatenated codes. In constructing an  $m$ -level modulation code,  $A^{(1)}$  is chosen as a signal space with  $2^m$  signal points, e.g.,  $2^m$ - PSK or  $2^m$ - QAM. First, a binary partition chain  $A^{(1)} / A^{(2)} / \dots / A^{(m)} / A^{(m+1)} = \{\mathbf{0}\}$  is formed and each signal point labeled by an  $m$ -bit label  $(a_1 a_2 \dots a_m)$ , where  $A^{(l+1)}$  is a subspace of  $A^{(l)}$  consisting of  $2^{m-l}$  signal points for  $1 \leq l \leq m$ , and  $\mathbf{0}$  represents the signal point labeled by the zero sequence of  $m$  bits. The labeling is accomplished by signal set partitioning [36]. The  $m$  outer codes  $B^{(1)}, B^{(2)}, \dots, B^{(m)}$  are chosen as  $m$  binary codes. For a set of  $m$  codewords  $\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)}$ , from  $B^{(1)}, B^{(2)}, \dots, B^{(m)}$ , respectively, form the following array:

$$\mathbf{b}^{(1)} = (b_1^{(1)}, b_2^{(1)}, \dots, b_j^{(1)}, \dots, b_N^{(1)})$$

$$\mathbf{b}^{(2)} = (b_1^{(2)}, b_2^{(2)}, \dots, b_j^{(2)}, \dots, b_N^{(2)})$$

⋮

$$\mathbf{b}^{(m)} = (b_1^{(m)}, b_2^{(m)}, \dots, b_j^{(m)}, \dots, b_N^{(m)})$$

If each column of the array is regarded as the label of a signal point in  $A^{(1)}$ , then the sequence  $\mathbf{b}^{(1)} \circ \mathbf{b}^{(2)} \circ \dots \circ \mathbf{b}^{(m)} \triangleq (b_1^{(1)} b_1^{(2)} \dots b_1^{(m)}, b_2^{(1)} b_2^{(2)} \dots b_2^{(m)}, \dots, b_N^{(1)} b_N^{(2)} \dots b_N^{(m)})$  is mapped into a sequence of  $N$  signals in  $A^{(1)}$ . Let  $\lambda(\cdot)$  denote the mapping defined by the signal labeling. Then  $\lambda(b_j^{(1)} b_j^{(2)} \dots b_j^{(m)}) = s_j$  is a signal point in  $A^{(1)}$ , and

$$\lambda(\mathbf{b}^{(1)} \circ \mathbf{b}^{(2)} \circ \dots \circ \mathbf{b}^{(m)}) = (\lambda(b_1^{(1)} b_1^{(2)} \dots b_1^{(m)}), \dots, \lambda(b_N^{(1)} b_N^{(2)} \dots b_N^{(m)})) = (s_1, s_2, \dots, s_N)$$

is a signal sequence. The following collection of signal sequences:

$$C = \{\lambda(\mathbf{b}^{(1)} \circ \mathbf{b}^{(2)} \circ \dots \circ \mathbf{b}^{(m)}) : \mathbf{b}^{(l)} \in B^{(l)} \text{ with } 1 \leq l \leq m\} \quad (3.4)$$

forms an  $m$ -level modulation code over  $A^{(1)}$ . This combining of coding and modulation is known as coded modulation [36]. Since block coding is used, it is called multilevel block coded modulation. If  $A^{(1)}$  is not partitioned into a binary chain, nonbinary outer codes can be used in constructing multilevel modulation codes. Many efficient multilevel MPSK and QAM codes have been constructed for both AWGN and fading channels.

### Parameters of MMC

For the  $m$ -level modulation code the minimum squared Euclidean distance is given as

$$d_E^2 = \min_{1 \leq l \leq m} \{ \delta_l \cdot d_{\min B^{(l)}} \}, \text{ where } \delta_l \text{ is the intraset distance [36] in level-}l.$$

### 3.5 Decomposition of RM codes as MCC

We have seen so far in this chapter that multilevel concatenation can be used to construct linear codes of large lengths that have large minimum distances. A problem very closely related to the multilevel code construction is code decomposition. A code is said to be decomposable if it can be expressed as a multilevel concatenated code. Such code can be decoded using efficient multistage decoding and therefore achieve good performance-complexity trade-off.

Not many codes have been proved to be decomposable in this way. Decomposition of some BCH and Euclidean Geometry (*EG*) codes are given in [37]. The most well known example of decomposable codes is the class of Reed-Muller (*RM*) codes [7]. We describe their decomposition in this section.

Consider the  $r^{\text{th}}$  order *RM* code of length  $2n$ , denoted  $\text{RM}(r, n)$ . The dimension of this code is

$$K(r, n) = \sum_{i=0}^r \binom{n}{i}$$

and the minimum Hamming distance  $2^{n-r}$ .

It was shown in [7] that for any  $\mu \leq r$ , the code  $\text{RM}(r, n)$  can be expressed as the following  $\mu$ -fold squaring construction,

$$\text{RM}(r, n) = |\text{RM}(r, n-\mu)/\text{RM}(r-1, n-\mu)/\dots/\text{RM}(r-\mu, n-\mu)|^{2^\mu}$$

where  $|A/B/\dots/Q|$  denotes a partition chain. For two positive integers  $i$  and  $j$  such that  $0 < i < j$ , let  $M(i, j)$  denote the dimension of the coset code  $[\text{RM}(i, j)/\text{RM}(i-1, j)]$ . Let  $(n, k, d)^p$  denote the  $2^p$ -ary code  $(n, k, d, 2^p)$  obtained by interleaving the binary  $(n, k, d)$  code by a depth of  $p$ , with each group of  $p$  interleaved bits regarded as a symbol in  $GF(2^p)$ .

The  $RM(r,n)$  code can be expressed as a  $(v+1)$ -level concatenated code, with  $1 \leq \mu \leq n-1$ ,  $v = \mu$  for  $r \geq \mu$ , and  $v = r$  otherwise, as follows [9]:

$$RM(r,n) = \{RM(0,\mu)^{M(r,n-\mu)}, RM(1,\mu)^{M(r-1,n-\mu)}, \dots, RM(v,\mu)^{K(r-v,n-\mu)}\} \\ \circ \{RM(r,n-\mu), RM(r-1,n-\mu), \dots, RM(r-v,n-\mu)\}.$$

From the previous expression we see that, by choosing different values for parameters  $\mu$  and an  $v$ ,  $RM$  code can be decomposed as a multilevel code in many ways and with different number of levels. All the component outer and inner codes in this decomposition are also  $RM$  codes, or subcodes of  $RM$  codes, with smaller lengths and dimensions. Two-level decompositions of some Reed-Muller codes are given in Table 3.1, and three-level decompositions in Table 3.2.

### Example 3.3

Consider the  $RM(3,6)$  code and its decomposition

$$(64, 42, 8) = \{(8, 1, 8)(8, 4, 4)^3, (8, 7, 2)^3, (8, 8, 1)\} \circ \{(8, 8, 1), (8, 4, 4)\}$$

given in [9]. Here, outer codes are  $B^{(1)} = (8, 1, 8)(8, 4, 4)^3$ , and  $B^{(2)} = (8, 7, 2)^3(8, 8, 1)$ .

We can see that the non-binary codes  $B^{(1)}$  and  $B^{(2)}$  consist each of four interleaved binary codes. Thus, the symbols of  $B^{(1)}$  and  $B^{(2)}$  are from  $GF(16)$ .

**Table 3.1 Two-level Decompositions of Some Reed-Muller Codes**

Code	Decoding Stage	Outer Code	Inner Code Partition
$RM(1,n)$	1	$(4,1,4)^{n-2}$	$[RM(1,n-2)/RM(0,n-2)]$
	2	$(4,3,2)$	$[RM(0,n-2)/\{0\}]$
$RM(n-2,n)$	1	$RM(n-3,n-1)$	$[(2,2,1)/(2,1,2)]$
	2	$RM(n-2,n-1)$	$[(2,1,2)/\{0\}]$
$RM(2,5)$	1	$(4,1,4)^3$	$[(8,7,2)/(8,4,4)]$
	2	$(4,3,2)^3$ $(4,4,1)$	$[(8,4,4)/(8,1,8)]$ $[(8,1,8)/\{0\}]$
	1	$(4,1,4)^6$	$[(16,11,4)/(16,5,8)]$
$RM(2,6)$	2	$(4,3,2)^4$ $(4,4,1)$	$[(16,5,8)/(16,1,16)]$ $[(16,1,16)/\{0\}]$
	1	$(8,1,8)$ $(8,4,4)^3$	$[(8,8,1)/(8,7,2)]$ $[(8,7,2)/(8,4,4)]$
$RM(3,6)$	2	$(8,7,2)^3$ $(8,8,1)$	$[(8,4,4)/(8,1,8)]$ $[(8,1,8)/\{0\}]$

**Table 3.2 Three-level Decompositions of Some Reed-Muller Codes**

Code	Decoding Stage	Outer Code	Partition
$RM(2,7)$ (128,29,32)	1	$(16,1,16)^3$ $(16,5,8)$	$[(8,7,2)/(8,4,4)]$ $[(8,4,4)/(8,3,4)]$
	2	$(16,5,8)^2$	$[(8,3,4)/(8,1,8)]$
	3	$(16,11,4)$	$[(8,1,8)/\{0\}]$
$RM(3,7)$ (128,64,16)	1	$(16,1,16)$ $(16,5,8)^2$	$[(8,8,1)/(8,7,2)]$ $[(8,7,2)/(8,5,2)]$
	2	$(16,5,8)$ $(16,11,4)$	$[(8,5,2)/(8,4,4)]$ $[(8,4,4)/(8,3,4)]$
	3	$(16,11,4)^2$ $(16,11,4)$	$[(8,3,4)/(8,1,8)]$ $[(8,1,8)/\{0\}]$
$RM(4,7)$ (128,99,8)	1	$(16,5,8)$ $(16,11,4)$	$[(8,8,1)/(8,7,2)]$ $[(8,7,2)/(8,6,2)]$
	2	$(16,11,4)^2$	$[(8,6,2)/(8,4,4)]$
	3	$(16,15,2)^3$ $(16,16,1)$	$[(8,4,4)/(8,1,8)]$ $[(8,1,8)/\{0\}]$

The trellis of  $B^{(l)}$ , for  $l=1,2$  can be obtained as a direct product of the trellises of its binary component codes. This trellis has the number of states equal to the product of the number of states, and number of incoming branches equal to the product of the number of incoming branches of the component code trellises at each section.

In this chapter we presented the multilevel concatenation as well as decomposition of codes into multilevel concatenated structure. The main advantage of codes constructed (or decomposed) in such a way is that they can be decoded using multistage decoding algorithm. This algorithm is presented in the next chapter.

## **Chapter 4**

### **Multistage Decoding**

Decoding of long codes represents a challenge for implementation, as decoding complexity usually grows exponentially with length. The more complicated the structure of the code, the more complex decoding algorithm is needed. One of the widely used methods for decoding codes whose trellises are available, is Viterbi decoding algorithm. However, the decoding complexity of this algorithm even for codes of medium lengths, say  $n=64$ , becomes too large for real implementation [3, 12, 37].

Multistage decoding (MSD) [4, 5] is devised for decoding decomposable codes and multilevel codes [4, 8] to achieve an efficient trade-off between error performance and decoding complexity. This algorithm utilizes multilevel structure to reduce the decoding of the overall multilevel code to decoding of its component codes, each of which can be decoded with much smaller decoding complexity. The component codes are decoded sequentially, level by level, with decoded information being passed from one decoding stage to the next.

In a conventional soft-decision MSD scheme, each stage uses soft-decision decoding, but hard-decision decoded output is used to modify the soft-decision received sequence for the next stage decoding. Decoded information is passed down from one

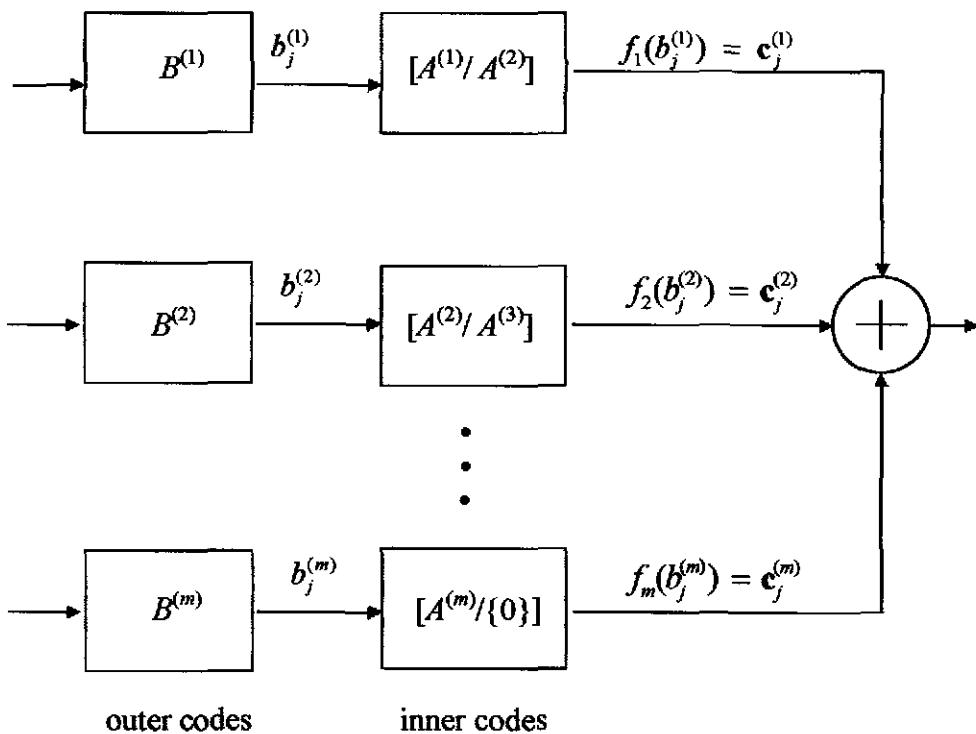
stage to the next until the last decoding stage. With this decoding, incorrect decoded information at one stage may result in error propagation through the subsequent decoding stages. Therefore, the decoding is not maximum likelihood (ML) even if every decoding stage is ML – it is a suboptimum decoding algorithm.

It was shown [9] that multistage decoding works well for short to moderately long codes. The trade-off between the error performance and decoding complexity is excellent. For longer codes, the trade-off is still good, as the decoding complexity is much smaller than the complexity of ML algorithms, say Viterbi algorithm. However, the performance degradation compared to ML decoding becomes significant [12].

Multistage decoding is widely used algorithm for multilevel codes. This method is also a base for the new algorithm presented in Chapter 5. Therefore, we describe the MSD algorithm in detail here. We also provide performance results for several codes, and make comparison to ML decoding in terms of performance, and Viterbi algorithm in terms of decoding complexity. Finally, we describe a modification of conventional MSD, namely List MSD [12, 16], which improves the performance of this suboptimum decoding technique for long codes.

## 4.1 Multistage Decoding Algorithm for multilevel concatenated codes

Consider the  $m$ -level concatenated code  $C = \{B^{(1)}, B^{(2)}, \dots, B^{(m)}\} \circ \{A^{(1)}, A^{(2)}, \dots, A^{(m)}\}$  described in section 3.1.1. The encoder for this code is given in Figure 3.3, and is reproduced here, for convenience, as Figure 4.1. Most important properties of the concatenated code  $C$  are reviewed as well.



**Figure 4.1 Multilevel concatenated scheme**

The  $m$  inner codes are coset codes formed from a binary linear block code  $A^{(1)}$  of length  $n$ , and a sequence of  $m$  linear subcodes of  $A^{(1)}$ , denoted by  $A^{(2)}, A^{(3)}, \dots, A^{(m)}=\{\mathbf{0}\}$ ,

with  $A^{(l+1)} \subset A^{(l)}$  for  $1 \leq l \leq m$ . The inner code at level- $l$ ,  $[A^{(l)} / A^{(l+1)}]$ , is a set of coset representatives of  $A^{(l)} / A^{(l+1)}$ , i.e.,  $[A^{(l)} / A^{(l+1)}] = \{\mathbf{a}_{r_j}^{(l)} : 1 \leq j \leq q_l\}$ , where  $\mathbf{a}_{r_j}^{(l)}$  denotes a coset representative.

For  $1 \leq l \leq m$ , the outer code  $B^{(l)}$  is an  $(N, K_l)$  linear block code over  $GF(q_l)$ , where  $q_l \triangleq |[A^{(l)} / A^{(l+1)}]|$ . Let  $f_l(\cdot)$  be a one-to-one mapping from  $GF(q_l)$  onto  $[A^{(l)} / A^{(l+1)}]$  that describes inner code encoding at level- $l$ . Then  $f_l(\cdot)$  maps each symbol of the alphabet of the outer code  $B^{(l)}$  to a coset representative from  $[A^{(l)} / A^{(l+1)}]$ .

Each codeword in  $A^{(l)}$  is a unique sum of  $m$  coset representatives of  $[A^{(l)} / A^{(2)}]$ ,  $[A^{(2)} / A^{(3)}]$ , ...,  $[A^{(m)} / \{0\}]$ , respectively. Thus, a codeword in  $A^{(l)}$  can be uniquely determined by  $m$  symbols from the alphabets of the outer codes  $B^{(1)}, B^{(2)}, \dots, B^{(m)}$ , respectively.

Let  $\mathbf{b}^{(l)} = (b_1^{(l)}, b_2^{(l)}, \dots, b_N^{(l)})$  be a codeword in the level- $l$  outer code  $B^{(l)}$ . During the inner encoding of level- $l$ ,  $\mathbf{b}^{(l)}$  is encoded into the following sequence,

$$\mathbf{c}^{(l)} = (\mathbf{c}_1^{(l)}, \mathbf{c}_2^{(l)}, \dots, \mathbf{c}_N^{(l)}) = (f_l(b_1^{(l)}), f_l(b_2^{(l)}), \dots, f_l(b_N^{(l)})) \quad (4.1)$$

where  $\mathbf{c}_j^{(l)} = f_l(b_j^{(l)})$  is a coset representative in  $[A^{(l)} / A^{(l+1)}]$ . Therefore, a codeword of component code  $C^{(l)} \triangleq B^{(l)} \circ [A^{(l)} / A^{(l+1)}]$  at level- $l$ ,  $\mathbf{c}^{(l)}$ , is a sequence of coset representatives from  $[A^{(l)} / A^{(l+1)}]$  such that  $f_l^{-1}(\mathbf{c}^{(l)}) \in B^{(l)}$ .

Every codeword  $\mathbf{c} \in C$  is a sum of  $m$  codewords  $\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \dots, \mathbf{c}^{(m)}$  from the  $m$  concatenated component codes  $C^{(1)}, C^{(2)}, \dots, C^{(m)}$  respectively, i.e.,

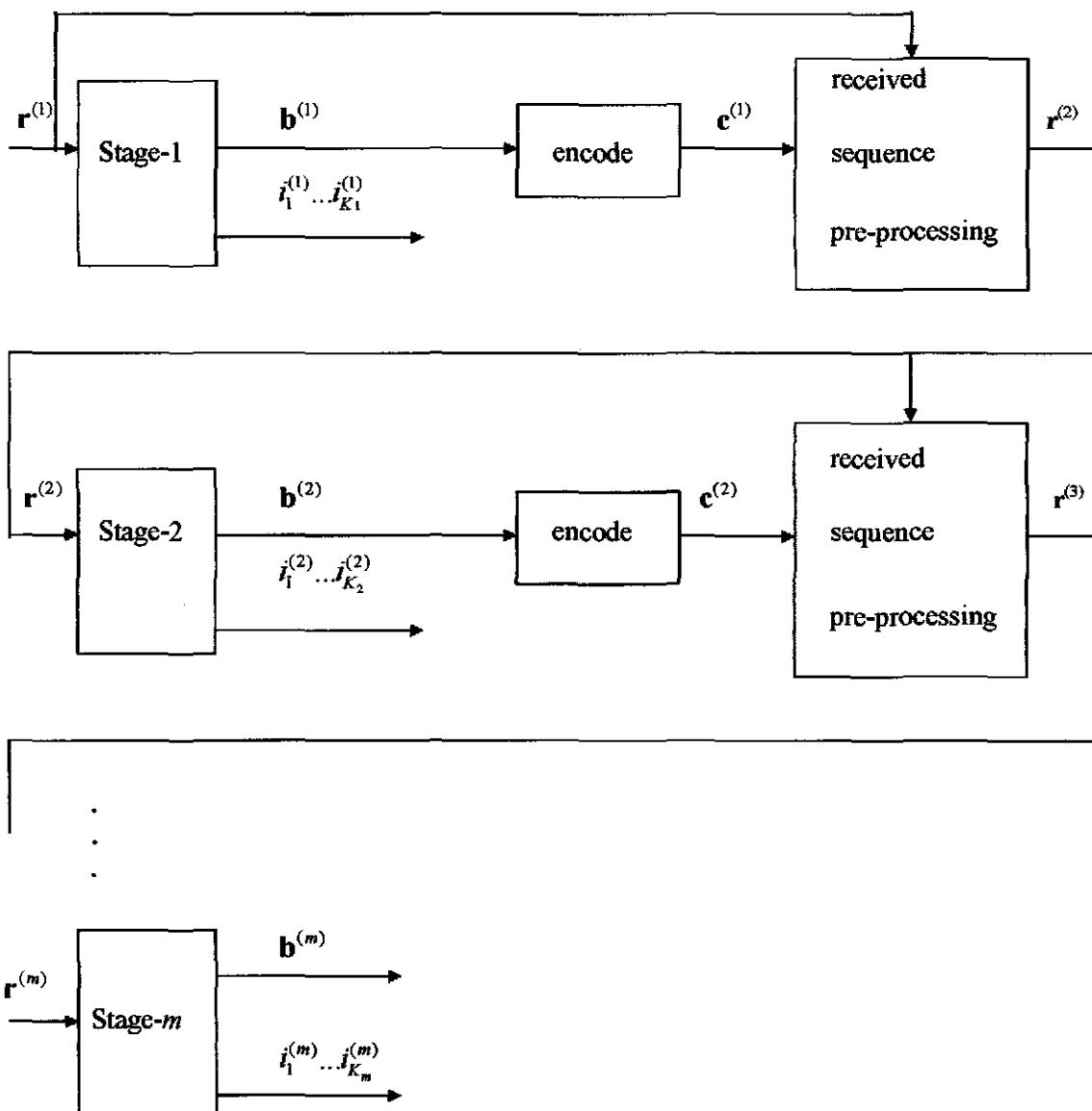
$$\mathbf{c} = \mathbf{c}^{(1)} + \mathbf{c}^{(2)} + \dots + \mathbf{c}^{(m)}. \quad (4.2)$$

From (4.1) and (4.2), we obtain

$$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N) = \left( \sum_{l=1}^m f_l(b_1^{(l)}), \sum_{l=1}^m f_l(b_2^{(l)}), \dots, \sum_{l=1}^m f_l(b_N^{(l)}) \right) \quad (4.3)$$

Therefore, section- $j$ ,  $\mathbf{c}_j, 1 \leq j \leq N$ , of the codeword  $\mathbf{c} \in C$ , is uniquely determined by the symbols  $b_j^{(l)}$  at  $j^{\text{th}}$  position of the outer code codewords  $\mathbf{b}^{(l)}, 1 \leq l \leq m$ .

