OPTIMUM BIT-BY-BIT POWER ALLOCATION FOR MINIMUM DISTORTION TRANSMISSION

A Thesis

by

ARZU KARAER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2005

Major Subject: Electrical Engineering

OPTIMUM BIT-BY-BIT POWER ALLOCATION FOR MINIMUM DISTORTION TRANSMISSION

A Thesis

by

ARZU KARAER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Costas N. Georghiades
Committee Members,	Scott L. Miller
	Henry F. Taylor
	Marina Vannucci
Head of Department,	Chanan Singh

December 2005

Major Subject: Electrical Engineering

ABSTRACT

Optimum Bit-by-Bit Power Allocation for Minimum Distortion Transmission.

(December 2005)

Arzu Karaer, B.S., Istanbul Technical University

Chair of Advisory Committee: Dr. Costas N. Georghiades

In this thesis, bit-by-bit power allocation in order to minimize mean-squared error (MSE) distortion of a basic communication system is studied. This communication system consists of a quantizer. There may or may not be a channel encoder and a Binary Phase Shift Keying (BPSK) modulator. In the quantizer, natural binary mapping is made. First, the case where there is no channel coding is considered. In the uncoded case, hard decision decoding is done at the receiver. It is seen that errors that occur in the more significant information bits contribute more to the distortion than less significant bits. For the uncoded case, the optimum power profile for each bit is determined analytically and through computer-based optimization methods like differential evolution. For low signal-to-noise ratio (SNR), the less significant bits are allocated negligible power compared to the more significant bits. For high SNRs, it is seen that the optimum bit-by-bit power allocation gives constant MSE gain in dB over the uniform power allocation. Second, the coded case is considered. Linear block codes like (3,2), (4,3) and (5,4) single parity check codes and (7,4) Hamming codes are used and soft-decision decoding is done at the receiver. Approximate expressions for the MSE are considered in order to find a near-optimum power profile for the coded case. The optimization is done through a computer-based optimization method (differential evolution). For a simple code like (7,4) Hamming code simulations show that up to 3 dB MSE gain can be obtained by changing the power allocation on the information and parity bits. A systematic method to find the power profile for linear

block codes is also introduced given the knowledge of input-output weight enumerating function of the code. The information bits have the same power, and parity bits have the same power, and the two power levels can be different.

To my family-annem, babam ve kardesime...

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Georghiades, for his guidance, support and encouragement. I would also like to thank my committee members: Dr. Miller, Dr. Taylor and Dr. Vanucci.

I would like to thank my mother, father and brother for everything they have done for me. Without their endless support and love, I would not be who I am today. Special thanks goes to my friends from WCL who made my life more meaningful. I would like to thank Rebecca F. Morrison for all her friendship and support. I would also like to thank Unoma Ndili and Ekpe Okorafor, Vivek Gulati, Dung Doan, Nitin Nangare, Kapil Bhattad, Janath and Dilani Peiris. I feel really lucky to have met these wonderful people and to have had a chance to study with them.

No words can describe how important Dr. Weerakhan Tantiphaiboontana (Dr. Kim)'s support was. I am very grateful to him for his guidance in the hardest years of my life. There is one more person from whom I learned many things- Angelos D. Liveris. I owe him a lot.

These people became my family in College Station. But my family is not complete without my sisters: Benat Kockar, Yakut Gazi and Burcu Baris Keskin. I would like to thank them for always being there for me and for their love and support. I feel blessed to have them as my sisters.

I also would like to thank Sharif H. Melouk, Vivek Choudry, Sena Karasipahi, Dr. Ibrahim Karaman, Neslihan and Emrah Ozensoy, Adriana Diaz, Trupti Kapadkar and Ezgi Can Eren for their support.

Saving the best for the last, I would like to thank Canim, Hari Sankar, for his love and understanding. Without his help and support, this work would not have been complete. Every minute, I feel thankful to have met him.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Communications is an important part of everyone's life. Everyday we use different communication media such as telephone, TV, radio, internet and cell phones to send and receive information. We owe the speed and accuracy in communication systems to the contributions of many researchers and scientists. Most of the major developments in the history of communications has happened in the past century and the growth in communications systems over the last 50 years has been phenomenal. The basic communication system can be described as a system designed to send messages reliably from a source to a destination.

The functional block diagram of a basic communication system is shown in Figure 1. The information generated by the source which can be in the form of speech, images or text files is sent through a channel to the destination or destinations. As it can be seen from the figure a communication system consists of three main parts- the transmitter, the channel and the receiver. The transmitter converts the message from the source to a suitable form so that it can be transmitted on the channel reliably. The communications channel is the physical medium that is used to send messages from the transmitter to the receiver. This physical medium might be the atmosphere, wire lines, optical fiber cables. The important characteristic of the communication channel is that it corrupts the message sent from the transmitter. The most common form of corruption is called *thermal noise* and it is additive in nature. In wireless transmission, a different kind of corruption occurs due to multipath which results in

---

This thesis follows the style of *IEEE Transactions on Selected Areas of Communications.*

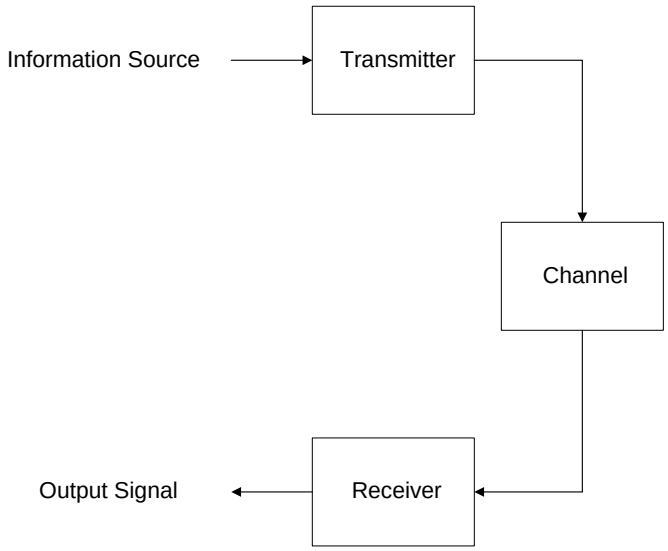

Fig. 1. Basic communication system

the fluctuation of the received signal amplitude. This phenomenon is known as *fading* and is usually very deleterious. The main function of the receiver is to recover the message signal that is contained in the received signal. The presence of noise must be taken into account while designing the optimum receiver. The receiver reverses the operations performed on the message signal by the transmitter.

Communication can be broadly classified into - analog communication and digital communication. Analog signals can be transmitted directly via carrier modulation and demodulated accordingly at the receiver. We call this communication system an analog communication system. Modulation schemes such as Amplitude Modulation (AM) and Frequency Modulation (FM) are examples of analog modulation. Digital Communication is another important way of transmitting data from the source to the destination. Analog signal can either be transmitted by carrier modulation over a channel or can be converted into a digital signal and transmitted via digital modulation. Digital communication is used in the transmission of analog and continuous

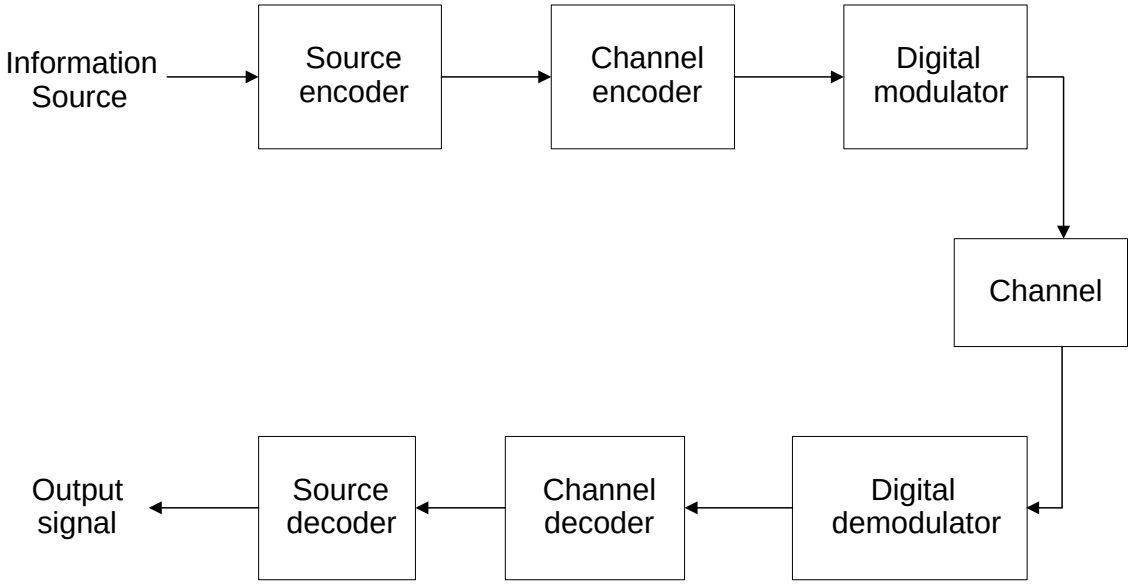

Fig. 2. Digital communication system

time signals such as speech and images or digital signals such as text files. There are some advantages of transmitting an analog signal by using digital modulation techniques. Some of the main advantages of digital modulation are digital transmission provides us with better control of signal fidelity, the digital message can be regenerated in long distance signal transmission. In practice analog and continuous time signals are converted to digital signals for transmission. To transmit an analog signal digitally, the signal is first sampled at Nyquist rate, f_s Hz, where f_s is greater than or equal to twice the highest frequency component in the signal. Figure 2 shows a more functional communication system. Then each sample of the signal is quantized to a set of discrete levels. A group of bits is usually assigned to each level. These bits are usually passed through a source encoder which removes all redundancy from the input bits and it outputs information bits. It is then passed through a channel encoder which adds redundancy in controlled levels to protect against errors that might occur

in the channel. For example, a channel code might take in K information bits which are output from the source encoder and output N coded bits thus forming an (N, K) channel code. The coded bits are passed through a pulse-shaping filter, represented by pulses and modulated on a carrier and transmitted on the channel. The primary purpose of the modulator is to convert a digital pulse into an analog signal which is the only practical signal that can be transmitted.