Suppose a codeword  $\mathbf{c}_{tr} = f_1(\mathbf{b}_{tr}^{(1)}) \oplus f_2(\mathbf{b}_{tr}^{(2)}) \oplus \dots \oplus f_m(\mathbf{b}_{tr}^{(m)})$  corresponding to  $m$  outer code codewords  $\mathbf{b}_{tr}^{(1)}, \mathbf{b}_{tr}^{(2)}, \dots, \mathbf{b}_{tr}^{(m)}$  is transmitted. Let  $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$  be the received sequence at the output of the matched filter in the receiver, where section  $\mathbf{r}_j = (r_{j1}, r_{j2}, \dots, r_{jn})$  consists of  $n$  real numbers. The received sequence  $\mathbf{r}$  represents a noisy BPSK signal sequence corresponding to the transmitted codeword  $\mathbf{c} \in C$ . At each position, received sequence symbol is a sum of the transmitted signal and the additive noise. Assume BPSK signaling with the mapping of codeword bit  $c$  into BPSK signal described by  $c \rightarrow (-1)^c$ . Then for  $1 \leq j \leq N$ , and  $1 \leq s \leq n$ ,  $r_{j,s} = (-1)^{\mathbf{c}_{tr,j}} + \mathcal{N}_{j,s}$ , where  $\mathcal{N}_{j,s}$  denotes the additive noise component at position ' $j,s$ '.



**Figure 4.2 Schematic representation of MSD**

Multistage decoding consists of  $m$  decoding stages, one stage for each level of the multilevel concatenated code. At the first decoding stage, the received sequence  $\mathbf{r}^{(1)} \triangleq \mathbf{r}$  is decoded into a codeword  $\mathbf{b}^{(1)}$  in the first outer code  $B^{(1)}$ . Then the effect of the decoded estimate  $\mathbf{b}^{(1)}$  (or  $\mathbf{c}^{(1)}$ ) is removed from  $\mathbf{r}^{(1)}$ . This results in a modified received

sequence  $\mathbf{r}^{(2)} = (\mathbf{r}_1^{(2)}, \mathbf{r}_2^{(2)}, \dots, \mathbf{r}_N^{(2)})$  with  $\mathbf{r}_j^{(2)} = (r_{j,1}^{(2)}, r_{j,2}^{(2)}, \dots, r_{j,n}^{(2)})$ . The modified sequence  $\mathbf{r}^{(2)}$  is then used for decoding at the second stage and is decoded into a codeword  $\mathbf{b}^{(2)} \in B^{(2)}$ . Again, the effect of  $\mathbf{b}^{(2)}$  is removed from  $\mathbf{r}^{(2)}$  to obtain a modified received sequence  $\mathbf{r}^{(3)}$  for decoding in stage-3. This continues until the estimate  $\mathbf{b}^{(m)}$  at the last decoding stage is obtained. The schematic representation of multistage decoding is given in Figure 4.2.

Each stage of decoding consists of three steps:

- 1) Preprocessing (modifying) of the received sequence,
- 2) Inner decoding, and
- 3) Outer decoding.

After each stage of decoding is completed, information about the decoded estimate is passed to the following stage.

#### **Pre-processing of the received sequence at stage-*l*:**

The received sequence used at decoding stage-*l* is prepared in such a way that it only contains the contributions from the codewords  $\mathbf{c}^{(l)}, \mathbf{c}^{(l+1)}, \dots$  and  $\mathbf{c}^{(m)}$ . Thus, the contributions of the codewords from the levels before *l* need to be removed.

1. The received sequence used in the first stage,  $\mathbf{r}^{(1)}$  is the same as the received sequence  $\mathbf{r}$ , i.e.,  $\mathbf{r}^{(1)} \triangleq \mathbf{r}$ .

2. Let for  $l \geq 1$   $\mathbf{c}^{(l)} = f_l(\mathbf{b}^{(l)})$  be the estimate of stage- $l$ , where  $\mathbf{b}^{(l)}$  denotes the estimate of the level- $l$  outer decoder. Then the received sequence at stage- $(l+1)$ ,  $\mathbf{r}^{(l+1)}$ , is obtained from  $\mathbf{r}^{(l)}$  and  $\mathbf{c}^{(l)}$  in the following manner:

$$r_{j,s}^{(l+1)} = r_{j,s}^{(l)} \cdot (-1)^{c_{j,s}^{(l)}}$$

where  $r_{j,s}^{(l)}$  denotes the symbol in position ‘ $s$ ’ of the section- $j$  of the received sequence  $\mathbf{r}^{(l)}$ , and  $c_{j,s}$  the bit in position ‘ $s$ ’ of the section- $j$  of the estimate  $\mathbf{c}^{(l)} = \mathbf{c}_1^{(l)} \mathbf{c}_2^{(l)} \dots \mathbf{c}_N^{(l)}$ .

### **Inner code decoding of stage- $l$ :**

Decoding of inner codes is performed using the closest coset decoding algorithm [42]. The number of cosets at level- $l$  is equal to the size of the outer code alphabet,  $q_l$ . Coset  $\{f_l(\beta^{(l)})\}$ , corresponding to the symbol  $\beta^{(l)}$ , is defined as

$$\{f_l(\beta^{(l)})\} = \left\{ f_l(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)} : \mathbf{a}^{(l+1)} \in A^{(l+1)} \right\}.$$

For each coset, the codeword closest to the section- $j$ ,  $\mathbf{r}_j^{(l)} = r_{j,1}^{(l)} r_{j,2}^{(l)} \dots r_{j,n}^{(l)}$ , of the received sequence  $\mathbf{r}^{(l)}$  is found, and its metric is used as coset metric. Metric of the symbol  $b_j^{(l)}$  at position  $j$ ,  $1 \leq j \leq N$ , is defined as the metric of its corresponding coset, that is,

$$M(b_j^{(l)}) = M(\{f_l(b_j^{(l)})\}) = \underset{\mathbf{a}^{(l+1)} \in A^{(l+1)}}{\text{best}} M(\mathbf{r}_j^{(l)}, f_l(b_j^{(l)}) \oplus \mathbf{a}^{(l+1)}).$$

Function best finds the best metric of a codeword depending on how the metric is defined. For correlation metric, function best finds the maximum, and for squared Euclidean distance, the minimum value.

Let  $\mathbf{a}^{(l+1)*}(b_j^{(l)}, \mathbf{r}_j^{(l)})$  denote the codeword in  $A^{(l+1)}$  such that

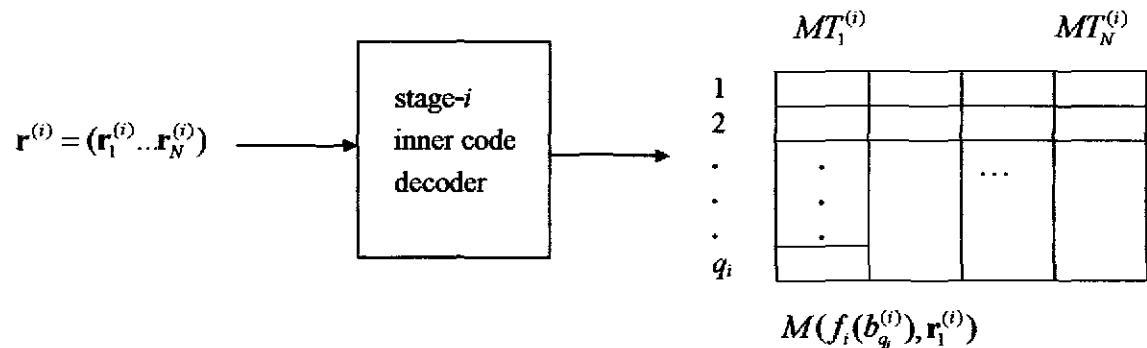
$$M(f_l(b_j^{(l)}) \oplus \mathbf{a}^{(l+1)*}(b_j^{(l)}, \mathbf{r}_j^{(l)}), \mathbf{r}_j^{(l)}) = \underset{\mathbf{a}^{(l+1)} \in A^{(l+1)}}{\text{best}} M(f_l(b_j^{(l)}) \oplus \mathbf{a}^{(l+1)}, \mathbf{r}_j^{(l)}),$$

i.e.,  $f_l(b_j^{(l)}) \oplus \mathbf{a}^{(l+1)*}(b_j^{(l)}, \mathbf{r}_j^{(l)})$  is the closest codeword in the coset  $\{f_l(b_j^{(l)})\}$  to the section  $\mathbf{r}_j^{(l)}$  of the received sequence. There is one such best codeword for each coset (correspondingly, to each symbol of the alphabet of  $B^{(l)}$ ) at each section  $1 \leq j \leq N$ . We call  $f_l(b_j^{(l)}) \oplus \mathbf{a}^{(l+1)*}(b_j^{(l)}, \mathbf{r}_j^{(l)})$  **coset winner** of the coset  $\{f_l(b_j^{(l)})\}$  with respect to  $\mathbf{r}_j^{(l)}$ .

**Definition 4.1:** For every coset in  $A^{(l)} / A^{(l+1)}$ , or correspondingly, for every symbol  $\beta^{(l)}$  of the outer code at level- $l$ ,  $B^{(l)}$ , define **coset winner** with respect to section  $\mathbf{r}_j^{(l)}$ ,  $1 \leq j \leq N$ , of the received sequence  $\mathbf{r}^{(l)}$ , as the codeword in that coset that has the best metric with respect to section  $\mathbf{r}_j^{(l)}$ . Define **coset metric** as the metric of the coset winner.

Let  $\beta^{(l)}$  be a symbol of  $B^{(l)}$ . Let  $\mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)})$  denote the codeword in  $A^{(l+1)}$  such that  $M(f_l(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)}), \mathbf{r}_j^{(l)}) = \underset{\mathbf{a}^{(l+1)} \in A^{(l+1)}}{\text{best}} M(f_l(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)}, \mathbf{r}_j^{(l)})$ . Then the coset winner and coset metric of  $\{f_l(\beta^{(l)})\}$  are respectively  $f_l(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)})$  and  $M(f_l(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)}))$ .

At stage- $i$  of decoding, the inner code decoder processes the  $N$  sections of the received sequence  $\mathbf{r}^{(i)} = (\mathbf{r}_1^{(i)}, \mathbf{r}_2^{(i)}, \dots, \mathbf{r}_N^{(i)})$  independently and forms  $N$  tables, one for each section of  $\mathbf{r}^{(i)}$ . A table denoted by  $MT_j^{(i)}$  stores the metric  $M(\{\mathbf{a}_{\mathbf{r}}^{(i)}\}, \mathbf{r}_j^{(i)})$  for each of the  $q_i$  cosets  $\{\mathbf{a}_{\mathbf{r}}^{(i)}\} \in A^{(i)} / A^{(i+1)}$ . This table is called the **metric table** for  $\mathbf{r}_j^{(i)}$ . The inner decoding at stage- $i$  involves forming  $N$  metric tables, one for each section of the received sequence  $\mathbf{r}_j^{(i)}$ . These  $N$  metric tables are then passed to the outer code decoder. The process of inner code decoding is represented in Figure 4.3.



**Figure 4.3 Stage- $i$  inner code decoding**

#### Outer code decoding at stage- $i$ :

Let  $\mathbf{b}^{(i)} = (b_1^{(i)}, b_2^{(i)}, \dots, b_N^{(i)})$  be a codeword in  $B^{(i)}$ . The metric of the outer code codeword  $\mathbf{b}^{(i)}$  is equal to the sum of metrics of its symbols, i.e.,

$$M(\mathbf{b}^{(i)}) = \sum_{j=1}^N M(b_j^{(i)}).$$

Since the metric of the symbol  $b_j^{(i)}$  is defined as the metric of the coset into which  $b_j^{(i)}$  is mapped by  $f_i(\cdot)$ , we can further write the expression for the metric of  $\mathbf{b}^{(i)}$  as

$$M(\mathbf{b}^{(i)}) = \sum_{j=1}^N M(\{f_i(b_j^{(i)})\}) \quad (4.5)$$

where  $M(\{f_i(b_j^{(i)})\})$  is the metric of the coset  $\{f_i(b_j^{(i)})\}$  with  $f_i(b_j^{(i)})$  as the coset representative. The outer code decoding at stage- $i$ , as represented in Figure 4.4, involves finding the codeword  $\hat{\mathbf{b}}^{(i)} \in B^{(i)}$ , which has the best metric among all the codewords in  $B^{(i)}$ , i.e.,

$$\hat{\mathbf{b}}^{(i)} = \arg \min_{\mathbf{b} \in B^{(i)}} M(\mathbf{b})$$

This can be achieved by using a trellis-based decoding algorithm, such as the Viterbi algorithm [2], or some other ML decoding algorithm.

#### METRIC TABLES

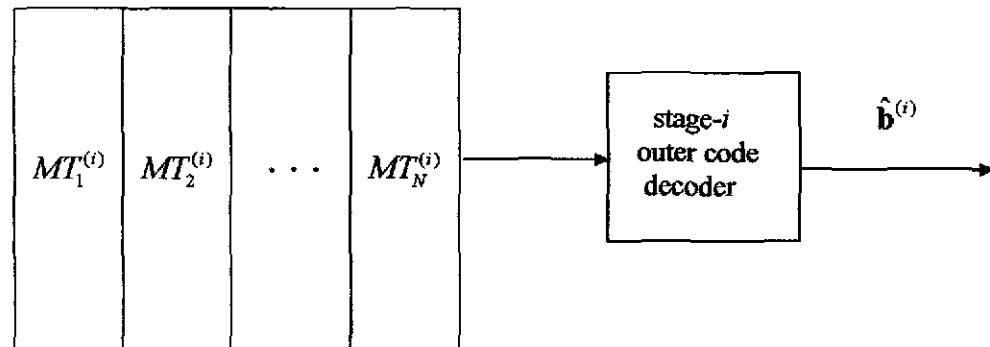


Figure 4.4 Stage- $i$  outer code decoding

Once the stage- $I$  of the decoding is completed, the information about decoded estimate is passed to the stage-( $I+1$ ). The three-step procedure is repeated for all levels of the multilevel concatenated code.

## 4.2 Multistage Decoding Algorithm for multilevel modulation codes

In this section, we describe the MSD algorithm applied to Block Coded Modulation (BCM) codes.

As mentioned in section 3.4, modulation codes are based on the partition of the signal constellation  $A^{(I)}$  into subconstellations in which the intra-symbol distance is increasing. Due to similarity between BCM codes and multilevel concatenated codes (MCC) described in 3.4, the algorithm described in 4.1 can be applied for decoding of BCM codes with a few minor modifications.

There are two basic modifications of the algorithm concerning:

1. Details of inner decoding, and
2. Type of information passed to the subsequent decoding stage.

Let  $\mathbf{b}^{(I)} = b_1^{(I)} b_2^{(I)} \dots b_N^{(I)}$  be a codeword in outer code  $B^{(I)}$ . During the inner encoding of level- $I$ ,  $\mathbf{b}^{(I)}$  is encoded into the following sequence:

$$\mathbf{s}^{(I)} = (\lambda_I(b_1^{(I)}), \lambda_I(b_2^{(I)}), \dots, \lambda_I(b_N^{(I)}))$$

where  $\lambda_i(b_j^{(l)})$  is a label of a signal sub-constellation  $A^{(l)} / A^{(l+1)}$ . A signal sequence of the component modulation code  $C^{(l)}$  at level- $l$  is a sequence of signal sub-constellation labels such that

$$\lambda_l^{-1}(\mathbf{s}^{(l)}) \in B^{(l)}$$

A signal-codeword  $\mathbf{s} \in C$  is obtained as a combination of signal words  $\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \dots, \mathbf{s}^{(m)}$  from  $m$  component modulation codes, i.e.,

$$\mathbf{s} = \underbrace{\lambda_1(b_1^{(1)})\lambda_2(b_1^{(2)})\dots\lambda_m(b_1^{(m)})}_{s_1} \underbrace{\lambda_1(b_2^{(1)})\lambda_2(b_2^{(2)})\dots\lambda_m(b_2^{(m)})}_{s_2} \dots \underbrace{\lambda_1(b_N^{(1)})\lambda_2(b_N^{(2)})\dots\lambda_m(b_N^{(m)})}_{s_N}.$$

Each section  $s_j$  represents a point in signal space  $A^{(l)}$  that is uniquely determined by the partition labels  $\lambda_1(b_j^{(1)})\lambda_2(b_j^{(2)})\dots\lambda_m(b_j^{(m)})$ , i.e. by  $b_j^{(1)}b_j^{(2)}\dots b_j^{(m)}$ .

Suppose a signal sequence  $\mathbf{s}$  corresponding to outer code codewords  $\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \mathbf{b}^{(3)}, \dots, \mathbf{b}^{(m)}$  is transmitted. Then  $\mathbf{s}$  can be expressed as

$$\mathbf{s} = s_1 s_2 \dots s_N = \begin{Bmatrix} \mathbf{s}^{(1)} \\ \mathbf{s}^{(2)} \\ \vdots \\ \mathbf{s}^{(m)} \end{Bmatrix} = \begin{Bmatrix} \frac{\lambda_1(\mathbf{b}^{(1)})}{\lambda_2(\mathbf{b}^{(2)})} \\ \frac{\lambda_2(\mathbf{b}^{(2)})}{\lambda_3(\mathbf{b}^{(3)})} \\ \vdots \\ \frac{\lambda_m(\mathbf{b}^{(m)})}{\lambda_1(\mathbf{b}^{(1)})} \end{Bmatrix}$$

Let  $\mathbf{r}_s = (\mathbf{r}_{s_1}, \mathbf{r}_{s_2}, \dots, \mathbf{r}_{s_N})$  be the received signal sequence where  $\mathbf{r}_{s_j} = (\mathbf{r}_{x_j}, \mathbf{r}_{y_j})$  consists of a pair of real numbers and  $\mathbf{r}_{s_j} = \mathbf{s}_j + \mathbf{n}_j = (s_{x_j}, s_{y_j}) + (n_{x_j}, n_{y_j})$ .

## Inner decoding

Let  $N$  be the length of the outer codes and  $q_l$  the size of the alphabet of the outer code, which is equal to the number of signal sub-constellations in the partition  $[A^{(l)} / A^{(l+1)}]$ .

Like in inner decoding of MCC,  $N$  metric tables are formed, one for each signal point  $\mathbf{r}_{s_j} = (r_{x_j}, r_{y_j})$  in the received signal sequence  $\mathbf{r} = \mathbf{r}_{s_1} \mathbf{r}_{s_2} \dots \mathbf{r}_{s_N}$ .

Let  $\text{sub}_\mu$  denote the set of points in sub-constellation- $\mu$  of the partition  $[A^{(l)} / A^{(l+1)}]$ . The metric of  $\text{sub}_\mu$  with respect to received point in position- $\mu$ ,  $\mathbf{r}_j = (r_{x_j}, r_{y_j})$ , is defined as the best metric of a signal point in  $\text{sub}_\mu$ , (here the best metric is the smallest Euclidean distance between any signal point in the subconstellation and the received signal point  $\mathbf{r}_j$ ). Metric table  $MT_j^{(l)}$  stores  $q_l$  metrics, one corresponding to each subconstellation  $\text{sub}_\mu$ , for  $1 \leq \mu \leq q_l$ .

## Passing the information to the next stage

For BCM types of codes, the same received sequence  $\mathbf{r}_s = (\mathbf{r}_{s_1}, \mathbf{r}_{s_2}, \dots, \mathbf{r}_{s_N})$  is used in all decoding stages. The information that is passed from stage to stage is a sequence of signal-sub-constellation labels

$$\lambda_1(b_j^{(1)}) \lambda_2(b_j^{(2)}) \dots \lambda_l(b_j^{(l)}), \quad 1 \leq j \leq N$$

The label at position- $j$  determines the signal sub-constellation that  $\mathbf{r}_{s_j}$  will be compared with during inner decoding of stage- $l$ .

### 4.3 Examples: Multistage decoding of RM codes

In this section, we present application of MSD to codes that belong to the class of decomposable codes described in section 3.5. We provide several examples, as well as the theoretical estimate of the performance of the multistage decoding.

As seen in [7] and summarized in section 3.5, *RM* codes can be decomposed into multilevel concatenated codes consisting of *RM* codes of smaller lengths and dimensions as inner and outer codes at each level:

$$RM(r,n) = \{RM(0,\mu)^{M(r,n-\mu)}, RM(1,\mu)^{M(r-1,n-\mu)}, \dots, RM(v,\mu)^{K(r-v,n-\mu)}\} \\ \circ \{RM(r,n-\mu), RM(r-1,n-\mu), \dots, RM(r-v,n-\mu)\}. \quad (4.6)$$

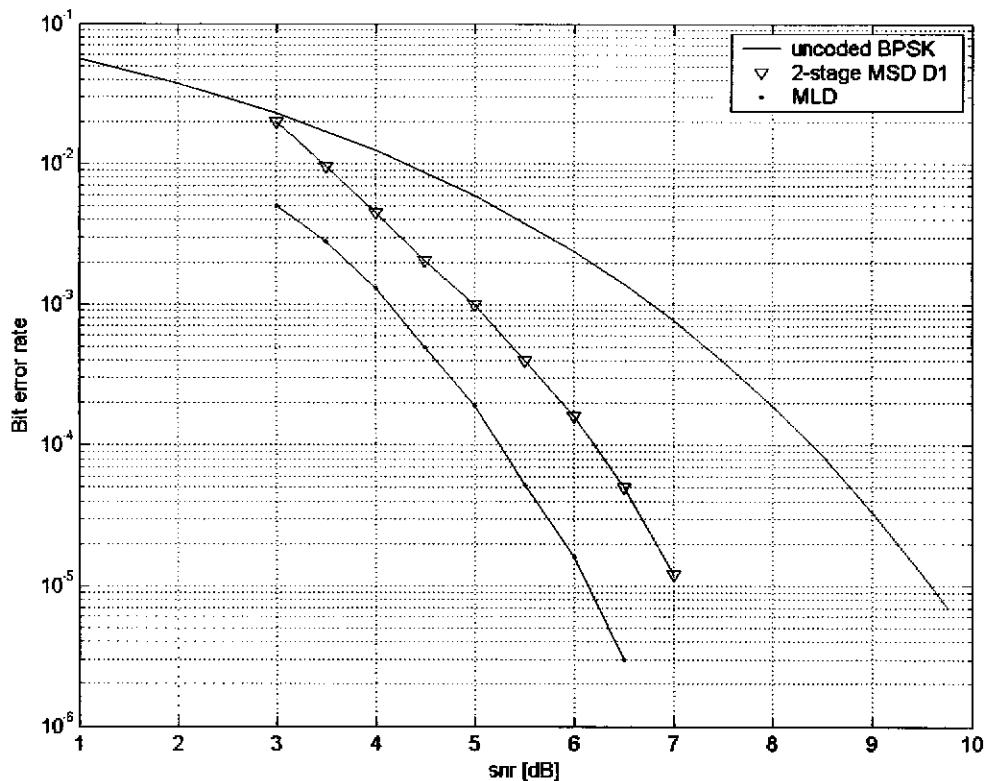
This multilevel concatenated structure allows the application of the MSD algorithm described in section 4.1. Performance results for several codes are reported in the following examples.

#### Example 4.1

Consider the  $RM(1,5)$  code, which is a  $(32,6,16)$  code of length  $n=32$ , dimension  $k=6$ , and minimum distance  $d=16$ . For  $\mu=4$  and  $v=2$ , we obtain two-level decomposition

$$D_1 : (32,6,16) = \{(16,1,16),(16,5,8)\} \circ \{(2,2,1),(2,1,2)\}.$$

The performance of this code [9] over AWGN channel when MSD algorithm is used for decoding is represented in Figure 4.5. We can see that MSD decoding loses 0.8 dB at  $P_e = 10^{-5}$  compared to MLD. This result does not justify our claims that the performance of MSD for short and medium length codes is excellent. However, it brings up an important question of choosing the correct decomposition for a given code.



**Figure 4.5 Performance of (32,6,16) RM code with 2-stage decoding**

Namely, it was mentioned in section 3.5 that there are many different decompositions of a given *RM* code. The parameter  $\mu$  in (4.6) determines the length of the inner codes, and parameter  $v$ , a function of  $\mu$  and  $r$ , the number of binary component codes of all the outer codes. For different values of parameter  $\mu$ , we obtain different decompositions of a code. Also, for a fixed  $\mu$ , the  $v+1$  binary component outer codes can be grouped in different ways to obtain an  $m$ -level concatenated code, where  $1 \leq m \leq v+1$ . It is important to choose the decomposition which has the best performance/complexity ratio.

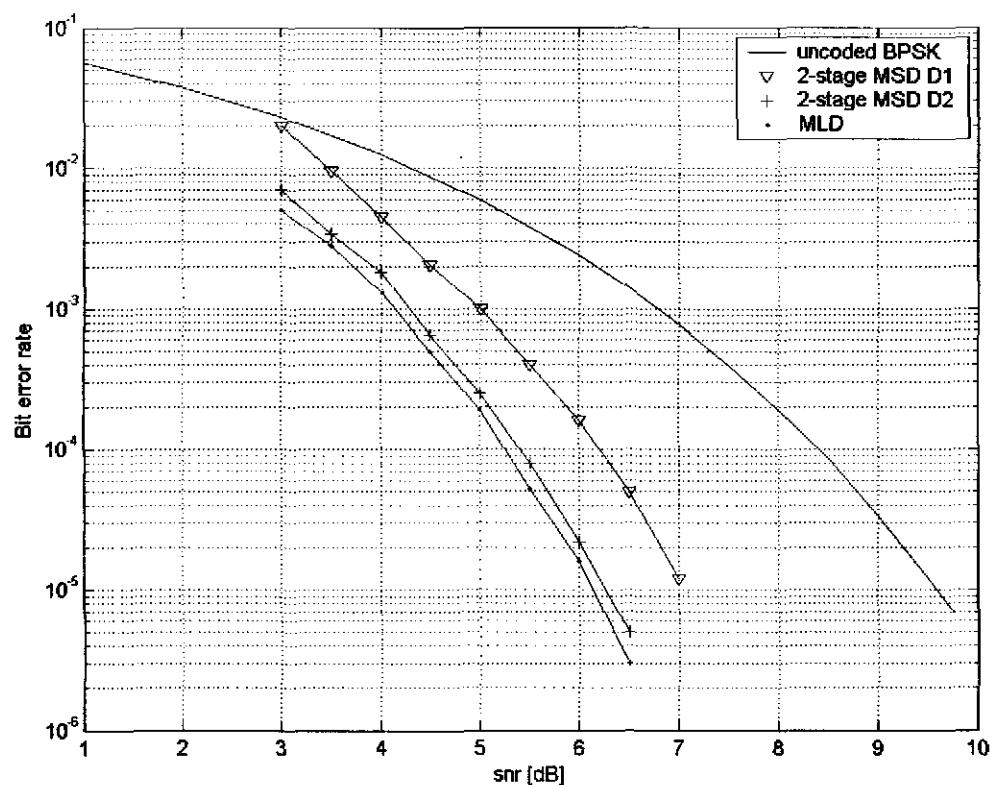
#### **Example 4.2**

Consider the  $RM(1,5)$  code from Example 4.1. For  $\mu=2$  and  $v=4$ , we obtain two-level decomposition

$$D_2 : (32, 6, 16) = \{(4, 1, 4)^3, (4, 3, 2)\} \circ \{(8, 4, 4), (8, 1, 8)\}$$

It was shown in [9] that the error coefficients for the decomposition  $D_1$  and  $D_2$  are 32783 and 59 respectively. Simulation results show that the decomposition  $D_2$  outperforms decomposition  $D_1$  in a two-stage decoding, thus being in agreement with the result on error coefficients. In the same paper, it was shown that, for two level decomposition, performance decreases when the number of redundant bits in the second level outer code increases. Therefore, it was suggested that a good two-level decomposition has a small number of redundant bits in the second level code. The bit-

error performance of the decomposition  $D_2$  in comparison to  $D_1$  is reproduced from [9] in Figure 4.6.



**Figure 4.6 Performance of two different decompositions of (32,6,16) RM code**

### **Example 4.3**

Consider the  $RM(3,6)$  code that is a  $(64,42,8)$  code. Choosing the parameters  $\mu = 3$  and  $\nu = 3$ , and grouping the binary component codes into two and three levels, we obtain the two-level decomposition  $D_3$  and the three-level decompositions  $D_4$  and  $D_5$  as

$$D_3 : (64,42,8) = \{(8,1,8)(8,4,4)^3, (8,7,2)^3(8,8,1)\} \circ \{(8,8,1), (8,4,4)\}$$

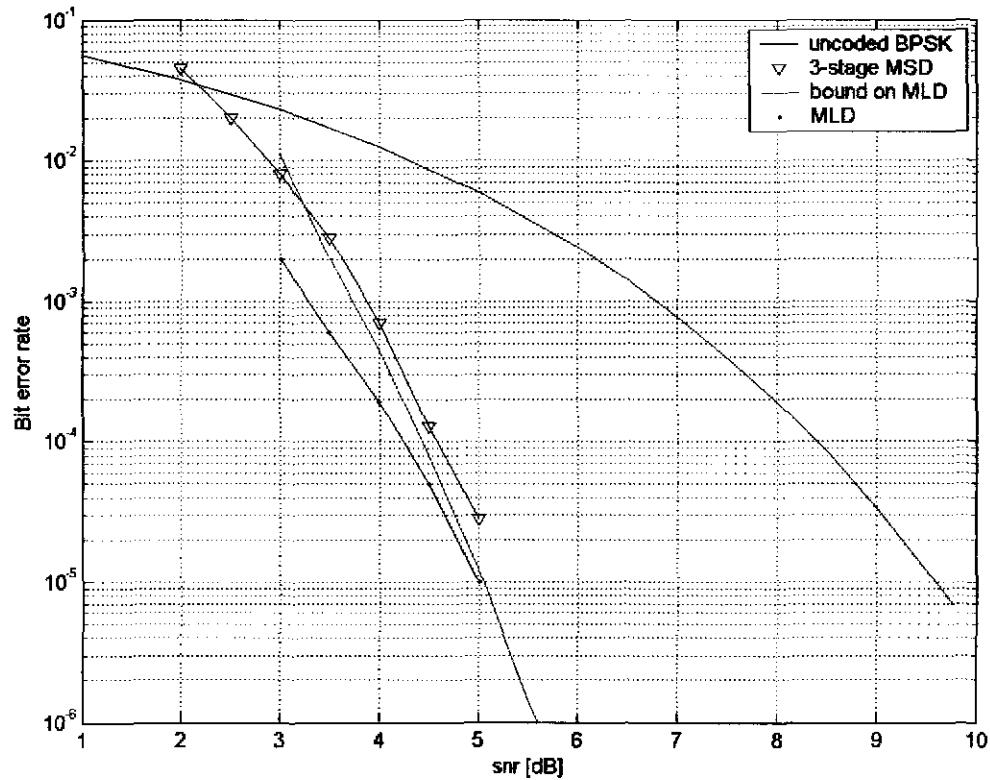
$$D_4 : (64,42,8) = \{(8,1,8)(8,4,4)^3, (8,7,2)^2, (8,7,2)(8,8,1)\} \circ \{(8,8,1), (8,4,4), (8,2,4)\}$$

and

$$D_5 : (64,42,8) = \{(8,1,8)(8,4,4), (8,4,4)^2, (8,7,2)^3(8,8,1)\} \circ \{(8,8,1), (8,6,4), (8,4,4)\}$$

The performances of decompositions  $D_3$  and  $D_4$  were shown [12] to be almost the same, and much better than that of composition  $D_5$ . The importance of having a strong code in the first level is emphasized. Here, we give the performance of three-stage decomposition  $D_3$  in Figure 4.7.

It can be seen from the figure that the three-stage decoding results in suboptimum performance, with 0.4 dB loss compared to ML decoding around  $BER = 10^{-5}$ .



**Figure 4.7 Performance of multistage decoding of (64,42,8) RM code**

#### Example 4.4

Consider the  $RM(3,7)$  code, which is a  $(128,64,16)$  code. For  $\mu=4$  and  $\nu=3$ , two possible decompositions are  $D_6$  (two-level) and  $D_7$  (three-level):

$$D_6 : (128,64,16) = \{(16,1,16)(16,5,8)^3, (16,11,4)^3(16,15,2)\} \circ$$

$$\{(8,8,1),(8,4,4)\}$$

and

$$D_7 : (128, 64, 16) = \{(16, 1, 16)(16, 5, 8)^2, (16, 5, 8)(16, 11, 4), (16, 11, 4)^2(16, 15, 2)\} \circ$$

$$\{(8, 8, 1), (8, 5, 2), (8, 3, 4)\}.$$

Simulation results for  $D_6$  are not available, since the trellis for Viterbi decoding of the first level outer code comprises of 8,192, and the trellis for the second level outer code of 4,096 states, when 8-section trellises are used. Both trellises have very large number of incoming branches at each section. Two problems arise with this decomposition: 1) storage problem, space is needed for the trellis representation (states and corresponding metrics, survivors, etc.), and 2) computational complexity problem, as large number of computations is needed to process the corresponding trellis.

Results for the decomposition  $D_7$  are given in Figure 4.8. The three stage decoding loses about 1.3 dB in bit error performance at  $\text{BER}=10^{-5}$ , compared to MLD. This example shows that, in case of codes of large lengths, the performance loss of multistage decoding compared to MLD is significant for the SNRs of interest.

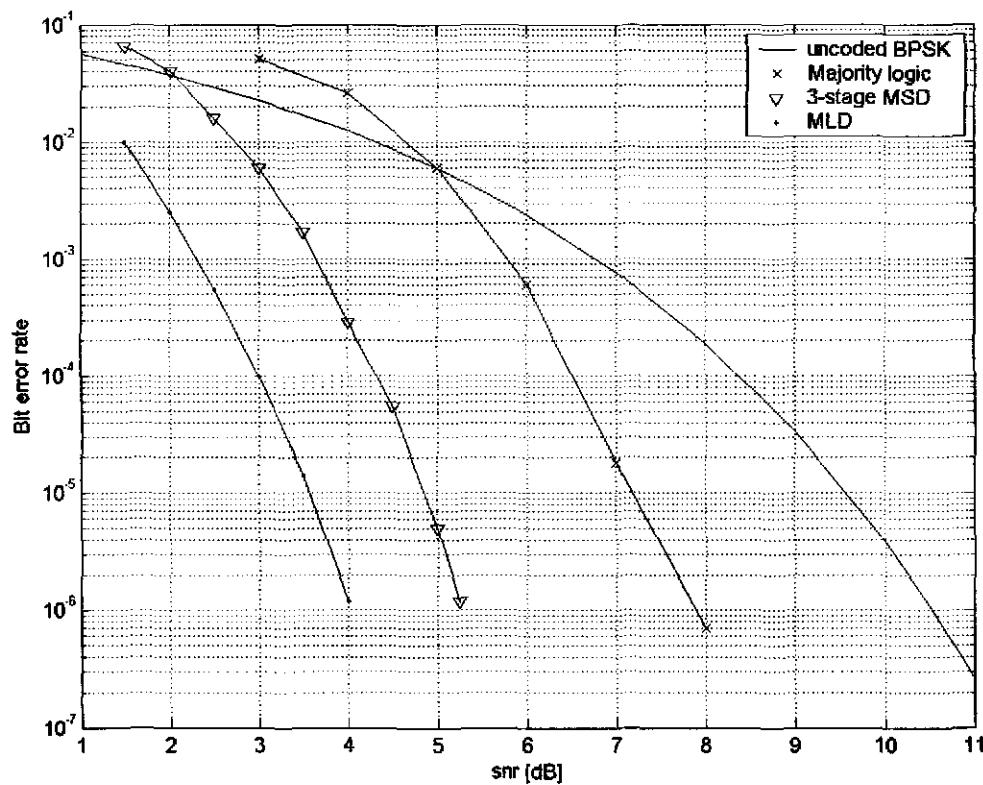
#### **Example 4.5**

Consider the  $RM(4, 7)$  code, which is a  $(128, 99, 8)$  code. By choosing parameters to be  $\mu = 4$  and  $\nu = 4$ , and grouping component codes into three levels, we obtain the three-level decomposition  $D_8$ ,

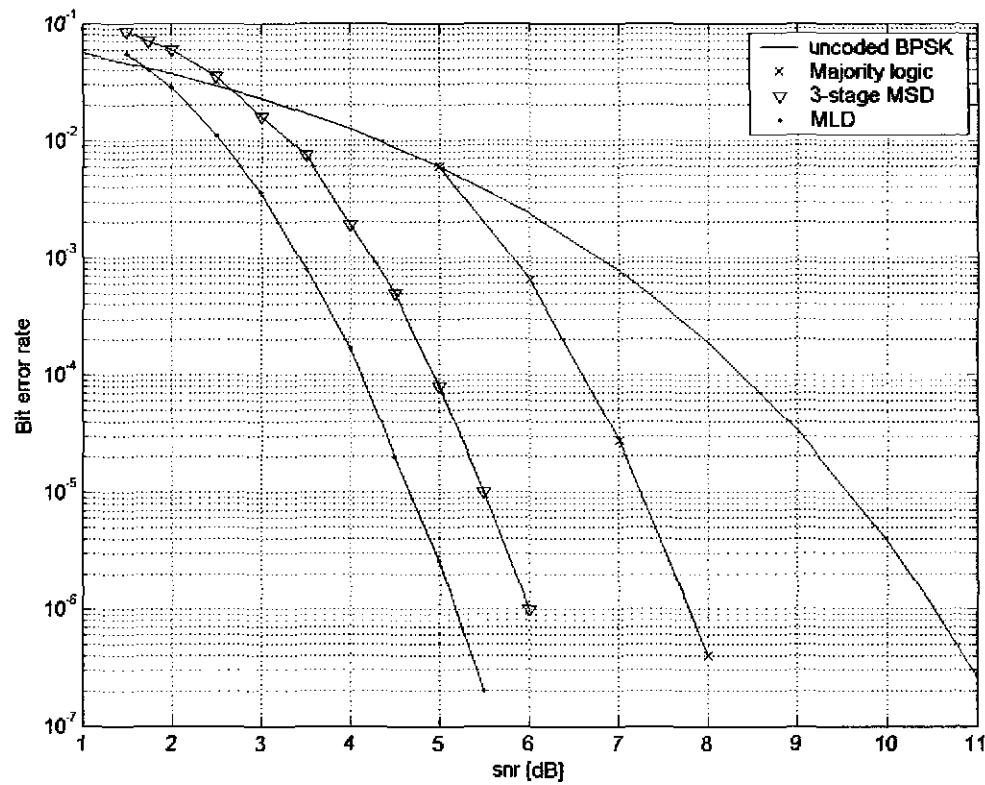
$$D_8 : (128, 99, 8) = \{(16, 5, 8)(16, 11, 4), (16, 11, 4)^2, (16, 15, 2)^3(16, 16, 1)\} \circ$$

$$\{(8, 8, 1), (8, 6, 2), (8, 4, 4)\}$$

The performance of this code using multistage decoding for the three-level decomposition is shown in Figure 4.9. The loss compared to MLD is 0.8 dB at BER=10<sup>-5</sup>.



**Figure 4.8 Performance of 3-stage decoding of (128,64,16) RM code**



**Figure 4.9 Performance of 3-stage decoding of (128,99,8) RM code**

We have seen in the given examples that different decompositions of the same code result in different error performances of multistage decoding. Thus, the choice of decomposition is very important. It was shown in [11, 12] that the performance of MSD is related to the structure, that is, weight distribution of component inner and outer codes. The upper bound on probability of block error was computed for several codes.

Overall, the results show that the multistage decoding performs well for *RM* codes of small and medium lengths if a good decomposition of a given code is chosen.

The loss in performance compared to MLD at  $P_e \approx 10^{-5}$  is of the order of few tenths of a dB for  $RM$  codes of length 64, and a few dB for  $RM$  codes of length 128. Since decoding complexity is drastically reduced, the tradeoff between the loss in performance and decrease in complexity can be considered satisfactory. It is of interest, though, to see if the error performance can be improved by introducing certain modifications into the conventional multistage decoding algorithm. We show in the next section that the error performance of MSD can be improved by passing a list of  $L$  decoded estimates with best metrics from one stage to the next. At the last stage, the codeword with the best metric (largest correlation, or smallest Euclidean distance) is chosen as the decoded codeword [12, 16].