At the receiver, the demodulator extracts the received signal from the carrier waveform and obtains a value for each transmitted bit. This might correspond to an actual magnitude of the bit value transmitted or to a likelihood value of the bit being '1' or '0'. The channel decoder takes in as input the received values corresponding to the N transmitted coded bits and makes the decision on the K information bits. Coding helps in better performance over the uncoded case for the same bandwidth and same power expended, at the cost of increased complexity. Once the information bits are obtained, the source decoder can decompress them to obtain the original bits from the output of the sampler. These samples can be used to reconstruct the original analog signal that was transmitted. Basically, the operations at the transmitter are inverted at the receiver to obtain the signal that was transmitted. A perfect reconstruction of the signal is never possible owing to the noise in the channel, presence of quantizer which is a non-invertible operation and also the presence of a low-pass filter before the sampler. Each of the above operations is important for a good reproduction of the transmitted signal at the receiver. The design of a communication system is generally constrained by one or more of the following three major factors: power available for transmission of the signal, bandwidth available for transmission and the complexity of the receiver. The ultimate aim of the digital communication system is to minimize bit-error probability or block-error probability or mean-squared error of the system. In this thesis, we are going to look at a joint source-channel coding prob-

lem which results in an optimum/near optimum bit-by-bit power allocation scheme to minimize mean-squared error distortion of an uncoded/coded communication system.

A. Previous Work

A modulation technique that is commonly used for the transmission of digital signals is PCM, pulse code modulation [1]. Conventional PCM transmits all the bits obtained from the quantizer with the same energy. As will be seen in later sections, this scheme does not yield the minimum distortion for an uncoded case. Hence, in 1958 Bedrosian [2] proposed "weighted PCM" for an uncoded conventional PCM system. His work showed that minimization of distortion can be achieved by "weighing" the PCM pulses. In "weighted PCM", the relative amplitudes of the pulses within PCM words are adjusted so as to minimize the distortion between the transmitted and the received amplitude. Adjusting the amplitudes of the pulses is the same as adjusting the energy of each bit while keeping the total energy for each PCM word constant. The energies of the bits are "weighed" differently in order to minimize the mean-squared error between the transmitted and received amplitudes. "Weighted PCM" has been studied further by [3]-[6] and they have suggested near optimum methods for transmitting groups of bits at a particular energy level. For the coded case, providing different protection to streams with different reliabilities have been considered by many authors. This falls under the broad topic of unequal error protection (UEP) [7]. Most of the previous work has approached the problem by allocating different number of parity bits to each of these streams or in other words allocating a lower effective code rate to the stream which requires higher reliability [8]. In this work, we approach the problem differently by allocating different power to different bits of a block code to minimize mean-squared error of the system. Moreover, it is a near

optimum power allocation strategy for linear block codes.

B. Proposed Research

In this thesis, we concentrate on a simple communication system which in essence is very similar to a PCM system. The quantizer is assumed to be uniform and it outputs levels which are represented by a group of K bits. Since we want to keep the system simple, there is no source encoder. The presence of the source encoder complicates matters and since the object of this work is to try to understand a simple communication system, the source encoder is dropped. Two cases are considered: in the first case, there is no coding and the output bits of the quantizer are directly modulated and transmitted on the channel. In the second case, a simple code like single parity check code or a Hamming code is introduced and the resulting system is studied. The ultimate objective of this thesis is to minimize the distortion between the levels obtained at the output of the quantizer in the transmitter and the reconstructed levels at the input of the dequantizer.

As discussed above, for the uncoded case, "weighted PCM" approach has to be followed. We look into finding a closed form expression for the optimum power levels for each bit position that minimize the distortion of the system. In order to solve this optimization problem, Lagrange Multipliers method is used [9]. Finding a closed form analytical expression for power profiles is not possible. Therefore, we use a Chernoff bound [10] on the probability of error expressions in the MSE expression and derive an optimized power profile for the bit positions in the binary value representing a level of the quantizer, which minimizes the MSE. This derivation of power profile values is not explicitly stated in Bedrosian's paper. We also look into the MSE gain

which is given by:

$$\text{MSE gain} = \text{MSE}/\text{MSEuniform} \tag{1.1}$$

where MSE is the MSE from the optimized power profile and MSEuniform is the MSE for the case with equal power for each bit position, is constant beyond a certain SNR. Since Chernoff bound is approximate especially for lower SNRs, we also use a computer-based optimization which applies the principle of differential evolution [11], to get the optimum power profiles. For the computer-based optimization, we use exact probability of error expressions which can be obtained in terms of $\mathcal{Q}(.)$ function. We verify that the computer-based optimization gives power profiles and MSE gain in the order of the Chernoff bound and not surprisingly, the two are the same for higher SNRs.

For the coded case, however, things are trickier. This system applies a simple code like single parity check code or Hamming code to the output of the bits from the quantizer and then transmits them over the channel. The problem is the same as above - to obtain the optimal power profiles to minimize the distortion given by MSE. Analytical solutions for this case are extremely difficult owing to the intractability of the resulting probability of error expressions. Hence we have applied the computer-based optimization to solve this problem. In this case too, we show that a constant MSE gain can be obtained for high SNRs. In the coded case, we first consider knowing all the codewords. Then we try generalizing the MSE expression for different codes. In order to do this, some simplifications need to be made. First, it is assumed that Input-Output Weight enumerating function (IOWEF) of the code is known and the power profiles of the information bits are the same and the power profiles of the parity bits are the same whereas the two power profiles can be different from each other. This approach is used to find a generalization for the MSE expression without the

need to know all the codewords in the code. In the proposed research, we do not try to optimize the quantizer and instead a uniform quantizer is assumed.

C. Organization of the Thesis

This thesis is organized as follows: in Chapter II, we give a brief description of a PCM system and the system description. We define distortion and make a statement of the problem. In Chapter III, we consider power allocation for different bits for the uncoded case. We look into the actual expression and derive Chernoff bounds to the exact expression. We then derive the power profiles and show the performance of the system through simulations. In Chapter IV, we extend the power allocation to a coded case. We consider two different codes, namely, Single Parity Check (SPC) code and Hamming code. We carry out the analysis and derive the optimum power profiles. We later verify these analytical results with simulations. In Chapter V, we derive a more general analytical expression for the coded case based on the Input Output Weight enumerating function of the code under consideration. We verify the power profiles with simulation results.

CHAPTER II

PULSE CODE MODULATION (PCM) AND SYSTEM DESCRIPTION

A modulation technique that is commonly used for the transmission of a digital signal is PCM, pulse code modulation. In this section we will briefly describe a PCM system. In PCM, the continuous time signal is first sampled at a rate f_s Hz, where f_s is greater than or equal to twice the highest frequency component in the signal. Since most signals have a large number of harmonics, in practice, the analog signal is usually low pass filtered to half the frequency of sampling and then sampled. Each sample of the signal is quantized to a set of discrete levels. The quantizer can be uniform or non-uniform. In conventional PCM, it is usually uniform. Each of these quantized levels are represented by a group of K bits (2^K levels). A conventional PCM system transmits these bits directly or might apply a channel code to these bits and then transmit them. Different modulation schemes such as PSK or QAM can be used to transmit the bits through the channel. When binary phase shift keying is used as the modulation scheme, usually equal amplitude pulses are used to represent each of the bits that are transmitted from the transmitter to the receiver. This means that the probability of error is the same for all bit positions. However, in the design of a PCM system, distortion is an important criterion. PCM system is usually designed to reproduce the waveform at the output of the receiver with as small a distortion as possible. In a conventional PCM system, when the digital data are quantized to K-bit signal amplitudes, the amplitude can be represented as:

$$s = \sum_{k=0}^{K-1} b_k 2^k \tag{2.1}$$

in terms of b_k which is the kth bit and it can take values from $\{0, 1\}$. The mean squared distortion between the transmitted and received amplitude can be represented as:

$$\text{MSE} = \sum_{k=0}^{K-1} P_e^{(k)} 2^{2k} \tag{2.2}$$

where $P_e^{(k)}$ is the probability the kth received bit is in error. If the error probabilities were all equal, the most significant bit contributes $2^{2(K-1)}$ more to the mean squared error than the least significant bit. In other words, errors that occur due to characteristics of the channel in the more significant bits make a bigger contribution to the distortion than the less significant bits. A smaller mean-squared error can result if the error probability of the most significant bit is decreased and the probability of error of the least significant bit is increased. Based on this approach, it appears that in PCM an improved performance can be obtained by "weighting" the various pulses. This scheme is a modified form of PCM which is called the "weighted PCM".

A. System Description

The continuous time analog signal is first sampled and then quantized uniformly to a set of equally likely signal amplitudes as shown in Figure 3. The output of the quantizer is naturally mapped to K-bits. Then it is BPSK modulated and transmitted over the channel. The channel is assumed to be additive white Gaussian noise (AWGN) channel with two-sided power spectral density of $N_0/2$. Between the quantizer and the modulator channel coding may or may not be used. As will become obvious in the next section, the amplitudes of the transmitted bits in the PCM signals are weighted by different factors depending on the position of the bit to reduce the

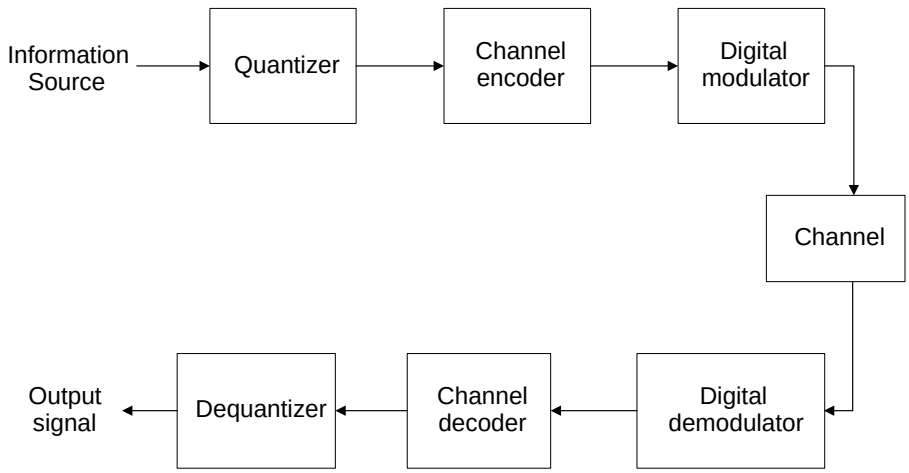

Fig. 3. System description

MSE. At the receiver, the received values r_k can be represented as:

$$r_k = \sqrt{E_s a_k} c_k + n_k \qquad k = 0, \ldots, K-1 \tag{2.3}$$

where E_s is the energy of the PCM symbol, a_k are the "weighing factors" $\sum_{k=0}^{K-1} a_k = 1$ and n_k is the AWGN of variance $N_0/2$. Here c_k is the BPSK symbol which can be represented as

$$c_k = 2b_k - 1 \tag{2.4}$$

and takes values from $\{-1, 1\}$.

B. Distortion

As stated before, the analog signal from the source is quantized into $2^K = M$ levels. These M levels are mapped onto binary numbers and either encoded with a channel code and modulated or directly modulated. Let i denote the level that is transmitted

and let j denote the level that is reconstructed after the demodulator and/or channel decoder. Then, distortion between received and transmitted amplitudes, D, can be expressed as:

$$\mathrm{D} = \sum_{i,j} \pi_i C_{i,j} \mathrm{P}[j|i], \tag{2.5}$$

where π_i is the *a priori* probability of the transmitted level i, $C_{i,j}$ is the cost function, and $\mathrm{P}[j|i]$ is the probability that j is received given that i is transmitted. In this thesis, cost function $C_{i,j}$ is defined as:

$$C_{i,j} = (i - j)^2, \tag{2.6}$$

and the distortion with this cost function is usually defined as the *mean-squared error* (MSE) distortion. The distortion in (2.5) should be minimized. It can be done by finding the power for each bit that minimizes the distortion given in (2.5). Therefore, the optimization problem can be stated as:

$$\mathrm{D}^* = min_{\{a_k\}} \sum_{i,j} \pi_j C_{i,j} P[i|j] \tag{2.7}$$

subject to

$$\sum_{k=0}^{K-1} a_k = 1, \qquad a_k \geq 0 \tag{2.8}$$

In (2.7), D* is the minimum distortion possible over all power profiles, $\{a_k\}$s. To minimize the distortion, we need to minimize the probability of symbol error which can be done through choosing the decision regions for the received amplitudes appropriately and also finding the best a_ks. Minimization of (2.7) over the power profile is a hard problem for the exact probability of error expression for BPSK modulation. Therefore, we resort to using a Chernoff bound approximation on the P_e expression.

As it will become clearer in the next section, this makes it easier to find a closed form analytical expression for the uncoded case. The Chernoff bound is loose for lower SNRs but becomes very tight at higher SNRs as we show through computer-based optimization of the exact expression using differential evolution for distortion. This will become obvious in the next chapter.

CHAPTER III

UNCODED CASE

In this chapter, we discuss the minimization of distortion for the uncoded PCM system. Figure 4 shows the basic communication system for the uncoded case. Similar to the system described in Chapter II, we assume a quantizer, BPSK modulation on AWGN channel and a hard decision decoder at the receiver.

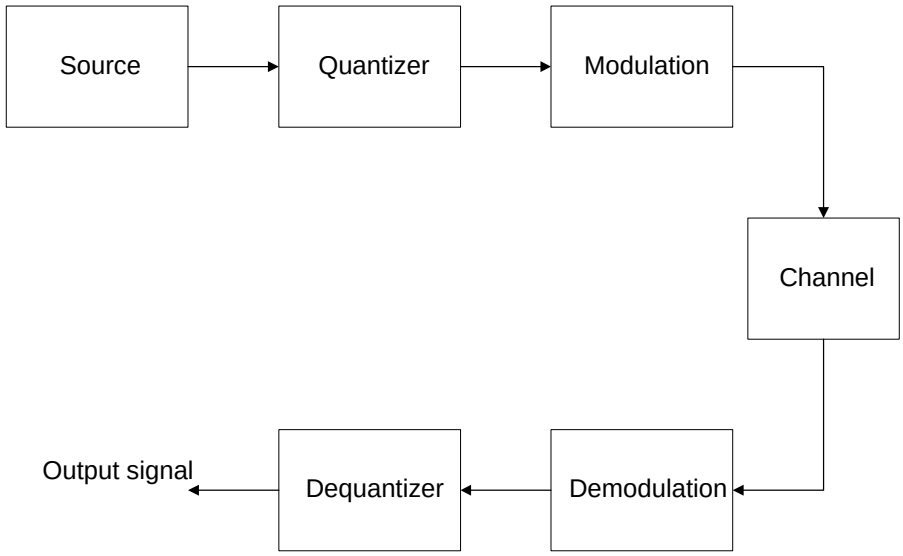

Fig. 4. Block diagram for uncoded communication system

The equation for distortion can be given by (2.5). For an uncoded case with hard-decision decoding, the decision regions can be explicitly determined, as each transmitted bit is independent of the others. Therefore, an exact expression for the probability of bit error can be determined in terms of $\mathcal{Q}(.)$ functions. This enables us to find an exact MSE expression for the uncoded case. Then the MSE expression is optimized in order to find the optimum power profile for each bit. This can be done in two ways. First, optimization can also be done analytically by using a Chernoff

bound on the probability of error expression and the power profiles can be determined analytically. Second, optimization of the MSE expression can be done by using a computer-based optimization algorithm called the differential evolution and this way the optimum power profile for each bit and the optimum MSE can be determined. These will be explained in detail in the next sections. It will also be shown that there is always a gain in MSE if an optimal power profile is chosen for transmission of each of the bit positions. This gain is a non-increasing function of SNR and becomes constant at high SNRs.

A. Distortion

In an uncoded PCM system with a uniform quantizer and natural binary mapping, the transmitted signal i is given by:

$$i = \sum_{k=0}^{K-1} b_k 2^k. \tag{3.1}$$

where $M = 2^K$ and b_k are the bits in the binary representation of i which takes values from $\{0, 1\}$. Note that π_i P$[j|i]$=P$[i, j]$ in (2.5) can be stated as the expected value of (2.6). Substituting (3.1) for i and j into MSE distortion,

$$\text{MSE} = \mathbb{E}[(\sum_{k=0}^{K-1} (b_k^{(i)} - b_k^{(j)}) 2^k)^2] \tag{3.2}$$

The terms inside the expected value will be nonzero only when the corresponding bit positions in the transmitted and received disagree. The cross-terms in the differing bits disappear since the channel is additive white Gaussian and the receiver makes a decision on each bit independent of the others. Then (3.2) can be simplified to:

$$\text{MSE} = \sum_k P[b_k^{(i)} \neq b_k^{(j)}] 2^{2k} \tag{3.3}$$

The probability of error expression in (3.3) is the probability of error for each bit. Then, the MSE distortion between the transmitted and received amplitude can be derived from (3.3) as

$$\text{MSE} = \sum_{k=0}^{K-1} P_e^{(k)} 2^{2k} \tag{3.4}$$

Since the most significant bit contributes more to the MSE distortion, a smaller minimum MSE distortion can be obtained by decreasing the probability of error for the more significant bits. If the modulation used is BPSK and the channel is additive white Gaussian noise (AWGN), probability of error is given by:

$$P_e = \mathcal{Q}\left(\sqrt{\frac{2E_s}{N_0}}\right), \tag{3.5}$$

where $\mathcal{Q}(.)$ is the $\mathcal{Q}$ function related to the Gaussian pdf and E_s/N_0 is the signal to noise ratio (SNR) of the received BPSK signal. From Eqn. (3.5), it can be seen that the probability of error can be decreased by increasing the energy of the pulse which is equivalent to increasing the amplitude of the pulse. Going back to Eqn. (2.2), the amplitudes of the more significant bits should be increased to reduce their probability of error and hence the overall MSE. From (2.2) and (3.5) it can be seen that the amplitudes of the pulses of the PCM signal can be optimized to get the minimum MSE. Therefore the optimization problem for the uncoded case can be stated as

$$\min_{\mathbf{a}} \sum_{k=0}^{K-1} 2^{2k} \mathcal{Q}\Big(\sqrt{(2\text{SNR}a_k)}\Big) \tag{3.6}$$

subject to :

$$\sum_{k=0}^{K-1} a_k = 1, \qquad a_k \geq 0 \tag{3.7}$$

where the total energy for any PCM codeword is taken to be E_s and SNR=$\frac{E_s}{N_0}$ and a_k are nonnegative weights that add up to 1. Lagrange multipliers can be used to find an analytical solution of this optimization problem. Therefore, solving (3.6) is enough to find the optimum energies for the bit positions. But finding a closed form exact solution through Lagrange multipliers is not possible. Therefore, it is easier to look for a numerical solution to the problem or to find an approximation to the distortion expression given in (3.6).

B. Chernoff Bound

A good approximation to the solution of (3.6) is given in [2]. Another way of finding a good analytical approximation is to use Chernoff bound instead of the exact probability of error expression to bound the probability of error and thus the mean squared error. $\mathcal{Q}(x)$ can be upper bounded as

$$\mathcal{Q}(x) \leq \frac{1}{2} e^{\frac{-x^2}{2}} \tag{3.8}$$

Using the Chernoff bound given in (A.1) on the probability of error, the problem in (3.6) becomes

$$\min_{\mathbf{a}} \frac{1}{2} \sum_{k=0}^{K-1} 2^{2k} e^{-\mathrm{SNR} a_k} \tag{3.9}$$

subject to:

$$\sum_{k=0}^{K-1} a_k = 1, a_k \geq 0 \tag{3.10}$$

which admits an analytical solution:

$$\hat{a_k} = \frac{(2k+1-K)\ln(2)}{\mathrm{SNR}} + \frac{1}{K} \quad k = 0, 1, \ldots, K-1 \tag{3.11}$$

where K is the total number of bits and $\hat{a_k}$ $\geq$0. The detailed solution is given in Appendix A. The positiveness condition for all k is satisfied for

$$\text{SNR} \geq K(K-1)\ln(2) \tag{3.12}$$

In this case the resulting MSE is

$$\text{MSE} = \frac{K}{4} 2^K e^{\frac{-\text{SNR}}{K}} \tag{3.13}$$

If the condition in (3.12) is not satisfied, then some a_k will be calculated to be negative. Then as in computing the capacity through "waterfilling" these a_k will be set to 0. Now the constraint is for the remaining $\hat{a_k}$s to add up to 1. This procedure can be summarized as

- Calculate $\hat{a_k}$ by using (3.11).
- If $\hat{a_k} \geq 0$, the value of $\hat{a_k}$ is set to the calculated value.
- If $\hat{a_k} < 0$, then the value of $\hat{a_k}$ is set to 0. Then the sum of the remaining $\hat{a}k$s must equal 1.

The above procedure is repeated for all k where k=0, 1, . . . , K-1. The optimization problem is solved for 4, 6, 8 bits and the plots are given in Subsection D.