#### 4.4 Performance comparison of MSD and ML decoding

Why is MSD of multilevel codes only asymptotically optimum, that is, why is the performance of MSD for practical SNRs not as good as that of ML decoding? The answer to this question comes from the observation of the closest coset decoding, which is used for decoding of inner codes.

Let the estimate of the first stage outer code  $B^{(1)}$  be  $\mathbf{b}^{(1)} = (b_1^{(1)}, b_2^{(1)}, \dots, b_N^{(1)})$ . Then the estimate of the stage-1,  $\mathbf{c}^{(1)}$ , is obtained as  $\mathbf{c}^{(1)} = f_1(\mathbf{b}^{(1)}) = f_1(b_1^{(1)})f_1(b_2^{(1)})\dots f_1(b_N^{(1)})$ . Metric of the estimate  $\mathbf{b}^{(1)}$  is equal to

$$M(\mathbf{b}^{(1)}) = \sum_{j=1}^N M(b_j^{(1)}) = \sum_{j=1}^N M(\{f_1(b_j^{(1)})\}).$$

Since the coset metric is defined as the metric of the coset winner, i.e.,

$$M(\{f_i(b_j^{(l)})\}) = M(f_i(b_j^{(l)}) \oplus \mathbf{a}_j^{(l+1)*}(b_j^{(l)}, \mathbf{r}_j^{(l)})) \quad (4.7)$$

where  $\mathbf{a}_j^{(l+1)*}(b_j^{(l)}, \mathbf{r}_j^{(l)}) \in A^{(l+1)}$ , we obtain

$$M(\mathbf{b}^{(l)}) = \sum_{j=1}^N M(f_i(b_j^{(l)}) \oplus \mathbf{a}_j^*(b_j^{(l)}, \mathbf{r}_j^{(l)})). \quad (4.8)$$

**Definition 4.2:** Let the inner code decoding with respect to the received sequence  $\mathbf{r}^{(l)}$  result in  $N$  metric tables containing for each coset in  $A^{(l)} / A^{(l+1)}$  the coset metric and the coset winner. Define **coset winner sequence** for the outer-code codeword  $\mathbf{b}^{(l)}$  as the sequence obtained by substituting each symbol  $b_j^{(l)}$  with the coset winner  $f_i(b_j^{(l)}) \oplus \mathbf{a}_j^*$ .

The coset winner sequence of the codeword  $\mathbf{b}^{(l)}$  is given by

$$L(\mathbf{b}^{(l)}) \triangleq (f_i(b_1^{(l)}) \oplus \mathbf{a}_1^{(l+1)*}, f_i(b_2^{(l)}) \oplus \mathbf{a}_2^{(l+1)*}, \dots, f_i(b_N^{(l)}) \oplus \mathbf{a}_N^{(l+1)*}) \quad (4.9)$$

where  $\mathbf{a}_j^{(l+1)*} \in A^{(l+1)}$  for  $1 \leq j \leq N$ .

**Lemma 4.1:** Metric of the outer code codeword  $\mathbf{b}^{(l)}$  is equal to the metric of the corresponding coset winner sequence.

Proof: Lemma 4.1 follows directly from (4.8) and (4.9).

Let  $\mathbf{b}^{(1)}$  denote the codeword in  $B^{(1)}$  with the best metric among the codewords in  $B^{(1)}$ . Let the metric of  $\mathbf{b}^{(1)}$  be denoted by  $M(\mathbf{b}^{(1)})$ . This metric is equal to the metric of the coset winner sequence of  $\mathbf{b}^{(1)}$ , however, it is not necessarily equal to the metric of any codeword in multilevel concatenated codes  $C$ . We have that

$$M(\mathbf{b}^{(1)}) = \max_{b_1^{(1)} b_2^{(1)} \dots b_N^{(1)} \in B^{(1)}} \sum_{j=1}^N M(f_1(b_j^{(1)}) \oplus \mathbf{a}^{(2)*}(b_j^{(1)}, \mathbf{r}_j^{(1)})),$$

$$\text{s.t. } M(f_1(b_j^{(1)}) \oplus \mathbf{a}^{(2)*}(b_j^{(1)}, \mathbf{r}_j^{(1)})) = \max_{\mathbf{a} \in A^{(2)}} M(f_1(b_j^{(1)}) \oplus \mathbf{a}).$$

Equivalently,

$$M(\mathbf{b}^{(1)}) = \max_{b_1^{(1)} b_2^{(1)} \dots b_N^{(1)} \in B^{(1)}} \sum_{j=1}^N \max_{\mathbf{a} \in A^{(2)}} M(f_1(b_j^{(1)}) \oplus \mathbf{a}), \text{ or}$$

$$M(\mathbf{b}^{(1)}) = \max_{\substack{b_1^{(1)} b_2^{(1)} \dots b_N^{(1)} \in B^{(1)} \\ \mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N \in (A^{(2)})^N}} \sum_{j=1}^N M(f_1(b_j^{(1)}) \oplus \mathbf{a}_j) \quad (4.10)$$

Thus, we maximize the summation in (4.10) over all codewords in the outer code  $B^{(1)}$ , and over all sequences  $\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N \in (A^{(2)})^N$ . However, only those expressions

$$f_1(\mathbf{b}^{(1)}) \oplus \mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N \quad (4.11)$$

for which  $\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N \in C$ , or alternatively,  $f_2^{-1}(\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N) \in B^{(2)}$ , will be valid codewords in  $C$ . There are total of  $(2^{k_2})^N = 2^{N \cdot k_2}$  sequences of the form (4.11) for a given  $\mathbf{b}^{(1)}$ , and only

$|B^{(2)}| = 2^{K_2+K_3+\dots+K_m}$  among them are codewords in  $C$ . Thus the multiplicity of sequences examined by the multistage decoding over the sequences examined by the ML decoder is

$2^{\frac{N \cdot k_2 - \sum_{j=2}^m K_j}{2}}$ . Strict bound on error probability would require more involved calculations and taking into account that error occurs only if the maximized sum in (4.10) occurs for  $\mathbf{b}^{(1)}$  that is not an ML solution. However, we can say that, roughly, MSD performance is

$2^{\frac{N \cdot k_2 - \sum_{j=2}^m K_j}{2}}$  times worse than that of ML decoding. This expression more or less fits the calculations of error coefficients presented by Wu et al. [9] for the decompositions  $D_1$  and  $D_2$  from examples 4.1 and 4.2, as well as the decompositions of the example 4.3.

In general, assuming that stages 1 through  $l-1$  produce ML estimate, the estimate of the stage- $l$  results in ML estimate on average 1 in  $2^{\frac{N \cdot k_{l+1} - \sum_{j=l+1}^m K_j}{2}}$  times. Substituting  $l=m$ , this number becomes  $2^{N \cdot 0 - 0} = 1$ , which agrees with the fact that the last stage decoding always produce ML estimate, given that all the previous stages result in estimates that are components of the ML solution.

## 4.5 List Multistage Decoding (LMSD) Algorithm

In the multistage decoding algorithm described in section 4.1, an estimate  $\mathbf{b}^{(l)}$  is obtained at stage- $l$  after the inner and the outer decoding have been performed. The information about the estimate is passed to the next decoding stage, in which the level- $(l+1)$

component code  $C^{(l+1)}$  is decoded. This information is used for processing of the received sequence  $\mathbf{r}^{(l)}$  in order to obtain  $\mathbf{r}^{(l+1)}$ .

In List Multistage decoding, the outer code decoder at level- $l$  provides  $W_l$  best estimates,  $\mathbf{b}^{(l),1}, \mathbf{b}^{(l),2}, \dots, \mathbf{b}^{(l),W_l}$ , instead of only one. The information about  $L_l$  best estimates, and the corresponding codewords of  $C_l$  -  $\mathbf{c}^{(l),1}, \mathbf{c}^{(l),2}, \dots, \mathbf{c}^{(l),L_l}$  - is passed to the subsequent decoding stages.

During stage-1 decoding, the outer code decoder  $B^{(1)}$  provides  $W_1$  best estimates,  $\mathbf{b}^{(1),1}, \mathbf{b}^{(1),2}, \dots, \mathbf{b}^{(1),W_1}$ . The information bits corresponding to  $\mathbf{b}^{(1),\omega}$  are stored at the first  $K_1$  positions of the vector  $\text{Info}^\omega$ , for  $1 \leq \omega \leq W_1$ . The corresponding codewords of  $C^{(1)}$  -  $\mathbf{c}^{(1),1}, \mathbf{c}^{(1),2}, \dots, \mathbf{c}^{(1),W_1}$  - are obtained based on (4.1). The  $L_1$  sets  $(\mathbf{c}^{(1),\omega}, \text{Info}^\omega, \mathbf{r}^{(1),\omega} = \mathbf{r}^{(1)})$ , for  $1 \leq \omega \leq W_1$  and  $L_1 = W_1$ , are passed to the stage-2.

At stage- $(l+1)$  of decoding, the following set of steps is performed for each of the  $L_l$  estimates  $\mathbf{c}^{(l),\omega}$  passed from the previous decoding stage:

1. Modified received sequence  $\mathbf{r}^{(l+1),\omega}$  is obtained based on  $\mathbf{r}^{(l),\omega}$  and  $\mathbf{c}^{(l),\omega}$ .
2. Inner code decoding is performed based on  $\mathbf{r}^{(l+1),\omega}$ , and then  $N$  metric tables  $MT_j^{(l+1),\omega}$ ,  $1 \leq j \leq N$ , are passed to the outer code decoder.
3. Outer code decoding is performed using  $N$  metric tables  $MT_j^{(l+1),\omega}$ ,  $1 \leq j \leq N$ , and  $W_{l+1}$  best estimates  $\mathbf{b}_{\text{temp}}^{(l+1),\omega,s}$ ,  $1 \leq s \leq W_{l+1}$  are obtained. For each of them, the

information about estimates from the previous levels is stored in **Info**<sup>ω</sup>. Additional estimates can be obtained using list Viterbi decoding [38] for the outer code  $B^{(l+1)}$ .

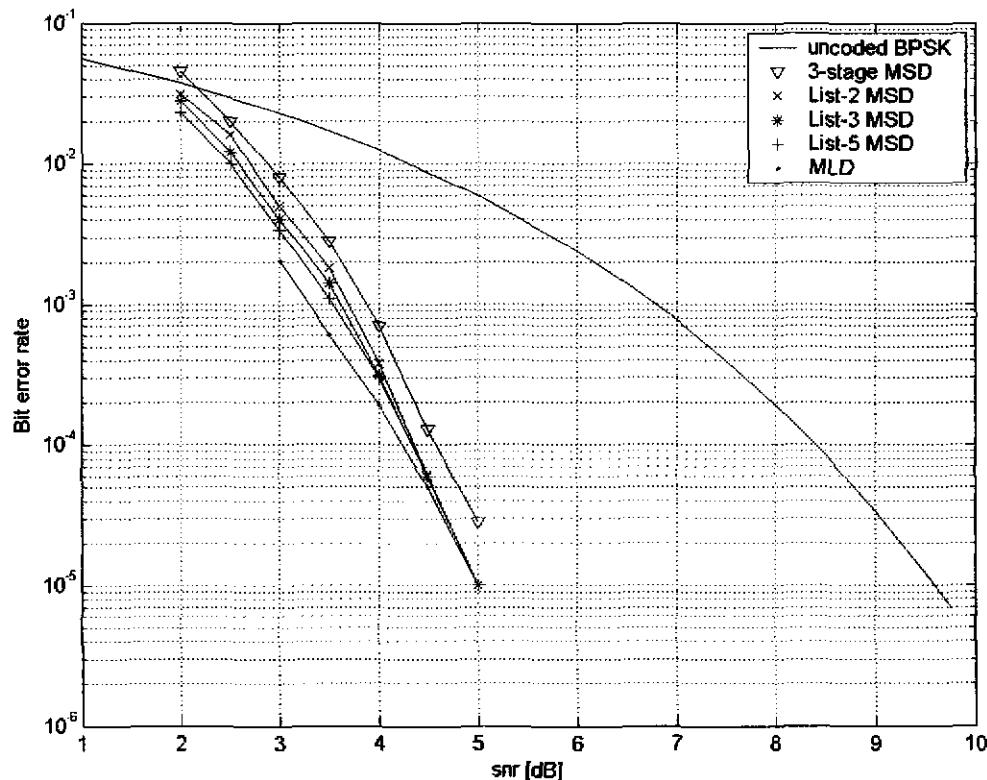
When this procedure is completed for all estimates  $\mathbf{b}^{(l),\omega}$  (or  $\mathbf{c}^{(l),\omega}$ ),  $1 \leq \omega \leq L_l$ , the total number of estimates  $\mathbf{b}_{\text{temp}}^{(l+1),\omega,s}$  at level-( $l+1$ ) is equal to  $L_l \cdot W_{l+1}$ . We choose  $L_{l+1}$  best among these, based on their metrics, and call them  $\mathbf{b}^{(l+1),\omega}$ , for  $1 \leq \omega \leq L_{l+1}$ . At the last decoding stage, only one estimate is found during the outer code decoding for each estimate  $\mathbf{b}^{(m-1),\omega}$  passed from the stage-( $m-1$ ), i.e.,  $W_m = 1$ . The estimate with best metric is chosen as the decoded codeword ( $L_m = 1$ ).

## 4.6 Examples: List Multistage Decoding of RM Codes

### Example 4.6

Consider the (64, 42, 8) code from the Example 4.3 and its two-stage decomposition  $D_3$ . Simulation results for  $W_1 = L_1 \in \{1, 2, 3, 5\}$  (where 1, 2, 3, or 5 decoding estimates are passed from the first to the second decoding stage) and  $W_2 = 1$  are shown in Figure 4.10. Passing additional one estimate from the first to the second decoding stage ( $W_1 = 2$ ), improves the performance by .25 dB at  $P_e = 10^{-3}$ . The difference in performance between list decoding with  $W_1 = 2$  and  $W_1 = 5$  is quite small. In addition, at  $P_e = 10^{-5}$ , the difference becomes indistinct, and both algorithms achieve MLD performance. The optimal performance is achieved with much smaller complexity than that of the Viterbi

algorithm for the single-level code, which achieve ML performance. The choice of  $W_1 = 2$  represents a good trade-off between improved performance and minimally increased decoding complexity.

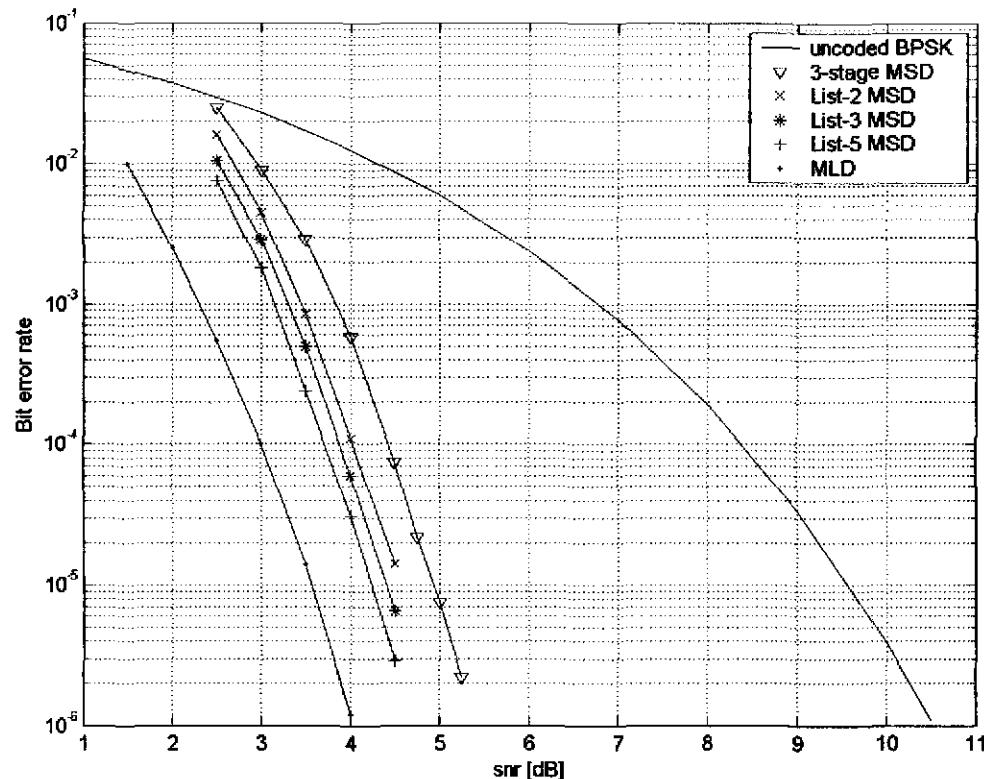


**Figure 4.10 Performance of List MSD of  $(64, 42, 8)$  code**

#### Example 4.7

Consider the  $(128, 64, 16)$  RM code from the Example 4.4 and its three-stage decomposition  $D_7$ . Simulation results for  $W_1 = L_1 = L_2 \in \{1, 2, 3, 4, 5\}$  and  $W_2 = W_3 = 1$  (where 1, 2, 3, 4, or 5 decoding estimates are passed from the first to the second decoding

stage, and from the second to the third decoding stage) are shown in Figure 4.11. As in the previous example, most of the gain in performance is obtained by passing two candidates from the first to the second decoding stage.



**Figure 4.11 Performance of List MSD of (128,64,16) code**

However, a non-trivial improvement is obtained if 5 estimates are passed to the second decoding stage. At  $P_e = 10^{-5}$ , half of the difference between three-stage decoding and MLD is recovered, which is around 0.65 dB.

### Example 4.8

Consider the  $(128,99,16)$  RM code from the Example 4.5 and its three-stage decomposition  $D_8$ . Simulation results for  $W_1 = L_1 = L_2 = 5$  and  $W_2 = 2$  are shown in Figure 4.12. For this code, at  $P_e = 10^{-5}$ , LMSD performs within 0.1 dB of the optimum ML decoding.

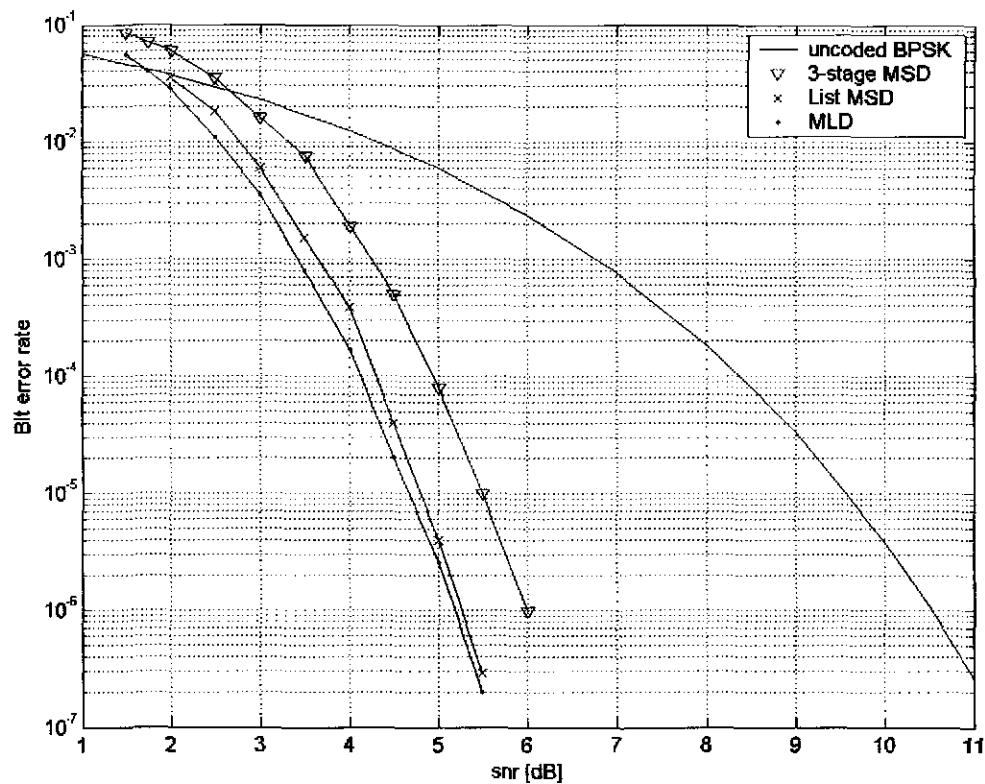


Figure 4.12 Performance of List MSD of  $(128,99,8)$  code

Based on the previous examples, we can conclude that multistage decoding with a list of estimates passed to the next stage represents a good alternative to optimum

decoding of long decomposable codes. The trellis state complexity is the same as for conventional multistage decoding, and, furthermore, decoding for different candidates can be done in parallel and using the same trellis structure for the outer code.

Of course, the computational complexity of the described LMSD algorithm increases multifoldly, compared to MSD. For long decomposable codes, in order to maintain the complexity of each component code in the range that allows practical implementation, the number of decoding stages must be increased. As a result, the size of the list must increase to maintain small performance degradation compared to MLD. This results in a large increase of computational complexity. To overcome this problem, a list of variable size depending on the signal-to-noise ratio (SNR) can be used to reduce the average computational complexity. The idea of a variable-size list is used in developing the new iterative decoding algorithm described in the next chapter.