C. Differential Evolution

In order to find a numerical solution to the problem stated in (3.6), computer-based optimization techniques have to be used. In order to solve the problem given in (3.6), $fmincon$ in MATLAB is a possible program which can be used. However, there are some problems in using this optimization tool. It was seen that it converged to a local minimum. Therefore, we used a different optimization algorithm called *Differential*

Evolution (DE) . DE is a simple, population-based algorithm which can minimize a stochastic function. In DE, the initial population is randomly generated. A new parameter is generated by adding the difference between two population members to a third member. After these operations are performed, the function is evaluated at the newly generated parameter and the predetermined population vector. If the resulting vector gives a lower objective function value than the original population member, the new vector is kept in place of the original population member. Because of this property, it almost always can end up in the global minimum.

D. Results

The results for the uncoded case is obtained by using the analytical method given in Section B and the differential evolution method on the exact MSE expression. These include the MSE plots, MSE gain plots and the power profiles for K=4, 6, 8 bits. The plot for the MSE for K=4, 6, 8 bits is shown in the Figure 5. This figure shows the optimum MSE that is calculated from the exact MSE expression and the upper bound on MSE that is calculated from the Chernoff bound expression given in (3.13). As can be seen from Figure 5, Chernoff bound is a close upper bound on the MSE expression and the Chernoff bound curves behave the same way as the MSE curves found from the exact MSE expression. Power profiles that are found by using the differential evolution on the exact MSE expression and the analytical method described in Section B are plotted. In these plots the most significant bit is the lowest one on the histogram chart whereas the least significant bit is the highest. Figure 6, Figure 7, Figure 8 show the energy profile distribution for K=4, 6, 8 respectively determined by using the differential evolution on the exact MSE expression. As can be seen from these plots, for lower SNRs more power is given to the more significant

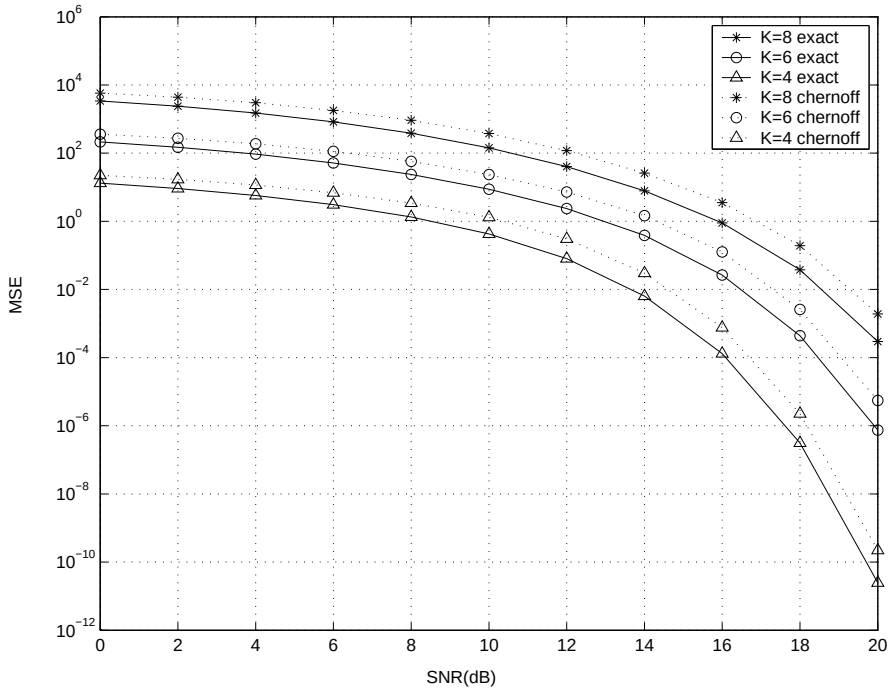

Fig. 5. Optimum MSE found by using the DE method and Chernoff for uncoded case

bits whereas the less significant bits get less energy. When the channel is noisy, more power is allocated to the more significant bit in order to protect it more against the errors introduced by the channel. As SNR increases, the power profiles tend towards uniform power profile for each bit. As can be seen from Figure 6, for K=4 for SNRs

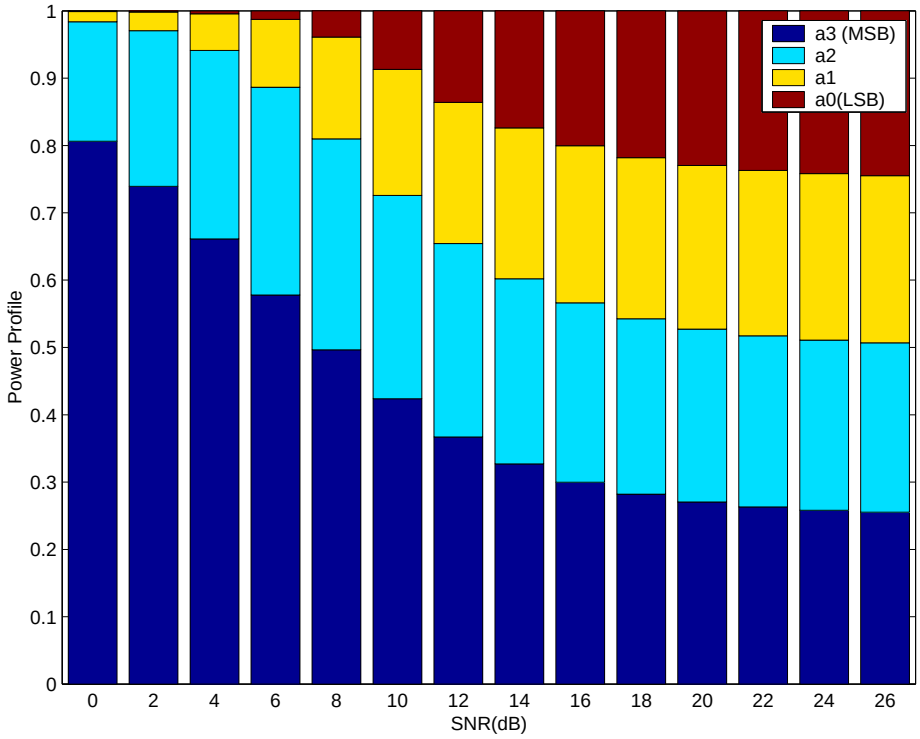

Fig. 6. Optimum power profiles found by using the DE method for the K=4 in uncoded case

lower than 8 dB, the power on the less significant bit is negligible compared to the power allocation on the more significant bit.

Therefore, for lower SNRs you do not need to transmit all the bits. For K=6, all the bits are allocated power only at 12 dB. For SNRs lower than 12dB, you can transmit a smaller number of bits.

As can be seen from Figure 8 for K=8, all the bits have power allocation only at

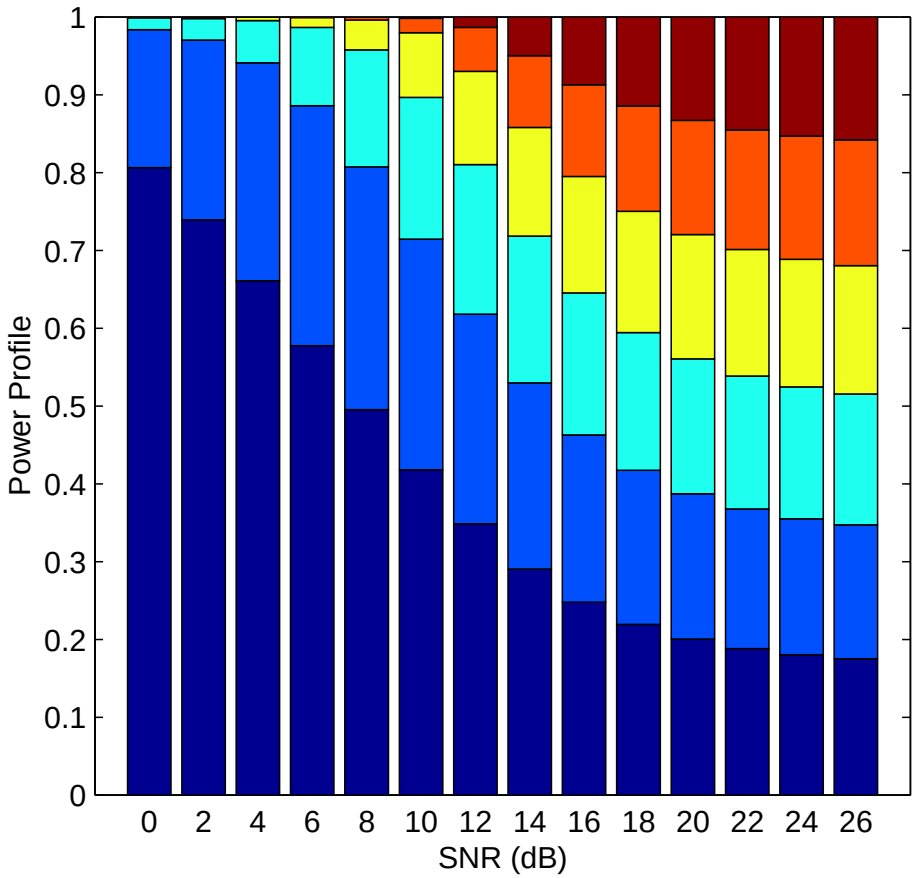

Fig. 7. Optimum power profiles found by using the DE method for the K=6 in uncoded case

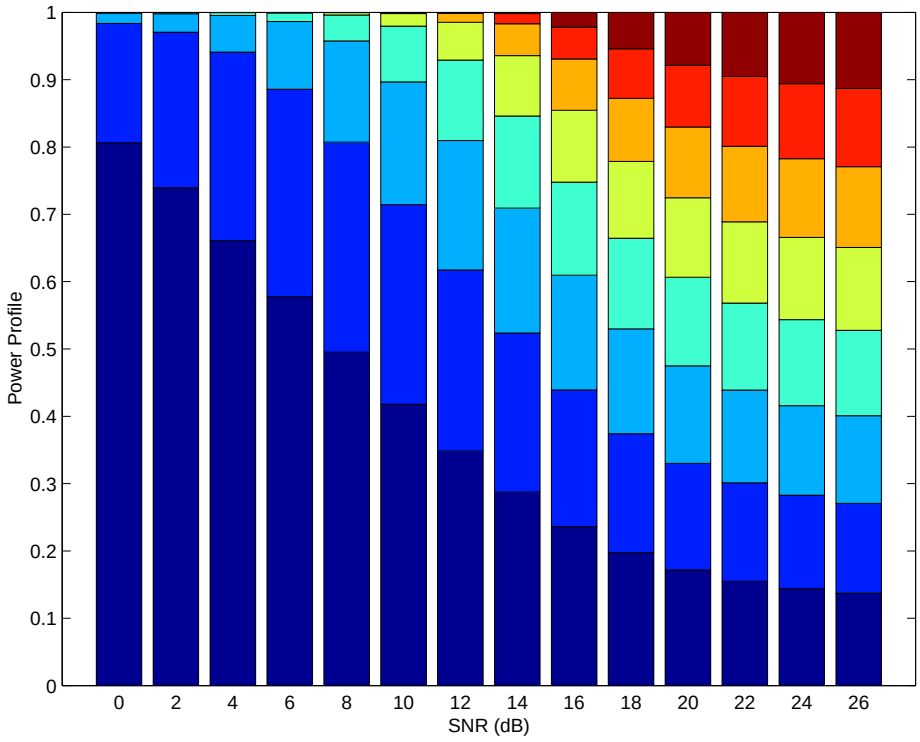

Fig. 8. Optimum power profiles found by using the DE method for the K=8 in uncoded case

16 dB. For SNRs lower than 16 dB, you do not need to transmit all the bits. But for higher SNRs all the bits need to be transmitted with nearly equal power allocation.