# **Chapter 5**

## **Iterative Multistage Maximum Likelihood Decoding Algorithm**

We have seen in the previous chapters that multistage decoding (MSD) of long multilevel codes has, for practical signal-to-noise ratios, a significant gap in performance compared to maximum likelihood (ML) decoding. The List MSD, a modification of MSD algorithm in which more than one estimate of a particular decoding stage is used for decoding in consecutive stages, achieves improvement in performance. However, the number of estimates necessary to achieve ML performance can be very large, which contributes to a significant increase in decoding complexity. Much of this added complexity is redundant as most of the time the first estimate of each stage results in the ML solution. Therefore, conditions determining whether the ML solution has been found are needed for computationally efficient decoding.

In this chapter, we describe a novel algorithm for decoding multilevel codes. The basic principle behind the algorithm is “pass additional estimates between decoding stages only until it can be said with certainty that the ML solution has been found”. This can be achieved by passing a list of estimates of variable size to the next decoding stage, iteratively, one by one, with tests determining the last element of the list. It was shown in [17] that in list decoding of single-level concatenated codes with variable list size, the average number of words on the list could be kept small. This was shown to be true for

decoding of the multilevel code as well, namely, the average complexity of the new algorithm is significantly smaller than the complexity of fixed-size list decoding with the comparable error performance.

The proposed algorithm is iterative in nature and has stopping criteria that guarantee optimality of the decoded codeword in maximum likelihood sense. In this chapter, we first described the theory behind the algorithm, that is, the theorems on which the algorithm is based. Then we describe the proposed IMS-MLD algorithm using a two-level concatenated code and prove its optimality. Later, we generalize the algorithm to the case of  $m$ -level codes, where  $m > 2$ .

## 5.1 Review of the conventional MSD

In chapter 4, the conventional multistage decoding algorithm was presented in detail.

Here we review it briefly, and introduce some useful modifications and notation.

Suppose codeword  $\mathbf{c}_t = f_1(\mathbf{b}_{t^{(1)}}) \oplus f_2(\mathbf{b}_{t^{(2)}}) \oplus \dots \oplus f_m(\mathbf{b}_{t^{(m)}})$  of the  $m$ -level concatenated code  $C$  is transmitted. Let  $\mathbf{r} = \mathbf{r}_1 \mathbf{r}_2 \dots \mathbf{r}_N$  be the received sequence sectionalized into  $N$  parts, where component  $\mathbf{r}_j$  consists of  $n$  real numbers,  $\mathbf{r}_j = r_{j,1} r_{j,2} \dots r_{j,n}$ . At each position, received sequence symbol is a sum of the transmitted signal and the additive noise. Assume BPSK signaling with the mapping of codeword bit  $c$  into BPSK signal described by  $c \rightarrow (-1)^c$ . Then  $r_{j,s} = (-1)^{\mathbf{c}_{t^{(j)}}} + \mathcal{N}_{j,s}$ , for  $1 \leq j \leq N$ , and  $1 \leq s \leq n$ , where  $\mathcal{N}_{j,s}$  is the additive noise component at position ' $j,s$ '.

Decoding is performed in  $m$ -stages, one for each level of the multilevel concatenated code  $C$ . Decoding of each stage consists of three-steps: 1) Preprocessing (modifying) of the received sequence, 2) Inner decoding, and 3) Outer decoding.

#### **Preprocessing of received sequence:**

Received sequence is prepared for the decoding at stage- $l$ , by removing the contribution of the estimates obtained in decoding stages 1 through ( $l-1$ ). Let  $\mathbf{c}^{(l)} = f_l(\mathbf{b}^{(l)})$  denote the estimate of the stage- $l$ , where  $\mathbf{b}^{(l)}$  is the estimate of the stage- $l$  outer decoder. Let  $\mathbf{r}^{(l)}$  denote the received sequence used for decoding of stage- $l$ .

1. By definition,  $\mathbf{r}^{(0)} \triangleq \mathbf{r}$ .
2. For  $l \geq 1$ ,  $\mathbf{r}^{(l+1)}$  is obtained from  $\mathbf{r}^{(l)}$  and  $\mathbf{c}^{(l)}$  in the following manner:

$$r_{j,s}^{(l+1)} = r_{j,s}^{(l)} \cdot (-1)^{c_{j,s}^{(l)}} \quad (5.1)$$

where  $r_{j,s}^{(l)}$  denotes the symbol in position ‘ $s$ ’ of the section- $j$  of the received sequence  $\mathbf{r}^{(l)}$ , and  $c_{j,s}^{(l)}$  the bit in position ‘ $s$ ’ of the section- $j$  of the estimate  $\mathbf{c}^{(l)} = \mathbf{c}_1^{(l)}\mathbf{c}_2^{(l)}\dots\mathbf{c}_N^{(l)}$  at stage- $l$ .

#### **Inner decoding of stage- $l$ :**

At stage- $l$  of decoding, the inner decoder processes the  $N$  sections of the received sequence  $\mathbf{r}^{(l)} = (\mathbf{r}_1^{(l)}, \mathbf{r}_2^{(l)}, \dots, \mathbf{r}_N^{(l)})$  independently and forms  $N$  metric tables, one for each section of  $\mathbf{r}^{(l)}$ . Forming the metric table  $MT_j^{(l)}$  at level- $l$  consists of finding  $q_l$  coset

metrics with respect to  $\mathbf{r}_j^{(l)}$ , one for each of the  $q_l$  cosets in  $A^{(l)} / A^{(l+1)}$ . Let  $\beta^{(l)}$  be a symbol of the outer code  $B^{(l)}$  and let  $\{f_i(\beta^{(l)})\}$  denote a coset with  $f_i(\beta^{(l)})$  as the coset representative. Coset metric  $M(\{f_i(\beta^{(l)})\})$  is equal to the metric of the coset winner, as defined in Chapter 4.

Let  $\mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)})$  denote the codeword in  $A^{(l+1)}$  such that

$$M(\mathbf{r}_j^{(l)}, f_i(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)})) = \underset{\mathbf{a}^{(l+1)} \in A^{(l+1)}}{\text{best}} M(\mathbf{r}_j^{(l)}, f_i(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)}). \quad (5.2)$$

Then  $f_i(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)})$  is the coset winner and  $M(f_i(\beta^{(l)}) \oplus \mathbf{a}^{(l+1)*}(\beta^{(l)}, \mathbf{r}_j^{(l)}))$  coset metric of  $\{f_i(\beta^{(l)})\}$ .

Let us expand the metric table  $MT_j^{(l)}$  so that it contains both the coset metric and the coset winner with respect to  $\mathbf{r}_j^{(l)}$  for each coset  $\{f_i(\beta^{(l)})\}$ . This information will be used to perform tests in the new algorithm.

### Outer code decoding of stage- $l$

Let  $\mathbf{b}^{(l)} = (b_1^{(l)}, b_2^{(l)}, \dots, b_N^{(l)})$  be a codeword in the outer code  $B_l$ . The metric of  $\mathbf{b}^{(l)}$  is defined as the following sum:

$$M(\mathbf{b}^{(l)}) = \sum_{j=1}^N M(b_j^{(l)}) = \sum_{j=1}^N M(\{f_i(b_j^{(l)})\}). \quad (5.3)$$

Again, let us introduce a modification of the conventional algorithm to allow the decoder of the outer code at level- $l$  to be a list decoder. The list decoder finds the codeword with the best metric among the codewords in  $B^{(l)}$  that have not yet been declared estimates.

## 5.2 Relations between metrics of estimates in MSD

In this section we derive some useful relations between the metrics of the estimates in different stages of multistage decoding. These results, presented in several lemmas and theorems, are used in the subsequent sections to formulate the new iterative multistage algorithm. Throughout this section, we use the following notation:

- Let  $N$  denote the length of the outer, and  $n$  the length of the inner codes of the  $m$ -level concatenated code  $C$ .
- Let, for  $1 \leq l \leq m$ ,  $\mathbf{c}^{(l)}$  denote a codeword in the component code  $C^{(l)}$
- Let, for  $1 \leq l \leq m$ ,  $\mathbf{r}^{(l)}$  denote the received sequence at stage- $l$ , where  $\mathbf{r}^{(l+1)}$  is obtained from  $\mathbf{r}^{(l)}$  and  $\mathbf{c}^{(l)}$  using equation (5.1).
- Let  $M(\mathbf{c}, \mathbf{r})$  denote either correlation metric or squared Euclidean distance metric of a codeword  $\mathbf{c}$  with respect to the received sequence  $\mathbf{r}$ .

**Lemma 5.1:** Let, for  $1 \leq l \leq m$ ,  $\mathbf{c}^{(l)}$  be sectionalized into  $N$  parts as  $\mathbf{c}^{(l)} = \mathbf{c}_1^{(l)} \mathbf{c}_2^{(l)} \dots \mathbf{c}_N^{(l)}$ .

Then for any codeword  $\mathbf{a} = (a_1 a_2 \dots a_n)$  of length  $n$ ,

$$M(\mathbf{c}_j^{(l)} \oplus \mathbf{a}, \mathbf{r}_j^{(l)}) = M(\mathbf{a}, \mathbf{r}_j^{(l+1)}), \quad \text{for } 1 \leq l < m. \quad (5.4)$$

**Proof:** Let BPSK signal  $t_{j,s} = (-1)^{c_{j,s}}$  correspond to the bit  $c_{j,s}$  of the codeword  $\mathbf{c}$ .

1) Let  $M_{\text{corr}}(\cdot)$  denote correlation metric. Then

$$\begin{aligned} M_{\text{corr}}(\mathbf{c}_j^{(l)} \oplus \mathbf{a}, \mathbf{r}_j^{(l)}) &= \sum_{s=1}^n (-1)^{c_{j,s}^{(l)} \oplus a_s} \cdot r_{j,s}^{(l)} = \sum_{s=1}^n (-1)^{c_{j,s}^{(l)} + a_s} \cdot r_{j,s}^{(l)} \\ &= \sum_{s=1}^n (-1)^{a_s} \cdot \underbrace{(-1)^{c_{j,s}^{(l)}} \cdot r_{j,s}^{(l)}}_{r_{j,s}^{(l+1)}} = \sum_{s=1}^n (-1)^{a_s} \cdot r_{j,s}^{(l+1)} \\ &= M_{\text{corr}}(\mathbf{a}, \mathbf{r}_j^{(l+1)}). \end{aligned}$$

2) Let  $M_E(\cdot)$  denote squared Euclidean distance metric. Then

$$\begin{aligned} M_E(\mathbf{c}_j^{(l)} \oplus \mathbf{a}, \mathbf{r}_j^{(l)}) &= \sum_{s=1}^n ((-1)^{c_{j,s}^{(l)} \oplus a_s} - r_{j,s}^{(l)})^2 \\ &= \sum_{s=1}^n ((-1)^{c_{j,s}^{(l)} \oplus a_s})^2 - 2 \cdot \sum_{s=1}^n ((-1)^{c_{j,s}^{(l)} \oplus a_s} \cdot r_{j,s}^{(l)}) + \|\mathbf{r}_j^{(l)}\|^2 \\ &= n - 2 \cdot M_{\text{corr}}(\mathbf{c}_j^{(l)} \oplus \mathbf{a}, \mathbf{r}_j^{(l)}) + \|\mathbf{r}_j^{(l)}\|^2. \end{aligned}$$

By part 1) of this proof,  $M_{\text{corr}}(\mathbf{c}_j^{(l)} \oplus \mathbf{a}, \mathbf{r}_j^{(l)}) = M_{\text{corr}}(\mathbf{a}, \mathbf{r}_j^{(l+1)})$ . In addition,

$\|\mathbf{r}_j^{(l)}\|^2 = \|\mathbf{r}_j^{(l+1)}\|^2$  since, by (5.1),  $(r_{j,s}^{(l+1)})^2 = (r_{j,s}^{(l)})^2$ . Therefore,

$$\begin{aligned} M_E(\mathbf{c}_j^{(l)} \oplus \mathbf{a}, \mathbf{r}_j^{(l)}) &= n - 2 \cdot M_{\text{corr}}(\mathbf{a}, \mathbf{r}_j^{(l+1)}) + \|\mathbf{r}_j^{(l+1)}\|^2 \\ &= M_E(\mathbf{a}, \mathbf{r}_j^{(l+1)}). \end{aligned} \quad \text{QED}$$

**Theorem 5.2:** Let the codeword  $\mathbf{c}$  in  $m$ -level concatenated code be decomposed into

component codewords as  $\mathbf{c} = \mathbf{c}^{(1)} \oplus \mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}$ . Then,

$$M(\mathbf{c}^{(1)} \oplus \mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(1)}) = M(\mathbf{c}^{(l)} \oplus \mathbf{c}^{(l+1)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(l)}), \quad \text{for } 1 \leq l \leq m$$

and in particular

$$M(\mathbf{c}) = M(\mathbf{c}^{(1)} \oplus \mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(1)}) = M(\mathbf{c}^{(m)}, \mathbf{r}^{(m)}).$$

**Proof:** By substituting  $\mathbf{a} = \mathbf{c}^{(2)} \oplus \mathbf{c}^{(3)} \oplus \dots \oplus \mathbf{c}^{(m)}$  and  $l=1$  into (5.4) of Lemma 5.1, and summing over  $j$ ,  $1 \leq j \leq N$ , we obtain

$$M(\mathbf{c}^{(1)} \oplus \mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(1)}) = M(\mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(2)}). \quad (5.5)$$

Applying the result of Lemma 5.1 to right side of (5.5) with  $\mathbf{a} = \mathbf{c}^{(3)} \oplus \dots \oplus \mathbf{c}^{(m)}$  and  $l=2$ , we obtain

$$M(\mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(2)}) = M(\mathbf{c}^{(3)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(3)}).$$

Substitute  $\mathbf{a} = \mathbf{c}^{(l+1)} \oplus \dots \oplus \mathbf{c}^{(m)}$  into (5.4) of Lemma 5.1 to obtain

$$M(\mathbf{c}^{(l)} \oplus \mathbf{c}^{(l+1)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(l)}) = M(\mathbf{c}^{(l+1)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(l+1)}), \quad \text{for } 1 \leq l \leq m-1. \quad (5.6)$$

By successive application of (5.6) to (5.5) and the results thereof, we obtain

$$M(\mathbf{c}^{(1)} \oplus \mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(1)}) = M(\mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}, \mathbf{r}^{(2)}) = \dots = M(\mathbf{c}^{(m)}, \mathbf{r}^{(m)}). \quad \text{QED}$$

Next, we present a result on optimality of the decoded estimate at stage-1.

**Lemma 5.3:** Let  $\mathbf{b}^{(1)} \in B^{(1)}$  be the codeword decoded during outer decoding of stage-1. If the coset winner sequence  $L(\mathbf{b}^{(1)})$  of  $\mathbf{b}^{(1)}$  is a codeword in the overall concatenated code  $C = \{B^{(1)}, B^{(2)}, \dots, B^{(m)}\} \circ \{A^{(1)}, A^{(2)}, \dots, A^{(m)}\}$ , then it is the codeword in  $C$  with the best metric and hence, the most likely codeword.

**Proof:** Let  $\mathbf{a}_j \in A^{(2)}$ , for  $1 \leq j \leq N$ . Compare the coset winner sequence to the sequences of the form

$$(f_1(b_1^{(1)}) + \mathbf{a}_1, f_1(b_2^{(1)}) + \mathbf{a}_2, \dots, f_1(b_N^{(1)}) + \mathbf{a}_N). \quad (5.7)$$

By definition of the coset winner, and coset winner sequence, the metric of the coset label sequence  $L(\mathbf{b}^{(1)})$  is equal or better than the metric of any sequence described by (5.7).

All the codewords in  $C$  have the form given in (5.7), and in addition satisfy the condition that  $(f_2^{-1}(\mathbf{a}_1), f_2^{-1}(\mathbf{a}_2), \dots, f_2^{-1}(\mathbf{a}_N))$  is a codeword in the outer code  $B^{(2)}$ , thus  $C$  is a subset of all sequences described by (5.7). Then, the metric of  $L(\mathbf{b}^{(1)})$  is also equal or better than the metric of any codeword in  $C$ . Finally, if  $L(\mathbf{b}^{(1)})$  is a codeword in  $C$ , then it is the codeword with best metric, i.e., the most likely codeword. QED

Let us now consider the relation between the metric of the decoded estimate  $\mathbf{c} \in C$ , and the metric of  $\mathbf{b}^{(m)}$ , the estimate of the stage- $m$  outer decoder,

**Lemma 5.4:** Let  $\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)}$  be the estimates produced by the outer decoder at stages

1 through  $m$ . Let  $\mathbf{c} = \mathbf{c}^{(1)} \oplus \mathbf{c}^{(2)} \oplus \dots \oplus \mathbf{c}^{(m)}$ , where  $\mathbf{c}^{(l)} = f_l(\mathbf{b}^{(l)})$  for  $1 \leq l \leq m$ , be the codeword in  $C$  based on these estimates. Then

$$M(\mathbf{b}^{(m)}) = M(\mathbf{c}^{(m)}) = M(\mathbf{c}).$$

**Proof.** Metric of the estimate  $\mathbf{b}^{(m)}$  of the outer decoder at the last stage is given as

$$M(\mathbf{b}^{(m)}) = \sum_{j=1}^N M(\{f_m(b_j^{(m)})\}). \quad (5.8)$$

Since at level- $m$  cosets belong to partition  $A^{(m)} / A^{(m+1)} = A^{(m)} / \{\mathbf{0}\}$ , coset  $\{f_m(\mathbf{b}^{(m)})\}$  consists of exactly one element, namely,  $f_m(\mathbf{b}^{(m)})$ . The equation in (5.8) then becomes

$$M(\mathbf{b}^{(m)}) = \sum_{j=1}^N M(f_m(b_j^{(m)}), \mathbf{r}_j^{(m)}). \quad (5.9)$$

For  $1 \leq j \leq N$ ,  $f_m(b_j^{(m)}) = \mathbf{c}_j^{(m)}$ , where  $\mathbf{c}_j^{(m)}$  is a codeword in  $A^{(m)}$ , and  $\mathbf{c}^{(m)} \triangleq (\mathbf{c}_1^{(m)}, \dots, \mathbf{c}_N^{(m)})$  is a codeword in  $C^{(m)} = B^{(m)} \circ A^{(m)}$ . By substituting  $f_m(b_j^{(m)})$  with  $\mathbf{c}_j^{(m)}$  in (5.9), we obtain

$$M(\mathbf{b}^{(m)}) = \sum_{j=1}^N M(\mathbf{c}_j^{(m)}, \mathbf{r}_j^{(m)}) = M(\mathbf{c}^{(m)}, \mathbf{r}^{(m)}).$$

Now by Theorem 5.2,

$$M(\mathbf{b}^{(m)}) = M(\mathbf{c}^{(m)}, \mathbf{r}^{(m)}) = M(\mathbf{c}, \mathbf{r}^{(1)}) = M(\mathbf{c}). \quad \text{QED}$$

Note: In the rest of the chapter, we assume that the codeword metric is correlation metric, for which *best metric* is equivalent to *largest metric*. All the expressions can be derived for a different metric type if  $\geq$  sign is replaced by what is “better than or equal” for the corresponding metric type. For instance, if squared Euclidean distance is used, in which case *best metric* is equivalent to *smallest metric*, then the sign “ $\geq$ ” describing relation between metrics is replaced by “ $\leq$ ”, and “max” by “min”.

**Theorem 5.5:** Let the outer decoder at stage- $I$  compute codeword  $\mathbf{b}^{(I)}$  based on the estimates  $\mathbf{b}^{(1)}$  through  $\mathbf{b}^{(I-1)}$  passed from the previous stages, and let  $\mathbf{c}^{(I)} = f_I(\mathbf{b}^{(I)})$ . Then the metrics of the outer code estimates satisfy the following series of inequalities:

$$M(\mathbf{b}^{(1)}) \geq M(\mathbf{b}^{(2)}) \geq \dots \geq M(\mathbf{b}^{(m)}).$$

If for some  $I$ ,  $1 \leq I < m$ , coset winner sequence  $L(\mathbf{b}^{(I)})$  is a codeword in  $C$ , i.e.,  $L(\mathbf{b}^{(I)}) \in C$ , then

$$M(\mathbf{b}^{(I)}) = M(\mathbf{b}^{(I+1)}) = \dots = M(\mathbf{b}^{(m)}).$$

Proof: Consider the outer code codeword  $\mathbf{b}^{(I)}$  at level- $I$ ,  $1 \leq I < m$ . By Lemma 4.1, the metric of  $\mathbf{b}^{(I)}$  is equal to the metric of its coset winner sequence  $L(\mathbf{b}^{(I)})$ , i.e.,

$$M(\mathbf{b}^{(l)}) = M(L(\mathbf{b}^{(l)})) = \sum_{j=1}^N \max_{\mathbf{a}_j \in A^{(l+1)}} M(f_l(b_j^{(l)}) \oplus \mathbf{a}_j, \mathbf{r}_j^{(l)}). \quad (5.10)$$

Similarly,

$$M(\mathbf{b}^{(l+1)}) = M(L(\mathbf{b}^{(l+1)})) = \sum_{j=1}^N \max_{\mathbf{a}_j \in A^{(l+2)}} M(f_{l+1}(b_j^{(l+1)}) \oplus \mathbf{a}_j, \mathbf{r}_j^{(l+1)}). \quad (5.11)$$

By Lemma 5.1, for  $\mathbf{c}_j^{(l)} = f_l(b_j^{(l)})$ , (5.11) results in

$$M(\mathbf{b}^{(l+1)}) = \sum_{j=1}^N \max_{\mathbf{a}_j \in A^{(l+2)}} M(f_l(b_j^{(l)}) \oplus f_{l+1}(b_j^{(l+1)}) \oplus \mathbf{a}_j, \mathbf{r}_j^{(l)}). \quad (5.12)$$

Since for  $\mathbf{a}_j \in A^{(l+2)}$ ,  $f_{l+1}(b_j^{(l+1)}) \oplus \mathbf{a}_j$  is a codeword in coset  $\{f_{l+1}(b_j^{(l+1)})\}$ , (5.12)

can be rewritten as

$$M(\mathbf{b}^{(l+1)}) = \sum_{j=1}^N \max_{\mathbf{a}_j \in \{f_{l+1}(b_j^{(l+1)})\}} M(f_l(b_j^{(l)}) \oplus \mathbf{a}_j, \mathbf{r}_j^{(l)}). \quad (5.13)$$

We can see from (5.10) and (5.13) that the expressions for  $M(\mathbf{b}^{(l)})$  and

$M(\mathbf{b}^{(l+1)})$  differ only in the set over which the sum  $\sum_{j=1}^N M(f_l(b_j^{(l)}) \oplus \mathbf{a}_j, \mathbf{r}_j^{(l)})$  is

maximized. In case of  $M(\mathbf{b}^{(l)})$ , we maximize over the set of all sequences

$\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N$  such that  $\mathbf{a}_j \in A^{(l+1)}$  for  $1 \leq j \leq N$ , while in case of  $M(\mathbf{b}^{(l+1)})$ , the

maximization is performed over the set all sequences  $\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N$  such that

$\mathbf{a}_j \in \{f_{l+1}(b_j^{(l+1)})\}$ . Since the latter set is a subset of the former, i.e.,

$\{f_{l+1}(b_j^{(l+1)})\} \subset A^{(l+1)}$ , we have

$$\max_{\mathbf{a}_j \in A^{(l+1)}} M(f_l(b_j^{(l)}) \oplus \mathbf{a}_j, \mathbf{r}_j^{(l)}) \geq \max_{\mathbf{a}_j \in \{f_{l+1}(b_j^{(l+1)})\}} M(f_l(b_j^{(l)}) \oplus \mathbf{a}_j, \mathbf{r}_j^{(l)})$$

and therefore  $M(\mathbf{b}^{(l)}) \geq M(\mathbf{b}^{(l+1)})$  for  $1 \leq l \leq m-1$ .

Substituting  $l=1,2,\dots,m-1$ , we obtain

$$M(\mathbf{b}^{(1)}) \geq M(\mathbf{b}^{(2)}) \geq \dots \geq M(\mathbf{b}^{(m-1)}) \geq M(\mathbf{b}^{(m)}).$$

If the coset winner sequence  $L(\mathbf{b}^{(l)})$  is a codeword in  $C$ , then it can be uniquely represented as a sum of codewords of component codes  $l$  through  $m$ , i.e.

$$L(\mathbf{b}^{(l)}) = \bar{\mathbf{c}}^{(l)} \oplus \bar{\mathbf{c}}^{(l+1)} \oplus \dots \oplus \bar{\mathbf{c}}^{(m)}.$$

Also, by definition of coset label sequence,

$$L(\mathbf{b}^{(l)}) = (f_l(b_1^{(l)}) \oplus \mathbf{a}_1^{(l+1)*}, f_l(b_2^{(l)}) \oplus \mathbf{a}_2^{(l+1)*}, \dots, f_l(b_N^{(l)}) \oplus \mathbf{a}_N^{(l+1)*}), \text{ so}$$

$$\mathbf{c}^{(l)} = f_l(b^{(l)}) = \bar{\mathbf{c}}^{(l)}, \quad \text{and}$$

$$(\mathbf{a}_1^{(l+1)*}, \mathbf{a}_2^{(l+1)*}, \dots, \mathbf{a}_N^{(l+1)*}) = \bar{\mathbf{c}}^{(l+1)} \oplus \bar{\mathbf{c}}^{(l+2)} \oplus \dots \oplus \bar{\mathbf{c}}^{(m)}. \quad (5.14)$$

Further decoding will result in codewords  $\mathbf{c}^{(l+1)}, \mathbf{c}^{(l+2)}, \dots, \mathbf{c}^{(m)}$ , so

$$\mathbf{c}^{(i)} = \bar{\mathbf{c}}^{(i)} \text{ for } l+1 \leq i \leq m.$$

Next, the metric of  $L(\mathbf{b}^{(l)})$  can be written as

$$\begin{aligned} M(L(\mathbf{b}^{(l)})) &= M((f_l(\mathbf{b}^{(l)}) \oplus \mathbf{a}_1^{(l+1)*} \mathbf{a}_2^{(l+1)*} \dots \mathbf{a}_N^{(l+1)*}), \mathbf{r}^{(l)}) \\ &= M(\mathbf{a}_1^{(l+1)*} \mathbf{a}_2^{(l+1)*} \dots \mathbf{a}_N^{(l+1)*}, \mathbf{r}^{(l+1)}) \quad \text{by Lemma 5.1} \\ &= M(\bar{\mathbf{c}}^{(l+1)} \oplus \bar{\mathbf{c}}^{(l+2)} \oplus \dots \oplus \bar{\mathbf{c}}^{(m)}, \mathbf{r}^{(l+1)}) \quad \text{by (5.14)} \\ &= M(\bar{\mathbf{c}}^{(m)}, \mathbf{r}^{(m)}) \quad \text{by Theorem 5.2} \\ &= M(\mathbf{b}^{(m)}) \quad \text{by Lemma 5.4} \end{aligned}$$

Thus, since  $M(\mathbf{b}^{(l)}) = M(L(\mathbf{b}^{(l)}))$  (Lemma 4.1), we obtain from the previous series of inequalities

$$M(\mathbf{b}^{(l)}) = M(\mathbf{b}^{(m)}).$$

QED

## 5.3 IMS-MLD for two-level concatenated codes

### 5.3.1 Theorems for the two-stage iterative decoding algorithm

For the iterative two-stage MLD algorithm, decoding iterations are based on the generation of a sequence of estimates at the first stage. The estimates are generated one at a time, in reverse order of their metrics, starting from the largest (best). This can be achieved using a serial list decoder for the outer code  $B^{(1)}$ , for instance serial list Viterbi algorithm [38]. Based on the  $N$  metric tables provided by the inner code decoder, the outer code decoder finds the codeword with the best metric among those that have not been chosen so far.

Let  $i$  denote the present iteration. At the  $i^{\text{th}}$  iteration, the first-stage outer decoder generates the  $i^{\text{th}}$  best estimate, denoted by  $\mathbf{b}^{(1),i} = (b_1^{(1),i}, b_2^{(1),i}, \dots, b_N^{(1),i})$ . Received sequence  $\mathbf{r}$  is then modified based on  $\mathbf{c}^{(1),i} = f_1(\mathbf{b}^{(1),i})$  to obtain  $\mathbf{r}^{(2),i}$ . Let  $\mathbf{b}^{(2),i}$  denote the decoded codeword at the second stage based on  $\mathbf{r}^{(2),i}$ . The overall estimate at iteration  $i$  is  $\mathbf{c}^i = f_1(\mathbf{b}^{(1),i}) \oplus f_2(\mathbf{b}^{(2),i})$ , with metric  $M(\mathbf{c}^i)$ , which is by Lemma 5.4 equal to  $M(\mathbf{b}^{(2),i})$ .

Let  $i_0$  be the integer such that  $1 \leq i_0 < i$  and

$$M(\mathbf{c}^{i_0}) = \max_{1 \leq j < i} M(\mathbf{c}^j) \quad (5.15)$$

that is,  $\mathbf{c}^{i_0}$  is the best decoded codeword in  $C$  during the first  $i-1$  iterations, and it is found in iteration  $i_0$ .

The proposed algorithm is based on the following two theorems.

**Theorem 5.6:** For iteration  $i \geq 1$ , metrics of the estimates  $\mathbf{b}^{(1),i}$  and  $\mathbf{b}^{(2),i}$  satisfy relation

$$M(\mathbf{b}^{(1),i}) \geq M(\mathbf{b}^{(2),i}), \text{ where equality holds if the coset winner sequence}$$

for  $\mathbf{b}^{(1),i}$ ,  $L(\mathbf{b}^{(1),i}) \triangleq (f_1(b_1^{(1),i}) \oplus \mathbf{a}_1^*, f_1(b_2^{(1),i}) \oplus \mathbf{a}_2^*, \dots, f_1(b_N^{(1),i}) \oplus \mathbf{a}_N^*)$ , is a

codeword in  $C$ , i.e., if  $(\mathbf{a}_1^*, \mathbf{a}_2^*, \dots, \mathbf{a}_N^*)$  is a codeword in  $C^{(2)} = B^{(2)} \circ A^{(2)}$ .

Proof: Theorem 5.6 follows directly from Theorem 5.5 for  $m=2$ .

**Theorem 5.7:** Given the notation of this section, the following implications are true:

- 1) If  $M(\mathbf{c}^{i_0}) \geq M(\mathbf{b}^{(1),i})$ , then  $\mathbf{c}^{i_0}$  is the most likely codeword with respect to the received sequence  $\mathbf{r}$ .
- 2) If  $M(\mathbf{c}^{i_0}) < M(\mathbf{b}^{(1),i})$  and the coset winner sequence  $L(\mathbf{b}^{(1),i})$  is a codeword in  $C$ , then  $L(\mathbf{b}^{(1),i})$  is the most likely codeword in  $C$  with respect to  $\mathbf{r}$ .

Proof:

- 1) Assume  $M(\mathbf{c}^{i_0}) \geq M(\mathbf{b}^{(1),i})$ . It is a property of list decoding that

$$\text{for } i \leq j \quad M(\mathbf{b}^{(1),i}) \geq M(\mathbf{b}^{(1),j}). \quad (5.16)$$

Then for all iterations  $j \geq i$ , by the assumption, (5.16), Theorem 5.5 and Lemma 5.4 for  $m=2$ ,

$$M(\mathbf{c}^{i_0}) \geq M(\mathbf{b}^{(1),i}) \geq M(\mathbf{b}^{(1),j}) \geq M(\mathbf{b}^{(2),j}) = M(\mathbf{c}^j), \text{ i.e.,}$$

$$\text{for } j \geq i, \quad M(\mathbf{c}^i) \geq M(\mathbf{c}^j) \quad (5.17)$$

that is,  $\mathbf{c}^i$  is the best estimate in iterations  $j$ ,  $j \geq i$ . Also, by (5.15),

$$\text{for } j < i, \quad M(\mathbf{c}^i) \geq M(\mathbf{c}^j), \quad (5.18)$$

that is,  $\mathbf{c}^i$  is the best estimate in iterations  $j$ ,  $1 \leq j \leq i-1$ . By (5.17) and (5.18),  $\mathbf{c}^i$  is the most likely codeword in  $C$ .

- 2) By assumption,  $M(\mathbf{c}^i) < M(\mathbf{b}^{(1)i})$  and  $L(\mathbf{b}^{(1)i})$  be a codeword in  $C$ . Then it follows from (5.15) that

$$\text{for } j < i, \quad M(\mathbf{b}^{(1)i}) > M(\mathbf{c}^i) \geq M(\mathbf{c}^j) \quad (5.19)$$

By (5.16), Theorem 5.5 and Lemma 5.4 for  $m=2$ , it follows that

$$\text{for } j \geq i, \quad M(\mathbf{b}^{(1)i}) \geq M(\mathbf{b}^{(1),j}) \geq M(\mathbf{b}^{(2),j}) = M(\mathbf{c}^j) \quad (5.20)$$

Thus by Lemma 4.1, (5.19) and (5.20),  $M(L(\mathbf{b}^{(1)i})) \geq M(\mathbf{c}^j)$  for all iterations  $j$ .

By assumption  $L(\mathbf{b}^{(1)i})$  is a codeword in  $C$ , thus  $L(\mathbf{b}^{(1)i})$  is the most likely codeword in  $C$ . QED

### 5.3.2 The Algorithm for two-stage decoding

Estimates are generated iteratively. Each iteration  $i$  starts by generating an estimate  $\mathbf{b}^{(1)i}$  at the first decoding stage. Estimates are generated until one of the criteria stated in

Theorem 5.7 is satisfied. Let  $i$  denote the current iteration number and  $i_0$  the iteration in which currently best estimate  $\mathbf{c}^{i_0}$  was computed. The algorithm consists of five steps:

**Step 1:** Set  $i=1$ . Compute the first (best) estimate  $\mathbf{b}^{(0),1}$  of the first decoding stage, and its metric  $M(\mathbf{b}^{(0),1})$ . Check whether the coset winner sequence  $L(\mathbf{b}^{(0),1})$  is a codeword in  $C$ . If it is,  $L(\mathbf{b}^{(0),1})$  is the most likely codeword and the decoding stops. Otherwise, go to Step 2.

**Step 2:** Perform second stage decoding and obtain the estimate  $\mathbf{b}^{(2),1}$  with metric  $M(\mathbf{b}^{(2),1})$ . Let  $i_0=1$ , and store  $\mathbf{c}^1 = f_1(\mathbf{b}^{(0),1}) \oplus f_2(\mathbf{b}^{(2),1})$  and the metric  $M(\mathbf{c}^1) = M(\mathbf{b}^{(2),1})$  into buffer registers for  $\mathbf{c}^{i_0}$  and  $M(\mathbf{c}^{i_0})$ . Go to Step 3.

**Step 3:** For iteration  $i \geq 1$ , the best estimate found so far,  $\mathbf{c}^{i_0}$ , is currently stored in a buffer register together with its metric  $M(\mathbf{c}^{i_0})$ . Increment the iteration number  $i$  by one. Determine the  $i^{\text{th}}$  best estimate  $\mathbf{b}^{(1),i}$  of the outer code  $B^{(1)}$ , and its metric  $M(\mathbf{b}^{(1),i})$ . If  $M(\mathbf{c}^{i_0}) \geq M(\mathbf{b}^{(1),i})$ , then  $\mathbf{c}^{i_0}$  is the most likely code in  $C$ , and decoding is finished. Otherwise, go to Step 4.

**Step 4:** Check if the coset winner sequence  $L(\mathbf{b}^{(1),i})$  is a codeword in  $C$ . If it is,  $L(\mathbf{b}^{(1),i})$  is the most likely codeword in  $C$  and decoding is finished. Otherwise, go to step 5.

**Step 5:** Generate  $\mathbf{b}^{(2),i}$ . Now  $\mathbf{c}^i = f_1(\mathbf{b}^{(1),i}) \oplus f_2(\mathbf{b}^{(2),i})$  and  $M(\mathbf{c}^i) = M(\mathbf{b}^{(2),i})$ . If  $M(\mathbf{c}^i) > M(\mathbf{c}^i)$ , set  $i_0 = i$  and store  $\mathbf{c}^i$  and  $M(\mathbf{c}^i)$  into buffers for  $\mathbf{c}^i$  and  $M(\mathbf{c}^i)$ . If maximum iteration number has been reached, stop. Otherwise, go to Step 3.

The decoding process iterates until the most likely codeword is found, or all estimates have been generated. The upper bound on the number of iterations is equal to the number of codewords in  $B^{(1)}$ , i.e.,  $2^{K_1}$ . This is the extreme case. In general, the number of iterations required to obtain the most likely codeword is very small compared to  $2^{K_1}$ .

## 5.4 IMS-MLD for $m$ -level concatenated codes

### 5.4.1 Theorems for $m$ -stage iterative decoding algorithm

Two-stage iterative MLD decoding can be generalized to the case of  $m$ -level concatenated codes. The algorithm is based on the following two theorems.

**Theorem 5.8:** Let  $l_1$  and  $l_2$ ,  $1 \leq l_1 < l_2 \leq m$ , denote two levels in the multilevel structure of code  $C$ . Then for all iterations  $i$ ,  $i \geq 1$ , metrics of the estimates in levels  $l_1$  and  $l_2$  in the same iteration satisfy the relation  $M(\mathbf{b}^{(l_1),i}) \geq M(\mathbf{b}^{(l_2),i})$ . If the coset winner sequence for the estimate  $\mathbf{b}^{(l_1),i}$ ,  $L(\mathbf{b}^{(l_1),i})$ , is a codeword in  $C$ , then  $M(\mathbf{b}^{(l_1),i}) = M(\mathbf{b}^{(l_2),i})$ , and moreover,  $M(\mathbf{b}^{(l_1),i}) = M(\mathbf{b}^{(l),i})$  for all  $l_1 \leq l \leq m$ .

**Theorem 5.9:** Let  $i$  denote the current iteration. Let  $i_0$ ,  $1 \leq i_0 \leq i-1$ , be the iteration in which the currently best codeword  $\mathbf{c}^{i_0}$  was computed.

- 1) For  $i \geq i_0 \geq 1$ , if  $M(\mathbf{c}^{i_0}) \geq M(\mathbf{b}^{(l),i})$ , then the codeword in  $C$  obtained during iteration  $i_0$  is the most likely codeword with respect to the received sequence  $\mathbf{r}$ .
- 2) If  $M(\mathbf{c}^{i_0}) \geq M(\mathbf{b}^{(l),i})$  for some  $l$ ,  $1 < l < m$ , then the codeword obtained at iteration  $i_0$  is better than any codeword that results from further list decoding at level- $l$ .
- 3) If  $M(\mathbf{c}^{i_0}) < M(\mathbf{b}^{(l),i})$  and the coset winner sequence  $L(\mathbf{b}^{(l),i})$  is a codeword in  $C$ , then  $L(\mathbf{b}^{(l),i})$  is more likely codeword with respect to  $\mathbf{r}$  than the codeword obtained during iteration  $i_0$ . Therefore  $L(\mathbf{b}^{(l),i})$  becomes currently best codeword.
- 4) If  $M(\mathbf{b}^{(m),i_0}) < M(\mathbf{b}^{(l),i})$  and the coset winner sequence  $L(\mathbf{b}^{(l),i})$  is a codeword in  $C$ , then  $L(\mathbf{b}^{(l),i})$  is the most likely codeword with respect to  $\mathbf{r}$ .

Proofs of the theorems 5.8 and 5.9 are based on Theorem 5.5 and the property of list decoding. These theorems are used to design the algorithm for decoding  $m$ -level concatenated code.

#### 5.4.2 The Algorithm for $m$ -stage decoding

In  $m$ -stage, a new decoding iteration can be initiated at any stage above the final stage. The decoding iteration begins with the generation of a new estimate at the starting stage, say stage- $l$ . If all the  $2^{K_l}$  estimates at stage- $l$  (resulting from a particular sequence of codewords from stages above the stage- $l$ ) have already been generated and tested, the

decoder moves up to the stage- $(l-1)$  and starts a new iteration with a new estimate. Decoding iterations continue until the ML codeword is found. Just like for the two-stage decoding, the final decoding decision is made at the first stage.

Suppose the decoding process is at the stage- $l$  of iteration  $i$ . Let  $\mathbf{b}^{(l),i}$  denote the decoded codeword in the outer code  $B^{(l)}$  during iteration  $i$ . Let  $L(\mathbf{b}^{(l),i})$  denote the coset winner sequence corresponding to  $\mathbf{b}^{(l),i}$ . The metric of  $\mathbf{b}^{(l),i}$  is denoted by  $M(\mathbf{b}^{(l),i})$ . Let  $\mathbf{c}^{i_0}$  denote currently best decoded codeword with metric  $M(\mathbf{c}^{i_0})$ . Then the buffer contains  $\mathbf{c}^{i_0}$  and  $M(\mathbf{c}^{i_0}) = M(\mathbf{b}^{(m),i_0})$ .

At the completion of the  $i^{\text{th}}$  decoding stage of  $j^{\text{th}}$  iteration, the decoder makes one of the following three moves:

1. If  $M(\mathbf{b}^{(i),j}) \leq M(\mathbf{b}^{(i_0),j_0})$  the decoder moves up to the stage  $(i-1)$  and starts a new iteration with a new estimate.
2. Otherwise, if  $M(\mathbf{b}^{(i),j}) > M(\mathbf{b}^{(i_0),j_0})$ , the coset winner sequence  $L(\mathbf{b}^{(i),j})$  is tested. If it is a codeword in  $C$ , then the buffer is updated ( $M(\mathbf{b}^{(i_0),j_0}) = M(\mathbf{b}^{(i),j})$ , etc.) The decoder moves up to the stage- $(i-1)$  and starts a new iteration with a new estimate.
3. If  $M(\mathbf{b}^{(i),j}) > M(\mathbf{b}^{(i_0),j_0})$ , and the coset label sequence  $L(\mathbf{b}^{(i),j})$  is not a codeword in  $C$ , then decoder moves down to the  $(i+1)^{\text{th}}$  stage of the  $j^{\text{th}}$  iteration.

When the decoder reaches the last ( $m^{\text{th}}$ ) stage, it must move up to the  $(m-1)^{\text{th}}$  stage and start a new iteration (since the coset winner sequence of the estimate  $\mathbf{b}^{(m),j}$  is always a codeword in  $C$ ).