The power profile graphs found by using the Chernoff bound method described in Section B and the differential evolution method for K=4, 6, 8 are compared. The power profile graphs generated by using the Chernoff bound is given in Figure 9, Figure 10, Figure 11 respectively. From the inspection of the power profile figures, it can be said that Chernoff bound also gives a close approximation on the power profiles for higher SNRs, especially for SNR greater than 10 dB. This shows that the analytical results are more consistent with the results obtained from the exact MSE expression for higher SNRs.

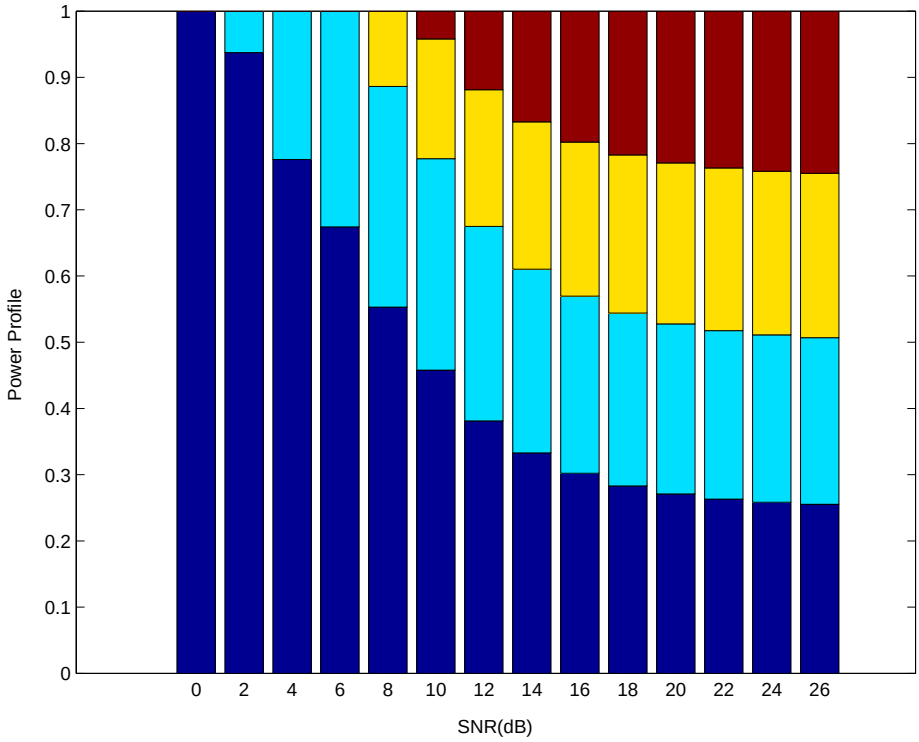

Fig. 9. Optimum power profiles found by using Chernoff bound for K=4 in uncoded case

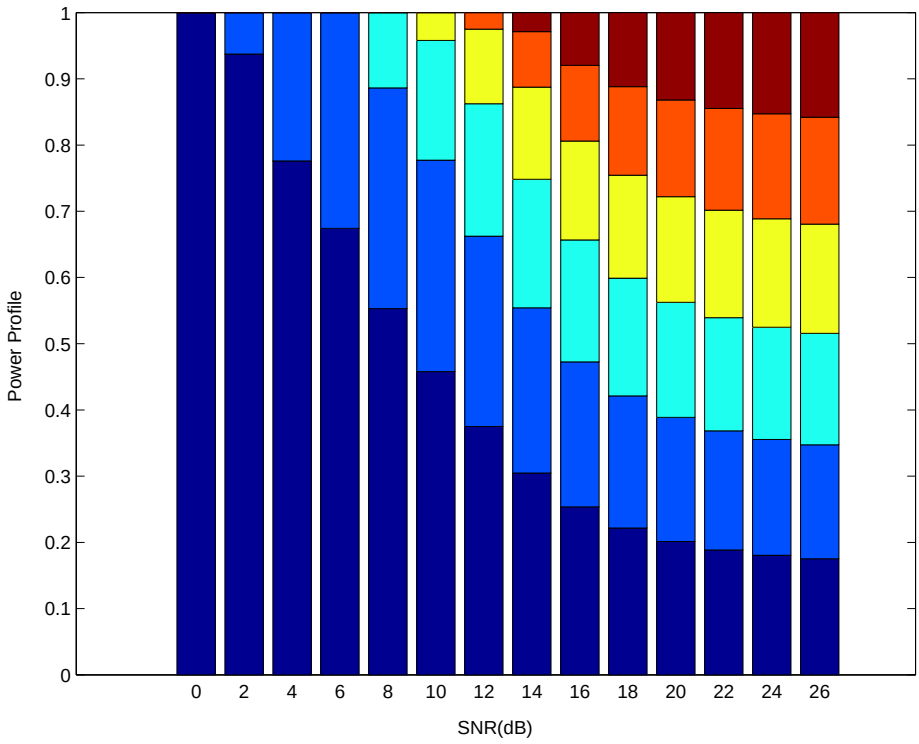

Fig. 10. Optimum power profiles found by using Chernoff bound for K=6 in uncoded case

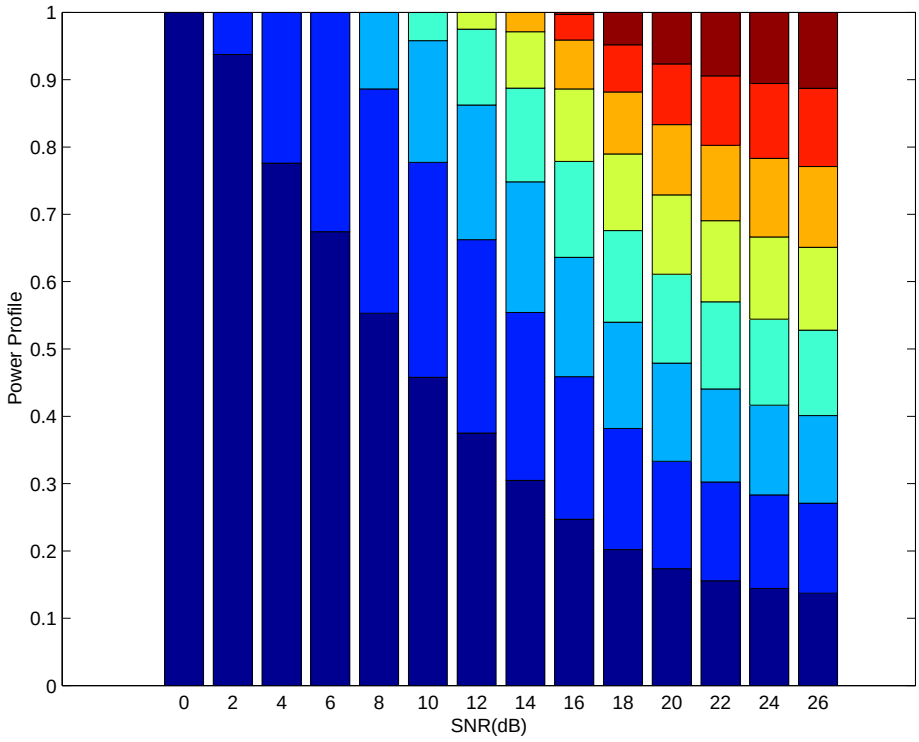

Fig. 11. Optimum power profiles found by using Chernoff bound for K=8 in uncoded case

The MSE gain over the uniform power profile for K=4, 6, 8 is given in Figure 12. This gives a good insight into how much MSE gain can be achieved over the uniform power profile. As can be seen from this plot, the MSE gain for each of the cases (K=4,6,8 bits) increases and it reaches a level where the increase is so small that it seems to be constant for high SNRs. This behavior suggests that MSE gain will reach a certain level and stay constant as SNR increases. This can be shown through analytical expressions too. Looking at the power profiles, it is seen that as SNR increases the power profile approaches a uniform power profile. Although it is not easy to see form the plots it does not equal the uniform power profile exactly. There is still some small deviation from the uniform power profile. Even a small deviation causes a MSE gain in dB over the MSEuniform. This can also be shown by using analytical expression given in (3.13). As had been mentioned earlier, MSE gain is the ratio of the optimum MSE to the MSE for the uniform power case. To find an analytical expression for the optimum MSE, (3.11) is substituted for a_k in (3.9). To find the MSE when the power profile is uniform

$$a_k = \frac{1}{K} \tag{3.14}$$

is substituted for a_k in (3.9). Then, the ratio of these two MSEs are found which yields the MSE gain expression which is found as

$$\frac{4^K - 1}{3K2^{(K-1)}} \tag{3.15}$$

which is independent of SNR. For $K = 8$ the MSE gain can be calculated as 13.29 dB, for $K = 6$ MSE gain is 8.51 dB and for $K = 4$ it is 4.24 dB. These match the values obtained by using the differential evolution for high SNRs (greater than 10 dB). These values agree with the values in Figure 12. In this figure solid lines show the MSE gain obtained from optimization of the exact MSE expression whereas the

dotted lines show the MSE gain obtained by using the power profile obtained from the Chernoff bound.