Whenever the decoder reaches the first stage at the beginning of an iteration, a decision is made at the completion of the first-stage decoding whether the decoding is terminated or continues. Suppose the decoder has reached and completed the first stage decoding at the  $j^{\text{th}}$  iteration. The decoder makes one the following moves:

1. If  $M(\mathbf{b}^{(i),j}) \leq M(\mathbf{b}^{(i_0),j_0})$ , then the decoding is finished. The ML codeword is formed from  $\mathbf{b}^{(i_0),j_0}$  and the codewords above  $i_0^{\text{th}}$  stage which resulted in the generation of  $\mathbf{b}^{(i_0),j_0}$ .
2. Otherwise, if  $M(\mathbf{b}^{(i),j}) > M(\mathbf{b}^{(i_0),j_0})$ , and the coset winner sequence  $L(\mathbf{b}^{(i),j})$  is a codeword in  $C$ , then  $L(\mathbf{b}^{(i),j})$  is the ML codeword. Decoding stops.
3. If  $M(\mathbf{b}^{(i),j}) > M(\mathbf{b}^{(i_0),j_0})$  and  $L(\mathbf{b}^{(i),j})$  is not a codeword in  $C$ , then the decoder moves down to the second stage and continues the decoding for the  $j^{\text{th}}$  iteration.

Thus, the tests performed at the first decoding stage are actually the optimality conditions.

## **5.5 Suboptimum versions**

The worst-case complexity for the proposed iterative algorithm is a problem that needs to be addressed. As with other variable size list decodings [17], the small average complexity does not guarantee that the worst-case complexity will be small. If all estimates are processed at the same time, in parallel, then large buffers are needed. However, since we use the serial list decoding, only one new estimate is generated at a time. Thus the only large buffer needed is that for the trellis of the list VA.

In this section we explore the possibility of keeping the decoding complexity low. This will result in suboptimum versions of the algorithm. The tradeoff between the complexity and performance is expected to be good.

### **5.5.1 Limited list size decoding**

It was suggested in [39], that the problem of large average number of iterations, and therefore the worst-case complexity and delay, can be overcome by setting a limit on the number of estimates generated by the list decoder of the outer codes. We introduce this restriction to the iterative algorithm. In this case, the decoding algorithm achieves near optimal error performance.

### **5.5.2 Threshold decoding**

Previous work [16,17] suggests that good results are obtained when the criterion of adding a codeword to the list is based on a heuristically determined threshold value. Here,

we describe a suboptimum version of the iterative algorithm presented in section 5.4, in which comparison to threshold is used as a stopping criterion for decoding [43]. The modifications of the original algorithm for the three-level code are as follows:

1. If the difference between the metric of the estimate after the first decoding stage, and the metric of the corresponding estimate after the last (here third) stage during iteration  $j$  is bigger than a given threshold  $\theta$ , i.e.,

$$M(\mathbf{b}^{(3),j}) - M(\mathbf{b}^{(0),j}) > \theta$$

then we proceed with the iteration  $j+1$  from the first stage, i.e., with the generation of a new estimate at the first level. The motivation for this heuristic rule is based on the belief that if a difference between the metric in the first stage and the metric of the corresponding estimate after the last stage is too large, it is indication that the coset sequence of the first stage does not include the most likely codeword.

2. If  $M(\mathbf{b}^{(3),j}) - M(\mathbf{b}^{(2),j}) < \delta$ , where  $\delta$  is a small value threshold, then we proceed with the iteration  $j+1$  from the first stage. This modification is based on the assumption that if the condition is satisfied, with high probability the best codeword that belongs to the coset sequence of  $\mathbf{b}^{(1),j}$  has been found.
3. The third modification is the addition of a stopping criterion. If the constellation winner sequence at the second stage is a codeword in  $C$ , then

we stop the iterations. Note that when this condition is tested in the first decoding stage, it is an optimality criterion, while in stages after first it is a suboptimum stopping criterion.

In this chapter, we presented the new iterative algorithm for multilevel codes, which achieves ML performance. Some of the possible suboptimum versions are discussed as well. The applications and the discussion of performance are given in the next chapter.

# **Chapter 6**

## **Applications of IMS-MLD Algorithm**

The algorithm derived in the previous chapter is tested for two classes of multilevel concatenated codes: decomposed *RM* codes and Block Coded Modulation (BCM) codes. The results are compared to other algorithms that can be applied for decoding of the multilevel codes in question.

### **6.1 Application of IMS-MLD to Decomposable RM codes**

In this section, we give two examples in which IMS-MLD is applied to two RM codes of length 128. We assume BPSK transmission over additive white Gaussian noise (AWGN) channel. We also use encoding in reduced echelon form in all simulated cases to minimize the BER [40]

#### **Example 6.1**

Consider the third order RM code of length 128, which is a (128, 64, 16) code. This code can be decomposed as multilevel concatenated code [9], and a possible decomposition is given by:

$$(128, 64, 16) = \{(16, 1)(16, 5)^2, (16, 5)(16, 11), (16, 11)^2(16, 15)\} \circ \{(8, 8), (8, 5), (8, 3)\}.$$

The outer codes are interleaved RM codes of length 16, while the inner codes are formed by a partition chain,  $(8,8)/(8,5)/(8,3)$ . The universal code is partitioned by the two subcodes of RM codes, namely  $(8,5) \subset (8,7)$  and  $(8,3) \subset (8,4)$ .

The 16-section full code trellis has  $2^{26}$  states [3]. However, after decomposition, the 8-section trellises of the component outer codes have  $2^9$ ,  $2^8$ , and  $2^9$  states, respectively. Therefore, for MSD, the total trellis state complexity is equal to  $2^9 + 2^8 + 2^9$ , which is much less than  $2^{26}$ .

The bit error performances for different algorithms are given in Figure 6.1. It can be seen that the performance curve of the IMS-MLD algorithm agrees with the union bound for the  $(128,64)$  RM code. The new algorithm outperforms the conventional MSD by  $1.35$  dB at  $\text{BER} = 2 \times 10^{-6}$ .

The computational complexity is expressed in terms of number of real operations (addition and comparisons) required for decoding one block of data. The computational complexity of Viterbi algorithm based on the full code trellis is  $9.3 \times 10^9$ . For conventional MSD, the computational complexity is  $4.26 \times 10^4$ . Due to the iterative nature of the IMS-MLD algorithm, the average complexity varies with the SNR. The values of the average number of real operation per block, and average number of estimates generated at each decoding level ( $\text{ave}_l$ , for  $l = 1, 2, 3$ ) are given in Table 6.1. All the values decrease as the SNR increases. At a certain point, the complexity of the IMS-MLD becomes even smaller than that of conventional MSD.

Another way to compare is to include a test on the coset winner sequence for the conventional MSD. If this sequence is a codeword at any level, the decoding of the following stages is not necessary. The complexity of this modified conventional MSD becomes variable with SNR, and ranges from  $2.6 \times 10^4$  for SNR=2.0 dB, to  $0.55 \times 10^4$  for SNR=5.0 dB. Complexities of various algorithms are given in Table 6.3 for comparison.

The problem of large average number of estimates generated at each stage, as well as the large worst-case complexity, can be overcome by using the Limited list size suboptimum version discussed in section 5.3. Setting this limits of number of estimates generated by the outer code decoders to 5 and 2 for the first and second outer decoder respectively, results in an iterative algorithm with the same performance as that of List MSD [12] with parameters 5 and 2. However, the computational complexity of the suboptimum iterative algorithm is much lower, as it can be seen from Table 6.2 (complexity of the list decoding is equal to the upper bound on the complexity of suboptimum IMS algorithm). This suboptimum version of IMS not only has large reduction in the average number of estimates generated at each stage for small SNR, but it has bounded worst case complexity, as well. Complexities of different algorithms are given in Table 6.3 for comparison.

From Figure 6.1 and Table 6.3, it can be seen that, compared to conventional MSD, IMS-MLD achieves significantly better performance with relatively small increase in average computational complexity. In Figure 6.2 and Table 6.3, the list decoding of [12] is also included. It can be seen that IMS-MLD outperforms the List MSD by 0.4 dB.

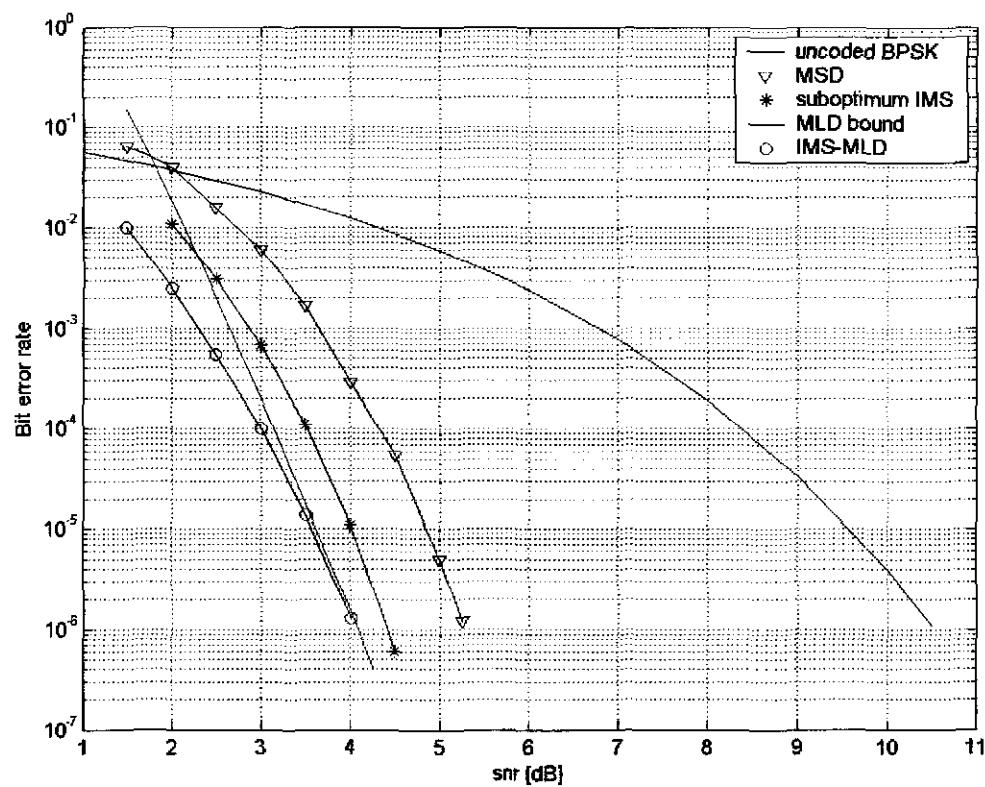
When compared to Viterbi decoding algorithm based on the full code trellis, IMS-MLD achieves the same performance with enormous reduction in both computational and trellis complexity.

**Table 6.1 Decoding Complexity for IMS-MLD of RM (3,7) = (128,64,16)**

<b>SNR [dB]</b>	<b>2.0</b>	<b>2.5</b>	<b>3.0</b>	<b>3.5</b>	<b>4.0</b>
<b>No of opr [10<sup>4</sup>]</b>	45.0	12.6	10.0	4.0	2.6
ave <sub>1</sub>	31	9.12	3.2	1.84	1.47
ave <sub>2</sub>	40	11	3	1.1	0.69
ave <sub>3</sub>	9.5	2.8	0.78	0.22	0.081

**Table 6.2 Decoding Complexity for Suboptimum IMS of RM (3,7)=(128,64,16)**

<b>SNR [dB]</b>	<b>2.0</b>	<b>2.5</b>	<b>3.0</b>	<b>3.5</b>	<b>4.0</b>
<b>No of opr [10<sup>4</sup>]</b>	6.25	3.3	1.82	1.15	0.83
ave <sub>1</sub>	3.24	2.55	2.04	1.71	1.46
ave <sub>2</sub>	3.5	2.20	1.36	0.92	0.66
ave <sub>3</sub>	1.41	0.67	0.30	0.14	0.07



**Figure 6.1 Performance of (128,64,16) over AWGN channel**

**Table 6.3 Computational Complexity of various algorithms for (128,64,16) code**

Decoding Algorithm	Average number of real operations at SNR [dB]							Upper bound on complexity
	2.0	2.5	3.0	3.5	4.0	4.5	5.0	
Conventional MSD [ $10^4$ ]	2.57	1.88	1.42	1.03	0.80	0.65	5.0	4.26
Suboptimum IMS [ $10^4$ ]	6.25	3.33	1.82	1.15	0.83	-	-	36.6
Optimum IMS-MLD [ $10^4$ ]	45.0	12.6	10.0	4.0	2.6	-	-	-
Viterbi [ $10^9$ ]	9.3				9.29	9.28	9.23	9.3

### **Example 6.2**

Consider the 4<sup>th</sup> order (128,99) RM code with the following decomposition

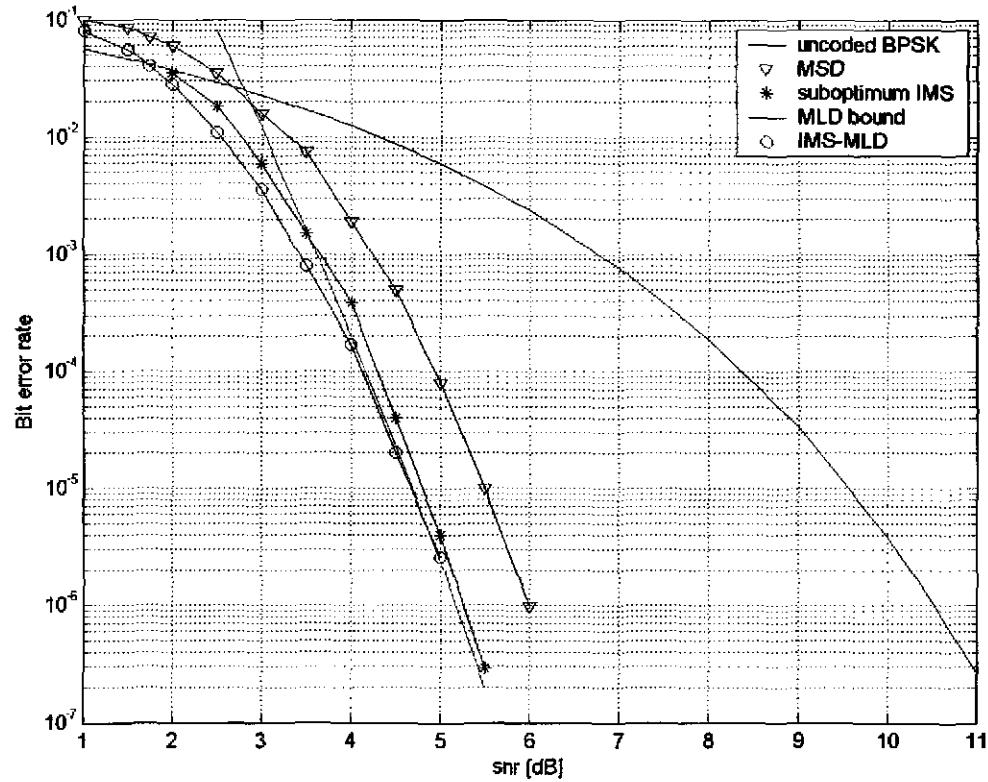
$$(128,99,8) = \{(16,5)(16,11), (16,11)^2, (16,15)^3(16,16)\} \circ \{(8,8), (8,6), (8,4)\}.$$

Again, the outer codes are obtained by interleaving the RM codes of length 16. The trellis state complexities of the three outer codes  $B^{(1)}$ ,  $B^{(2)}$ , and  $B^{(3)}$  are  $2^8$ ,  $2^8$ , and  $2^4$  respectively. These complexities are very small compared to  $2^{19}$  states of the full code trellis.

The IMS-MLD outperforms the conventional MSD by 0.75 dB, as seen in figure 6.2. If suboptimum IMS algorithm is used with the list sizes of the list Viterbi algorithm limited to 5 for the first stage outer code and 2 for the second stage outer code, it achieves almost optimum performance with a very small average complexity and with bounded worst-case complexity. Table 6.4 contains complexities of different decoding algorithms.

**Table 6.4 Computational Complexity of various algorithms for (128,99,8) code**

Decoding Algorithm	Average number of real operations at SNR [dB]:							Upper bound on complexity
	2.5	3.0	3.5	4.0	4.5	5.0	5.5	
Conventional MSD [ $10^4$ ]	1.2	1.1	0.9	0.7	0.56	0.4	0.3	1.5
Suboptimum IMS [ $10^4$ ]	3.3	2.1	1.3	0.8	0.56	0.4	-	7.8
Optimum IMS-MLD [ $10^4$ ]	22.0	7.9	6.3	1.0	0.56	0.4	-	-
Viterbi [ $10^9$ ]	4.62	4.61	4.56	4.44	4.20	3.82	3.29	4.63



**Figure 6.2 Performance of (128,99,8) over AWGN channel**

## 6.2 Application of IMS-MLD to BCM codes

In [16], an improved MSD algorithm for decoding multilevel codes based on list decoding of the outer codes was presented. The improvement is achieved by passing additional  $L$  estimates from the first to the second decoding stage if the distance measure between the decoded and the received sequence is larger than a given threshold. The results were given only for  $L = 2$  and short codes.

In this section, we investigate the application of the IMS-MLD algorithm to block coded modulation (BCM) codes. A suboptimum version of the algorithm, that can be considered as a possible generalization of [16] is presented at the end.

### **6.2.1 Application of the maximum likelihood version**

Two examples are given in which IMS-MLD algorithm is applied to BCM codes. We assume 8PSK signal constellation and transmission over the additive white Gaussian noise (AWGN) channel.

#### **Example 6.3**

Consider the three-level BCM code where

$B^{(1)} = (32, 6, 16)$  is  $RM(1, 5)$  code,

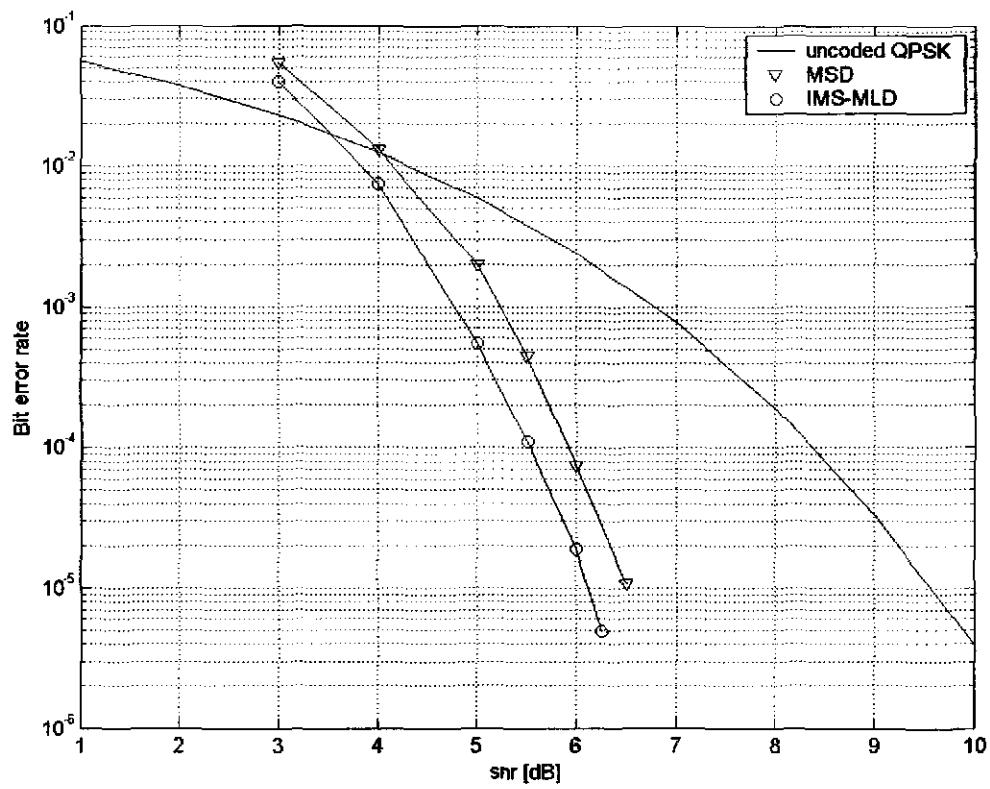
$B^{(2)} = (32, 26, 4)$  is  $RM(3, 5)$  code,

$B^{(3)} = (32, 31, 2)$  is parity check code.

This code has the minimum squared Euclidean distance (MSED) equal to 8, and spectral efficiency 1.97 bits/symbol.

We choose 4, 16, and 32 section trellises for  $B^{(1)}$ ,  $B^{(2)}$ , and  $B^{(3)}$ , respectively.

The bit error performances of conventional MSD and IMS-MLD are given in Figure 6.3. We can see that the iterative algorithm gains 0.4 dB, while the average number of iterations in the first stage varies between 1 and 3, and in the second stage between 0.15 and 2.15. Thus the cost of optimum performance compared to that of conventional MSD is very small.