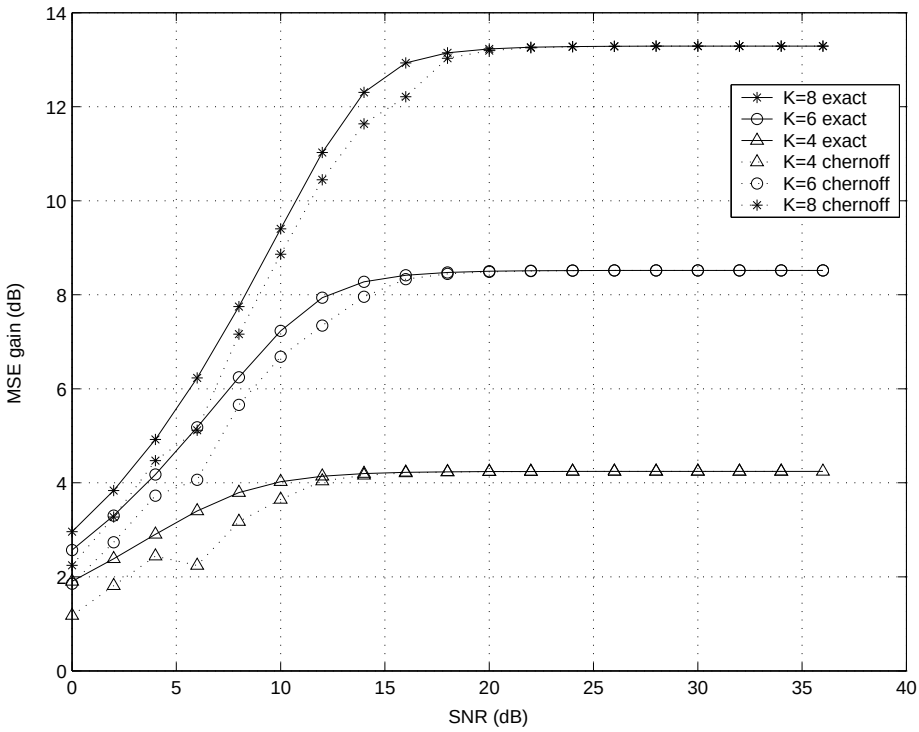

Fig. 12. MSE gain of the optimum energy over the uniform energy with exact expression (bold) and Chernoff bound (dotted)

Our results suggest that as SNR increases the MSE gain reaches a level where it stays constant.

The uncoded system is simulated for K=4 by using the optimum power profile found from differential evolution and the uniform power profile. As can be seen from Figure 13, the plot from the simulation results falls on top of the exact MSE which is determined by using the differential evolution. As it was shown before, Chernoff bound is an upper bound which closely approximates the behavior of the exact MSE and the MSE found from the simulations. MSE gain is also calculated from the

simulation results and plotted against the MSE gain found from the MSE expression. This is given in Figure 14. As it can be seen from this plot, the simulation gives the same result as the exact MSE gain up to 14 dB. It is very hard to run the simulations for higher SNRs since MSE value and hence bit error rate is very small. Figure 13 and Figure 14 show that simulation results are the same as the MSE and MSE gain values calculated from the MSE expression as expected.

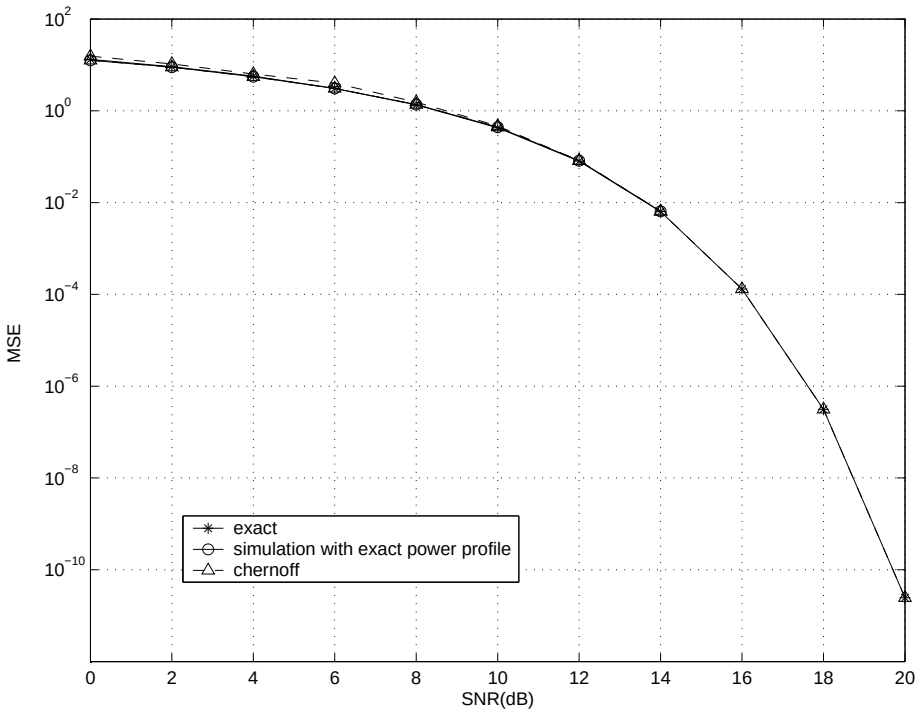

Fig. 13. MSE plot for K=4 from simulation, Chernoff bound and the exact expression

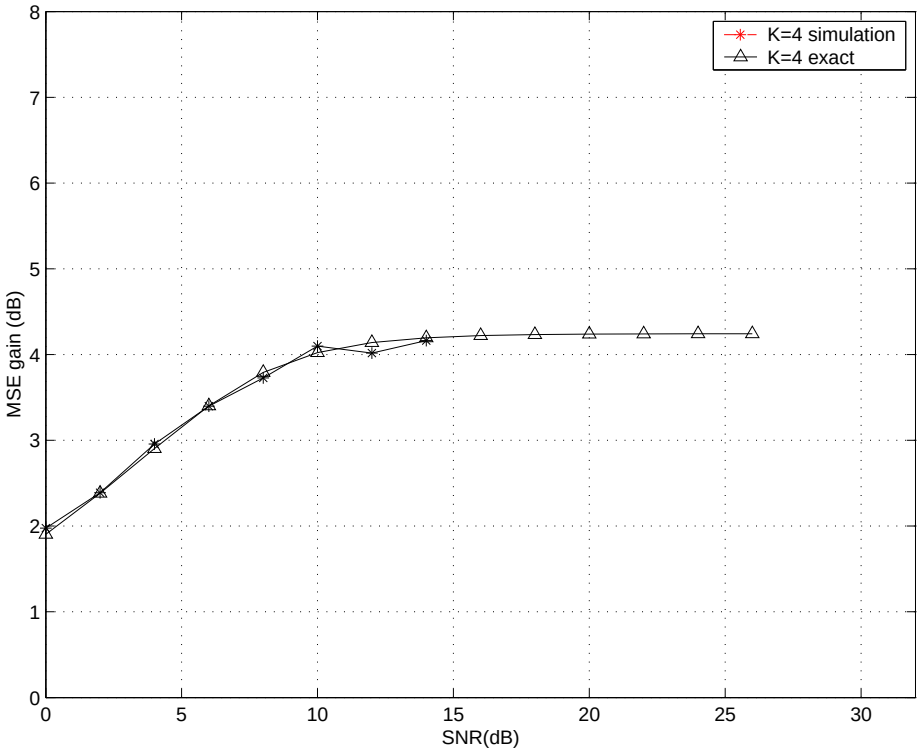

Fig. 14. MSE gain plot for K=4 from simulation, Chernoff bound and the exact expression

CHAPTER IV

CODED CASE

The system assumed for the coded case is similar to the one for the uncoded case except for the presence of a channel code. Before BPSK modulation, all the resulting bits from the quantizer are passed through a channel coder which maps K information bits to N bits. In this chapter, the code assumed is a Single Parity Check (SPC) code and a (7,4) Hamming code. The coded bits are then BPSK modulated and transmitted. At the receiver, they are demodulated and a soft-decision decoder is used for obtaining the information bits. With soft-decision decoding, the decision regions for each codeword are very difficult to obtain. Hence, an upper bound on the probability of error is usually sought which is not very exact especially at low SNRs. Finding an exact MSE expression for the coded case is not possible. In order to find a closed-form expression for the power profiles Chernoff bound can be used on the probability of error expressions, but this does not ultimately yield a closed-form analytical solution for the power profiles. Hence, in the first part of this chapter we obtain the power profile from an approximate MSE expression which is an upper bound by using DE optimization. As it will become clear, the derivation of this MSE expression requires the knowledge of all the codewords. In the next chapter, we extend the DE optimization to a more general distortion expression which only requires knowledge of the input output weight enumerating function (IOWEF). In the coded case, similar to the uncoded case, we obtain a constant gain in MSE asymptotic in SNR.

A. Optimum Receiver

Assume that the N bits of the codeword are modulated over BPSK (c_k) and transmitted over the AWGN channel to obtain:

$$r_k = \sqrt{E_s a_k} c_k + n_k \qquad k = 0, \ldots, N-1 \tag{4.1}$$

$$\mathbf{r} = [r_0 r_1 \ldots r_{N-1}] \tag{4.2}$$

$$\mathbf{c} = [c_0 c_1 \ldots c_{N-1}] \tag{4.3}$$

where n_k is the AWGN noise with two-sided power spectral density $N_0/2$.