**Figure 6.3 Performance of the (32,6)(32,26)(32,31) BCM code**

#### Example 6.4

To emphasize the potential of IMS-MLD algorithm over the conventional MSD algorithm, we consider the three-level BCM code in which

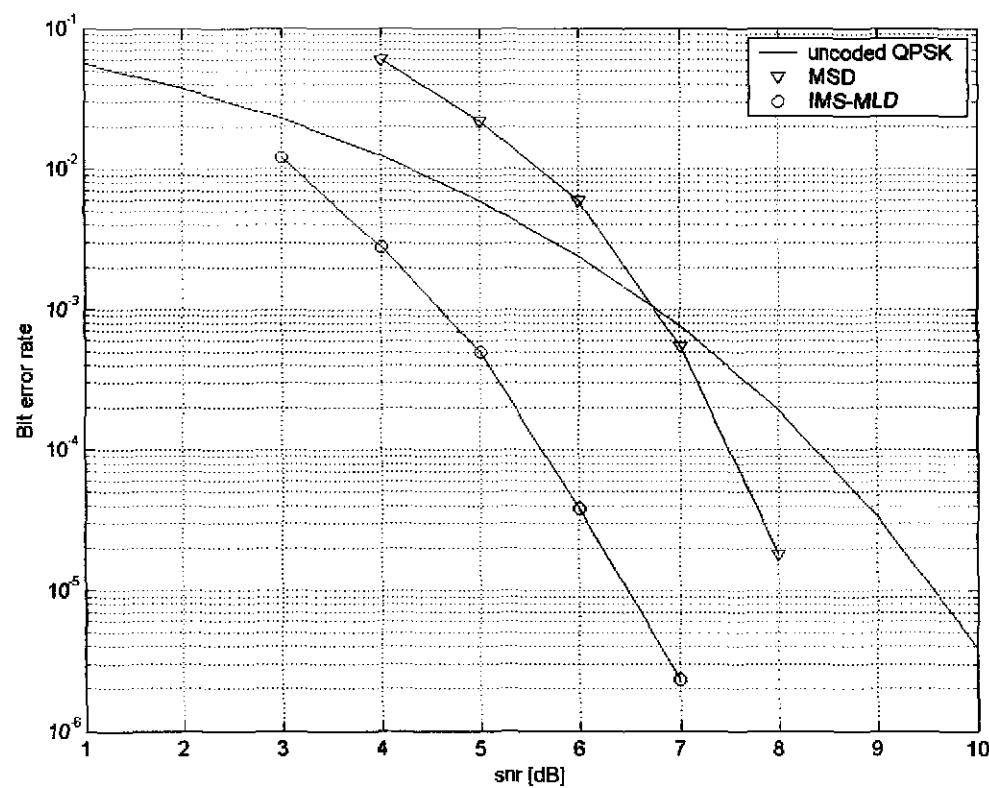
$B^{(1)} = (32, 6, 16)$  is  $RM(1, 5)$  code,

$B^{(2)} = (32, 6, 16)$  is  $RM(1, 5)$  code,

$B^{(3)} = (32, 31, 2)$  is parity check code.

This code has MSED equal to 8 and spectral efficiency 1.34 bits/symbol. From Figure 6.4, we observe that IMS-MLD gains 1.8 dB with respect to MSD. The average number of iterations varies between 1.27 and 16.21 in the first stage, and between 0.3 and 16.27 in the second decoding stage.

The two examples show that IMS-MLD achieves large coding gains compared to conventional MSD. It should be mentioned that the emphasis is on gaining the difference between the MSD and MLD performance in a relatively small number of iterations. A different problem, the construction of optimal BCM codes is discussed in [10, 41].



**Figure 6.4 Performance of the (32,6)(32,6)(32,31) BCM code**

### 6.2.2 Application of suboptimum threshold version

Let us examine the performance of this suboptimum version of iterative multistage decoding algorithm on the example of a BCM code whose outer codes are terminated convolutional codes.

#### Example 6.5

Let  $M$  denote the memory order of a convolutional code,  $R$  its rate, and  $d_{\text{free}}$  its minimum free distance. Consider three-level BCM code with

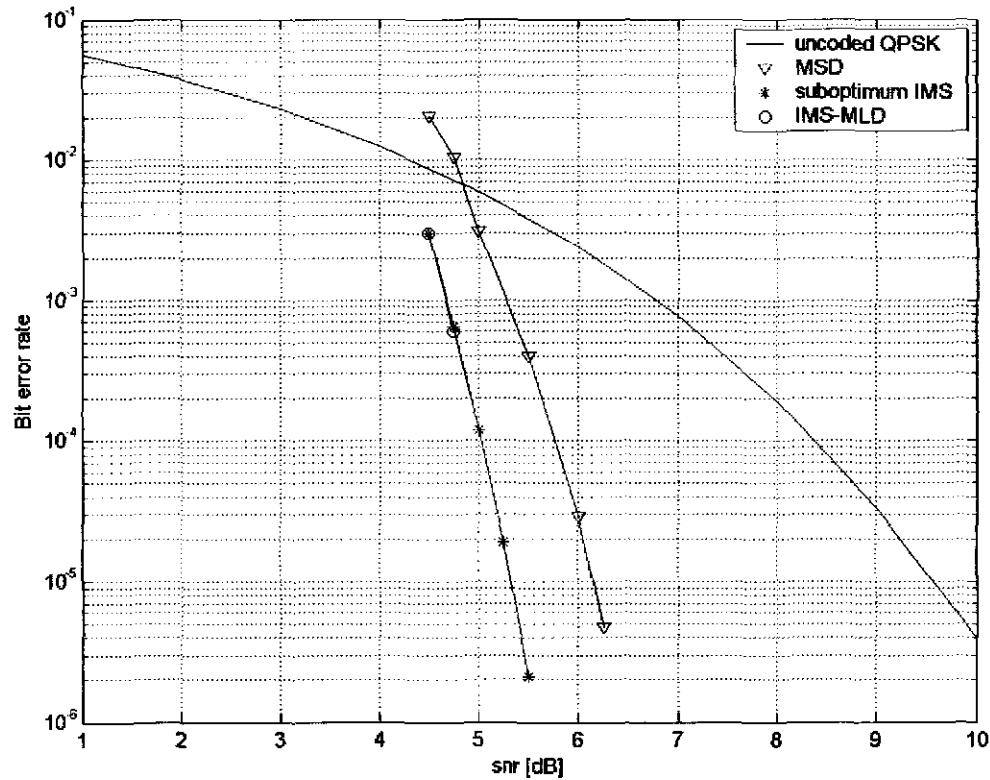
$$B^{(1)} : R = \frac{1}{3}, M = 8, d_{\text{free}} = 24$$

$$B^{(2)} : R = \frac{2}{3}, M = 5, d_{\text{free}} = 6$$

$$B^{(3)} : R = \frac{5}{6}, M = 3, d_{\text{free}} = 3.$$

Codes  $B^{(1)}$  and  $B^{(2)}$  are best (maximum  $d_{\text{free}}$ ) convolutional codes for the given rate, and  $B^{(3)}$  is a punctured convolutional codes. Outer convolutional codes were terminated and considered as block codes of length  $N = 480$ . The BCM code has MSEd equal to 12, and spectral efficiency 1.72 bits/symbol. Spectral efficiency is lowered slightly due to the termination of the convolutional outer codes.

The performance of MSD and that of the iterative decoding with bounded number of iterations starting at each stage (50 for the first and 3 for the second stage), are represented in Figure 6.5.



**Figure 6.5 Performance of BCM code with terminated convolutional outer codes**

The gain of the iterative algorithm compared to MSD is around 0.8 dB. The average number of iterations, however, reaches 46 for the first and 53 for the second level, and results in a large decoding delay.

To combat the complexity and delay, we apply the second suboptimum version, described as threshold decoding in section 5.3.2. The performance of this version of iterative algorithm for the code in Example 6.5 is also shown in Figure 6.5. It can be seen that the performance curve is on the top of the curve of the iterative MSD algorithm with

limited list sizes. However, with the newly introduced conditions, the average list sizes in the first and second stage are reduced from 46 to 6 and from 40 to 3, for the SNR 4.5 dB and 4.75 dB.

In this chapter, the performance of iterative multistage IMS-MLD algorithm was examined for decomposed Reed-Muller codes and BCM codes. Examples were given to support the claims. Results for all examined codes show that the optimum or “close to optimum” performance can be achieved with small decoding cost compared to conventional MSD.

# **Chapter 7**

## **Conclusion**

### **7.1 Work presented**

In this dissertation, we investigated the problem of difference in performance between the conventional multistage decoding algorithm and ML decoding of long codes. Several examples showed that a significant gap exists, more than 1 dB for codes of length 128. The goal was then set to find an improved algorithm that achieves optimum ML or close to optimum performance. Given the good computational efficiency of multistage decoding, it was of interest to keep the basic philosophy of the algorithm and use the multilevel code structure. The search resulted in a novel iterative multistage decoding algorithm, which achieves ML performance.

We started in chapter 4 by analyzing the performance of the list MSD, also presented in [12]. Considering more than one estimate of the particular decoding stage in decoding of subsequent stages proved to be a good method in reducing the error rate, but the decoding complexity increases significantly when the number of estimates becomes large. To achieve ML performance, however, large number of estimates is required in cases of signal being heavily contaminated with noise.

To reconcile the requirement for the large number of estimates passed between decoding stages when necessary, with the requirement for low decoding complexity, we searched for the criteria that determine whether the ML codeword has already been

found. These criteria allow the decoding complexity to be small for clean signal and use the large number of estimates only when necessary (source allocation).

Criteria for stopping or directing search towards the most likely codeword are based on the relation between the ordered metrics of the estimates produced by the list decoder, and on the relation between metrics of estimates in subsequent stages when inner codes are based on partition (coset codes or signal constellation partitioning). The theorems describing these important relations are presented in 5.2.

Based on the criteria formulated in several theorems of chapter 5, the new iterative multistage MLD algorithm is devised and presented in 5.3 for two-level, and in 5.3 for  $m$ -level concatenated codes. The new algorithm uses serial list decoding of the outer codes in which the estimates are generated one-by-one in the reverse order of their likelihoods (most likely one first). In the examples given in this dissertation, list Viterbi decoding [38] modified to accommodate trellises of block codes, was used. However, any list algorithm can be applied within the proposed IMS-MLD scheme. In addition, different list algorithms can be applied in different decoding stages. The only difference will be in implementation. The most computationally efficient list-decoding algorithm should be used for each particular outer code.

In Chapter 6, the algorithm was first applied to decomposed Reed-Muller codes of length 64 and 128. The simulation results for the bit error rate and the average decoding complexity results proved that the newly developed algorithm is highly efficient, and also easy to implement. When compared in average complexity to Viterbi decoding, it

achieves reduction of about  $10^5$ . The average complexity was shown to be a function of signal-to-noise ratio – large for small SNRs (noisy channel) and low for high SNRs (clean channel).

The computational complexity varies not only with SNR, but also from block to block of decoded data. Even when the average complexity is low, the worst case complexity might be very high, causing a long decoding delay. A possible solution for this problems is to request retransmission if ML codeword is not found in certain number of iterations. Another is to decode these “noisy” blocks in background (multi-decoder case), while other subsequent blocks are decoded in the main unit. Thus only this particular information would be delayed while the upcoming data, possibly clean of noise, can be decoded and delivered.

Next, the algorithm was applied to BCM codes with block and convolution codes as outer codes. High computational cost of the new algorithm for the terminated convolutional codes of very large length suggested that the suboptimum version of the *algorithm must be used*.

Several suboptimum versions are presented in chapter 5. One directly limits the maximum number of estimates provided at each stage. This version has the same performance as list MSD [12], but thanks to new criteria, achieves much smaller average computational complexity, as shown in the example.

Another suboptimum version is based on heuristically determined thresholds for the difference in metrics of subsequent estimates. This suboptimum version is heuristic in

nature, but the stopping criteria are essentially based on the optimum conditions presented in the original algorithm. The results show that the performance is very good, with drastically reduced average computational complexity.

## 7.2 Future work

The IMS-MLD algorithm is the first multistage decoding algorithm that achieves maximum likelihood performance. It was shown that extremely good performance - complexity ratios could be achieved for RM codes and BCM codes. The application is, however, in no way limited to these two classes of codes. The algorithm can be applied to wide range of codes, providing that the multilevel structure, as described in chapter 3, exists. Good codes can be constructed with a proper choice of component codes. One such construction is presented in [37], where, starting from multilevel structure of *RM* codes, different codes are constructed by substituting the component codes with other types of codes. Some of the best-known block codes result form this substitution. The IMS-MLD can then be applied to achieve ML performance.

The algorithm is also suited for several suboptimum versions that will be examined in the future research.

A challenging open problem is derivation of upper and lower bounds on error-rate for the list- $\omega$  decoding in multistage decoding algorithm, where  $\omega$  is the list size. These bounds would help the study of convergence of the algorithm and determining a good

value for the maximum number of iterations in each stage (limiting the worst case complexity in the way that least affects the performance).

In addition, we will be looking for the refined versions of the threshold stopping criteria introduced in section 5.5 for the suboptimum version of the iterative multistage algorithm.

Stopping criteria are not always stringent. Sometimes the ML codeword is found in early iteration, but the iterations continue for a while until the stopping criteria is satisfied. We will be looking into how this problem can be minimized.

Another topic left for future work is the examination of performance of IMS-MLD algorithm and its suboptimum versions over the wireless channels. We believe that the statistics of the noise will play an important role in the design of effective suboptimum versions.

## References

- [1] C. E. Shannon, "A Mathematical Theory on Communication," *Bell Sys. Tech. Journal*, vol. 27, pp. 379-423 and 623-656, July and October 1948.
- [2] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260-269, 1967.
- [3] S. Lin, T. Kasami, T. Fujiwara, and M. Fossorier, *Trellises and Trellis-based Decoding Algorithms for Linear Block Codes*. Boston: Kluwer Academic Publishers, 1998.
- [4] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error Correcting Codes," *IEEE Transactions on Information Theory*, vol. IT-23, pp. 371-376, May 1977.
- [5] A. R. Calderbank, "Multi-Level Codes and Multi-Stage Decoding," *IEEE Transactions on Communications*, vol. COM-37, pp. 222-229, March 1989.
- [6] T. Takata, S. Ujita, T. Kasami, and S. Lin, "Multistage Decoding of Multilevel Block M-PSK Modulation Codes and Its Performance Analysis," *IEEE Transactions on Information Theory*, vol. 39, pp. 1204-1218, July 1993.
- [7] G. D. Forney Jr., "Coset Codes II: Binary Lattices and Related Codes," *IEEE Transactions on Information Theory*, vol. IT-34, pp. 1152-1187, September 1988.
- [8] E. L. Blokh and V. V. Zyablov, *Generalized Concatenated Codes*. Moscow, 1976.

- [9] J. Wu, S. Lin, T. Kasami, T. Fujiwara, and T. Takata, "An Upper Bound on the Effective Error Coefficient of Two-Stage Decoding and Good Two-Level Decompositions of Some Reed-Muller Codes," *IEEE Transactions on Communications*, vol. 42, pp. 813-818, February/March/April 1994.
- [10] J. Huber, U. Wachsmann, "Design of Multilevel Codes," in *The Proceedings of the 1995 IEEE Information Theory Workshop (ITW)*, paper 4.6. Rydzyna, Poland, 1995.
- [11] H. T. Moorthy, D. Stojanovic, and S. Lin, "Bounds on Block Error Probability for Multilevel Concatenated Codes," in *The Proceedings of International Symposium on Information Theory and Its Applications*. Victoria, Canada, September 1996, pp. 602-605.
- [12] D. Stojanovic, "Multistage Decoding of Block and Convolutional Codes," University of Hawai'i at Manoa, Master Thesis 1996.
- [13] U. Wachsmann, R. Fisher, and J. Huber, "Multilevel Coding: Use of Hard Decisions in Multistage Decoding,".
- [14] P. M. Ebert, "Error Bounds for Parallel Communication Channels," M.I.T. Research Laboratory of Electronics, Cambridge, Report 448, 1966.
- [15] G. D. Forney Jr., *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.
- [16] U. Dettmar, J. Portugheis, and H. Hentsch, "New Multistage Decoding Algorithm," *Electronics Letters*, vol. 28, pp. 635-636, 1992.
- [17] G. D. J. Forney, "Exponential Error Bounds for Erasure, List, and Decision Feedback Schemes," *IEEE Transactions on Information Theory*, vol. IT-14, pp. 206-220, 1968.

- [18] I. S. Reed, "A Class of Multiple Error Correcting Codes and the Decoding Scheme," *IEEE Transactions on Information Theory*, vol. IT-14, pp. 38-49, 1954.
- [19] Z. Z. Ginzburg, "Multidimensional Signals for a Continuous Channel," *Probl. Perdach. Inform.*, vol. 20, pp. 28-46, 1984.
- [20] S. I. Sayegh, "A Class of Optimum Block Codes in Signal Space," *IEEE Transactions on Communications*, vol. COM-30, pp. 1043-1045, October 1986.
- [21] R. M. Tanner, "Algebraic Construction of Large Euclidean Distance Combined Coding Modulation Systems," in *Abstract of Papers, 1986 IEEE Int. Symp. Inform. Theory*. Ann Arbor, MI, October 1986.
- [22] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "Error Control Systems with Combined Block Coding and  $M$ -ary PSK Modulations," in *The Proceedings of 10th Symp. Inform. Theory and Its Application*. Enoshima, Japan, November 1987.
- [23] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "Construction of Block Codes for  $2^l$ -ary PSK and  $2^l$ -ary QASK Modulations," *Papers of IEICE of Japan*, vol. IT-87-126, pp. 49-54, March 1988.
- [24] E. Biglieri and M. Elia, "Multidimensional Modulation and Coding," *IEEE Transactions on Information Theory*, vol. 34, pp. 803-809, July 1988.
- [25] G. J. Pottie and D. P. Taylor, "Multilevel Channel Codes Based on Partitioning," *IEEE Transactions on Information Theory*, vol. 35, pp. 87-98, January 1989.
- [26] F. P. Kschischang, P. G. Buda, and S. Pasupathy, "Block Coset Codes for  $M$ -ary Phase Shift Keying," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 900-913, August 1989.

- [27] T. Takata, S. Ujita, T. Fujiwara, T. Kasami, and S. Lin, "Linear Structure and Error Performance Analysis of Block PSK Modulation Codes," *IEICE Transactions on Communications, Electronics, Information Systems Japan*, vol. J73-A, pp. 314-321, February 1990.
- [28] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "A Concatenated Coded Modulation Scheme for Error Control," *IEEE Transactions on Communications*, vol. COM-38, pp. 752-763, June 1990.
- [29] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "Cross-over Construction for Multilevel Block Modulation Codes," *IEICE Transactions on Communications, Electronics, Information Systems Japan*, vol. E-73, pp. 1657-1665, October 1990.
- [30] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Linear Structure and Phase Rotation Invariant Properties of Block  $2^l$ -PSK Modulation Codes," *IEEE Transactions on Information Theory*, vol. 37, pp. 164-167, January 1991.
- [31] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Multi-level Block Modulation Codes," *IEEE Transactions on Information Theory*, vol. 37, pp. 956-975, July 1991.
- [32] G. C. Clark Jr. and J. B. Cain, *Error Correction Coding for Digital Communications*. New York: Plenum Press, 1981.
- [33] G. D. Forney Jr., "Convolutional Codes II: Maximum-Likelihood Decoding," *Information and Control*, vol. 25, pp. 222-266, July 1974.
- [34] S. Lin and D. J. Costello Jr., *Error Control Coding Fundamentals and Applications*. New Jersey: Prentice-Hall, 1983.

- [35] V. A. Zinoviev, "Generalized Cascade Codes," *Problemy Perekachi Informatsii*, vol. 12, pp. 2-9, 1976.
- [36] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 55-67, 1982.
- [37] R. Morelos-Zaragoza, T. Fujiwara, T. Kasami, and S. Lin, "Construction of Generalized Concatenated Codes and Their Trellis-based Decoding Complexity," *IEEE Transactions on Information Theory*, vol. 45, pp. 725-731, March 1999.
- [38] N. Seshadri and C. W. Sundberg, "List Viterbi Decoding Algorithms with Applications," *IEEE Transactions on Communications*, vol. COM-42, pp. 313-322, February/March/April 1994.
- [39] D. Stojanovic, M. Fossorier, and S. Lin, "Iterative Multistage Maximum Likelihood Decoding of Multi-Level Codes," in *The Proceedings of Coding and Cryptography*. Paris, 1999.
- [40] M. Fossorier, S. Lin, and D. Rhee, "Bit Error Probability for Maximum Likelihood Decoding of Linear Block Codes and Related Soft Decision Decoding Methods," *IEEE Transactions on Information Theory*, vol. IT-44, pp. 602-605, November 1998.
- [41] U. Wachsmann, R. Fischer, J. Huber, "Multilevel Codes: Theoretical Concepts and Practical Design Rules," *IEEE Transactions on Information Theory*, vol. 45, pp. 1361-1391, 1999.
- [42] F. Hemmati, "Closet Coset Decoding of  $|u| u+v |$  Codes," *IEEE Journal on Selected Areas in Communications*, vol. JSAC-7, pp. 982-988, August 1989.

- [43] D. Stojanovic, S. Lin, and M. Fossorier, "Iterative Multistage Decoding of BCM Codes," in *The Proceedings of ISIT 2000*. Sorento, 2000.