The decision rule for the optimum receiver is

$$\max_{\mathbf{c}_m} \mathrm{p}(\mathbf{r}|\mathbf{c}_m) \tag{4.4}$$

where m shows the mth codeword and $m = 0, 1, 2, \ldots, 2^K - 1$. From 4.2, it can be seen that $\mathrm{p}(r_k|c_k)$ is Gaussian with mean $\sqrt{E_s a_k} c_k$ and variance $N_0/2$. Since they are statistically independent from each other (4.4) can be expressed in terms of $\mathrm{p}(r_k|c_k)$ as

$$\max_{\mathbf{c}_m} \prod_{k=0}^{N-1} \mathrm{p}(r_k|c_{m,k}) \tag{4.5}$$

which is equivalent to

$$\max_{\mathbf{c}_m} \sum_{k=0}^{N-1} \ln \mathrm{p}(r_k|c_{m,k}) \tag{4.6}$$

If the Gaussian pdf is substituted for $\mathrm{p}(r_k|c_{m,k})$ in (4.6) then

$$\max_{\mathbf{c}_m} \sum_{k=0}^{N-1} \ln \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(r_k - \sqrt{E_s a_k} c_{m,k})^2}{2\sigma^2}} \tag{4.7}$$

can be found. Maximizing (4.7) is equivalent to minimizing

$$\min_{\mathbf{c}_m} \sum_k \|(r_k - \sqrt{E_s a_k} c_{m,k})\|^2 \tag{4.8}$$

$$= \min_{\mathbf{c}_m} \sum_k (-2\mathrm{Re}[r_k \sqrt{E_s a_k} c^*_{m,k}] + \|r_k\|^2 + E_s a_k \|c_{m,k}\|^2) \tag{4.9}$$

The term r_k^2 in (4.9) is common to all the signals, so it can be ignored. The term $\sum_k E_s a_k \|c_{m,k}\|^2$ is constant and can be ignored. * shows conjugation. Then the decision rule becomes

$$\min_{\mathbf{c}_m} - \sum_k (2\mathrm{Re}[r_k \sqrt{E_s a_k} c^*_{m,k}]) \tag{4.10}$$

In the case of BPSK, (4.10) is equivalent to

$$\min_{\mathbf{c}_m} - \sum_k (2 r_k \sqrt{E_s a_k} c_{m,k}) \tag{4.11}$$

This is also equivalent to

$$\max_{\mathbf{c}_m} \sum_k r_k \sqrt{E_s a_k} c_{m,k} \tag{4.12}$$

B. Analysis of the MSE Expression for Coded Case

As mentioned in the system description, between the quantizer and modulator channel coding may be present. The channel codes that are used in this system are $(K+1, K)$ single parity check codes and (7,4) Hamming code. The difference between the uncoded and the channel coded system is in the receiver. In the coded case, soft-decision decoding is made. Also in the coded case unlike the uncoded case, there is no closed form expression for optimizing MSE. The probability of error $\Pr[j|i]$ in (2.5) can be derived from the soft-decision decoding. The decision rule for the optimum decoder is given in the subsection on optimum decoder. Let us assume that the

codeword, c_i, corresponding to the quantizer level i is transmitted. The receiver will decide on the codeword, c_j representing the quantizer level j if correlation metric given in (4.12) for the codeword, c_j, corresponding to the quantizer level j is maximum. Instead of trying to derive the exact probability of error for $P[j|i]$ we resort to an upper bound for the soft decision decoding. There will be a codeword error if

$$\sum_{k=0}^{N-1}\left(\sqrt{E_s a_k}c_{i,k}+n_k\right)\sqrt{a_k}c_{j,k} > \sum_{k=0}^{N-1}\left(\sqrt{E_s a_k}c_{i,k}+n_k\right)\sqrt{a_k}c_{i,k} \tag{4.13}$$

If the right-hand side of (4.13) is subtracted from the left side of (4.13), the only terms that are left will be the terms that differ in bit positions.

$$\sum_{c_{i,k}\neq c_{j,k}}\left(\sqrt{E_s}a_k(c_{i,k})^2-\sqrt{E_s}a_k c_{i,k}c_{j,k}+\sqrt{a_k}n_k c_{i,k}-\sqrt{a_k}n_k c_{j,k}\right) > 0 \tag{4.14}$$

After simplifications, (4.14) is a Gaussian random variable with mean $\sqrt{E_s}\sum_{c_{i,k}\neq c_{j,k}} a_k$ and variance $\frac{N_0}{2}\sum_{c_{i,k}\neq c_{j,k}} a_k$. P$[j|i]$ can be upper-bounded by:

$$\mathcal{Q}\left(\sqrt{\frac{2E_s\sum_{c_{i,k}\neq c_{j,k}} a_k}{N_0}}\right). \tag{4.15}$$

Since $SNR=\frac{E_s}{N_0}$, (4.15) can be written as

$$\mathcal{Q}\left(\sqrt{2\text{SNR}\sum_{c_{i,k}\neq c_{j,k}} a_k}\right) \tag{4.16}$$

As can be seen from (4.15) and (4.16) the probability of codeword error depends on the positions of the bits that are different.

C. MSE Expression for $(K+1,K)$ Single Parity Check (SPC) Code

(3,2),(4,3) and (5,4) Single Parity Check codes are considered and (2.5) is calculated for these codes. As mentioned earlier, the mapping in the quantizer is natural binary.

As an example the mapping of the codewords for (4,3) SPC codeword is given in Table I. In (2.5) the probability of error expression is calculated by using (4.16) for

Table I. Mapping for (4,3) SPC code

i	natural mapping	parity bits
0	000	0
1	001	1
2	010	1
3	011	0
4	100	1
5	101	0
6	110	0
7	111	1

each of the codewords in the code. As it is given in (4.15) and (4.16), the probability of a codeword error depends on the positions that the two codewords differ. Since we are looking at linear block codes, each error pattern corresponds to a codeword in the code. Therefore to calculate MSE, each codeword can be treated as an error pattern and the terms of the MSE distortion expression can be grouped in terms of these $l = 1, 2, \ldots, 2^K - 1$ error patterns. When the error pattern is fixed as the l-th one in (2.7) and all the terms are grouped according to the error pattern, $C_l = \sum_{i,j} \pi_i (i-j)^2$ can be calculated as follows: sum up all $(i-j)^2$ corresponding to adding this error pattern to the i-th codeword which results in the j-th codeword. Here i and j are the transmitted and received codewords respectively. This method of grouping the

terms with the same error pattern leads to

$$\mathrm{D} = \sum_{l=1}^{2^K-1} C_l P_l \tag{4.17}$$

The following MSE expressions for (3,2),(4,3) and (5,4) single parity check codes are found this way. The coefficients of $\mathcal{Q}$ functions are given in Table II.

The MSE expression can be found as for (3,2) SPC code:

$$MSE = B_1\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_3)}{N_0}}\Big) + B_2\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_3)}{N_0}}\Big) + B_3\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2)}{N_0}}\Big) \tag{4.18}$$

for (4,3) SPC code:

$$\begin{aligned} MSE = {} & B_1\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_3+a_4)}{N_0}}\Big) + B_2\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_4)}{N_0}}\Big) + B_3\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_3)}{N_0}}\Big) \\ & + B_4\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_4)}{N_0}}\Big) + B_5\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_3)}{N_0}}\Big) + B_6\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2)}{N_0}}\Big) \\ & + B_7\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_3+a_4)}{N_0}}\Big) \end{aligned} \tag{4.19}$$

Table II. Coefficients for the MSE expressions

Input Weight	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}
2	1	4	5	-	-	-	-	-	-	-	-	-	-	-	-
3	1	4	5	16	17	20	21	-	-	-	-	-	-	-	-
4	1	4	5	16	17	20	21	64	65	68	69	80	81	84	85

for (5,4) SPC code:

$$
\begin{aligned}
MSE = {} & B_1\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_4+a_5)}{N_0}}\Big) + B_2\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_3+a_5)}{N_0}}\Big) + B_3\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_3+a_4)}{N_0}}\Big) \\
& + B_4\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_5)}{N_0}}\Big) + B_5\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_4)}{N_0}}\Big) + B_6\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_3)}{N_0}}\Big) \\
& + B_7\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_3+a_4+a_5)}{N_0}}\Big) + B_8\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_5)}{N_0}}\Big) \\
& + B_9\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_4)}{N_0}}\Big) + B_{10}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_3)}{N_0}}\Big) \\
& + B_{11}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_3+a_4+a_5)}{N_0}}\Big) + B_{12}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2)}{N_0}}\Big) \\
& + B_{13}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_4+a_5)}{N_0}}\Big) + B_{14}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_3+a_5)}{N_0}}\Big) \\
& + B_{15}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_3+a_4)}{N_0}}\Big) \qquad (4.20)
\end{aligned}
$$

D. MSE Expression for (7,4) Hamming Code

In Table III, the mapping of the Hamming codewords are given. The MSE expression

Table III. The mapping of (7,4) Hamming codewords

i	natural mapping	parity bits
0	0000	000
1	0001	011
2	0010	110
3	0011	101
4	0100	111
5	0101	100
6	0110	001
7	0111	010
8	1000	101
9	1001	110
10	1010	011
11	1011	000
12	1100	010
13	1101	001
14	1110	100
15	1111	111

can be derived as

$$
\begin{aligned}
MSE = {} & B_1\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_4+a_6+a_7)}{N_0}}\Big) + B_2\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_3+a_5+a_6)}{N_0}}\Big) \\
& + B_3\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_3+a_4+a_5+a_7)}{N_0}}\Big) + B_4\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_5+a_6+a_7)}{N_0}}\Big) \\
& + B_5\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_4+a_5)}{N_0}}\Big) + B_6\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_3+a_7)}{N_0}}\Big) \\
& + B_7\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_3+a_4+a_6)}{N_0}}\Big) + B_8\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_5+a_7)}{N_0}}\Big) \\
& + B_9\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_4+a_5+a_6)}{N_0}}\Big) + B_{10}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_3+a_6+a_7)}{N_0}}\Big) \\
& + B_{11}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_3+a_4)}{N_0}}\Big) + B_{12}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_6)}{N_0}}\Big) \\
& + B_{13}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_4+a_7)}{N_0}}\Big) + B_{14}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_3+a_5)}{N_0}}\Big) \\
& + B_{15}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_3+a_4+a_5+a_6+a_7)}{N_0}}\Big) \qquad (4.21)
\end{aligned}
$$

Note that the coefficients do not depend on the code but rather on the mapping of information bits. This is due to the fact that only the mapping of information bits determine the coefficients $\sum_{i,j}(i-j)^2$ for a given error-codeword. The expression in (4.21) is optimized by using the DE optimization. As it can be seen from (4.21), as the number of codewords increases the MSE expression will become very untractable. (4.21) can be well approximated by taking only the codewords with minimum Hamming distance, d_{min} into account. Therefore, from (4.21) only the codewords with

d_{min} are taken into account. Then the simplified expression becomes

$$\begin{aligned} MSE = {} & B_1\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_4+a_6+a_7)}{N_0}}\Big) + B_2\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_3+a_5+a_6)}{N_0}}\Big) \\ & + B_5\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_4+a_5)}{N_0}}\Big) + B_6\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_2+a_3+a_7)}{N_0}}\Big) \\ & + B_8\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_5+a_7)}{N_0}}\Big) + B_{11}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_3+a_4)}{N_0}}\Big) \\ & + B_{12}\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+a_2+a_6)}{N_0}}\Big) \end{aligned} \tag{4.22}$$

(4.22) is optimized using DE.

E. Results

1. Results for SPC Codes

Results for the coded case with (3,2), (4,3) and (5,4) SPC codes are given in this section. These include the plots of MSE obtained by using the differential evolution method and plots of the results from the simulation of the actual systems. The simulations are done by using the power profiles determined by optimizing the MSE expressions given in (4.18), (4.19) and (4.20) by using differential evolution. The plots of these power profiles from the optimization of (4.18), (4.19) and (4.20) are given in Figure 15, Figure 16 and Figure 17 respectively. As can be seen from these graphs, as SNR increases the power profile approaches the uniform power profile.

Plots of MSE obtained from the differential evolution and MSE obtained from the simulation of the actual system are given in Figure 18, Figure 19 and Figure 20. As can be seen from these results, the MSE expressions are upper bounds on the actual MSE values obtained from the simulation of the system. This upper bound is very loose at the lower SNRs. It becomes tighter as the SNR increases. For (3,2)

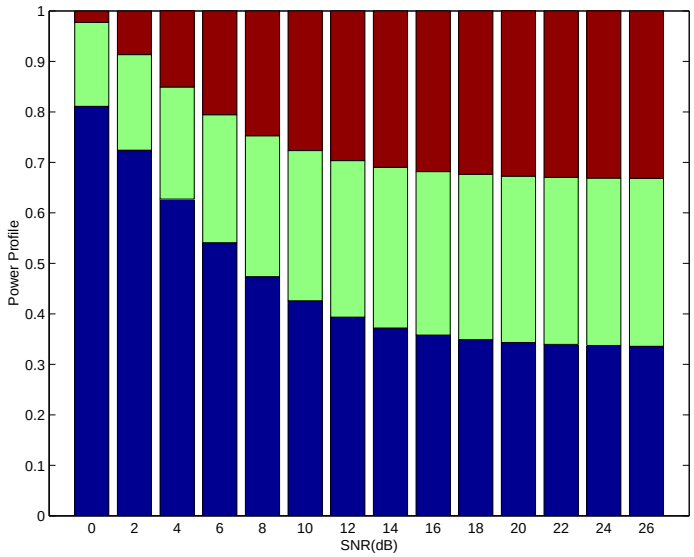

Fig. 15. Power profile for (3,2) SPC code

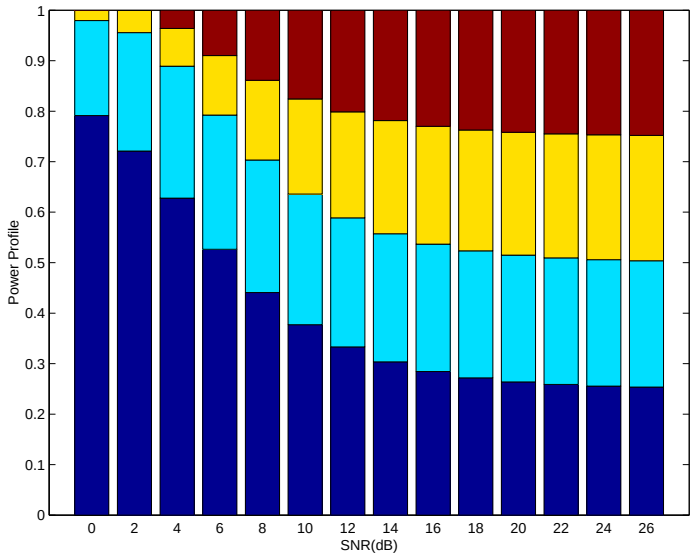

Fig. 16. Power profile for (4,3) SPC code

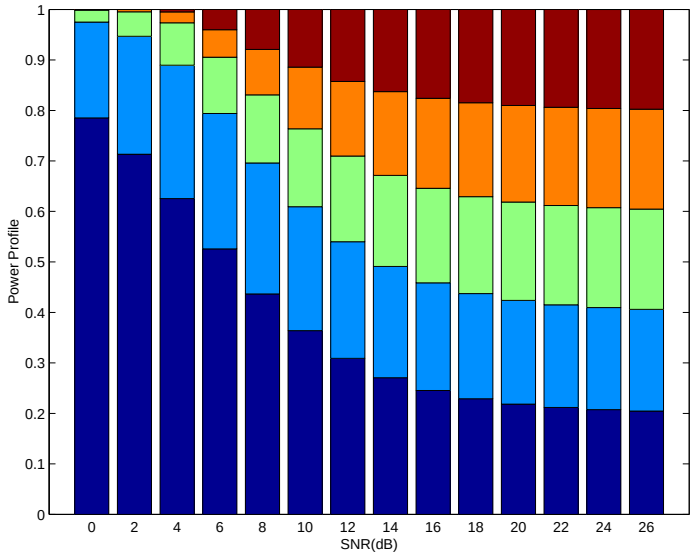

Fig. 17. Power profile for (5,4) SPC code

SPC code, it can be seen from Figure 18 that the upper bound is very close to the actual MSE found from the simulations at 10 dB. For (4,3) and (5,4) SPC codes, from Figure 19 and Figure 20 it is seen that the MSE values are very close to each other at 12 dB. For SNRs higher than 12 dB, the simulation results are hard to obtain since the MSE hence the probability of bit error are very small. Beyond these SNRs, it appears that the MSE expressions are very close to the actual MSE.

In order to see how much MSE gain is obtained by using the optimum power profile from differential evolution, MSE gain graphs are plotted. In the MSE gain plots, MSE gain is calculated from the MSE expressions and from the results of the simulations. For (3,2), (4,3) and (5,4) SPC codes the MSE gain graphs are shown in Figure 21, Figure 22 and Figure 23 respectively. As can be seen from these graphs, the power profiles obtained from differential evolution give reasonable gain in MSE in dB over the uniform power profile. The MSE expressions are upper bounds on the

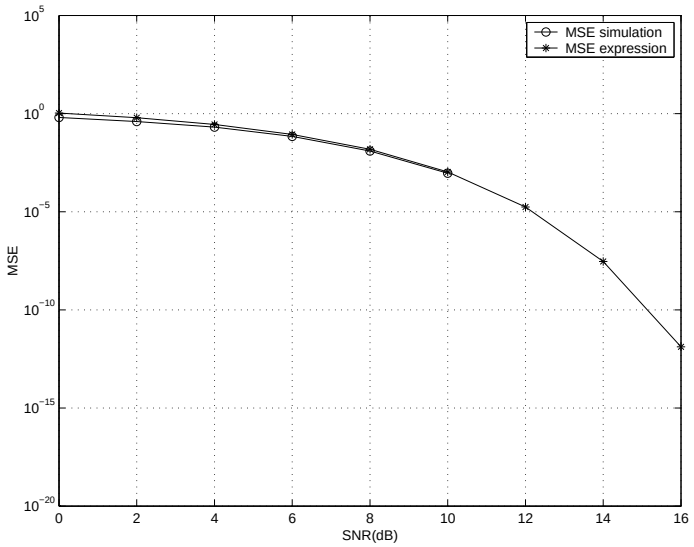

Fig. 18. MSE from expression (4.18) and simulation of the system for (3,2) SPC code

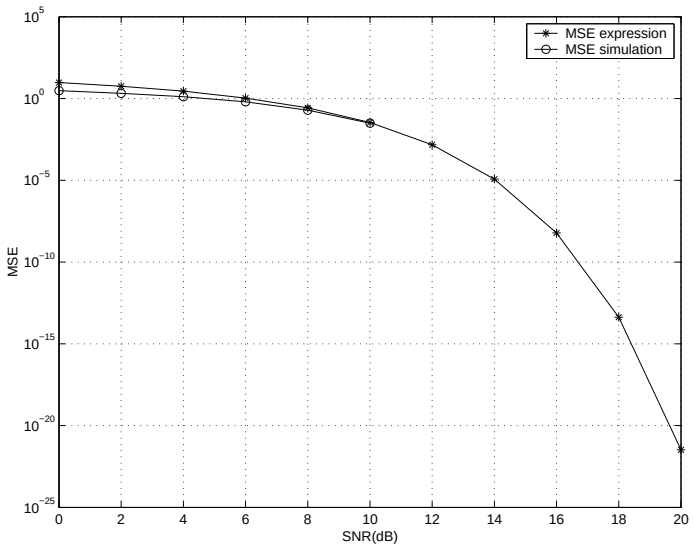

Fig. 19. MSE from expression (4.19) and simulation of the system for (4,3) SPC code

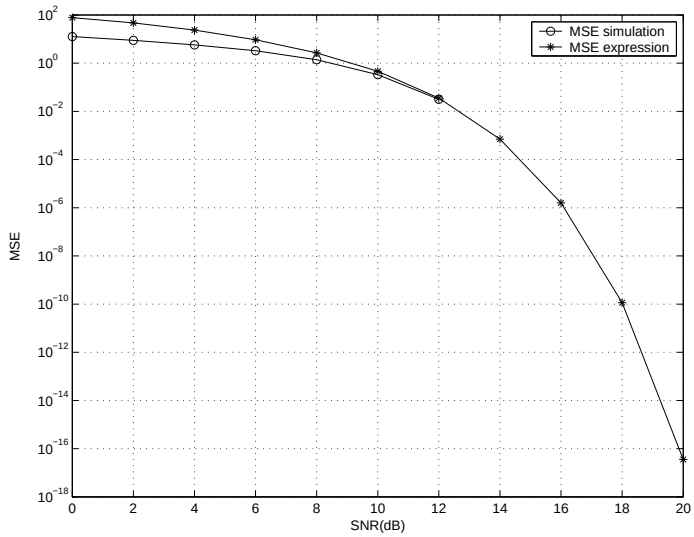

Fig. 20. MSE from expression (4.20) and simulation of the system for (5,4) SPC code

actual MSE of the system which get tighter as SNR increases. In all the MSE gain graphs, MSE gain obtained from simulations approaches MSE gain obtained from the expressions as SNR increases.

As the number of codewords in the code increases, the number of terms in the MSE expression also increases and the MSE expression becomes intractable. A simplification to the MSE expression is to use the d_{min} codewords. This is a good simplification for finding the power profiles since the number of terms is less than or equal to the number of terms in the complete MSE expression and this is particularly advantageous as the length of the code increases. For (3,2) SPC code, all the codewords have weight 2 except for the all zero codeword. Therefore, looking at only the d_{min} codewords does not simplify the expressions. For (4,3), there is only one term which can be omitted from the MSE expression. This is the last term in (4.19). Since $\sum_{i=1}^{4} a_i = 1$, this term does not make any difference in the optimization problem. The MSE plot for (4,3) SPC code with only the d_{min} weight codewords is the same as the

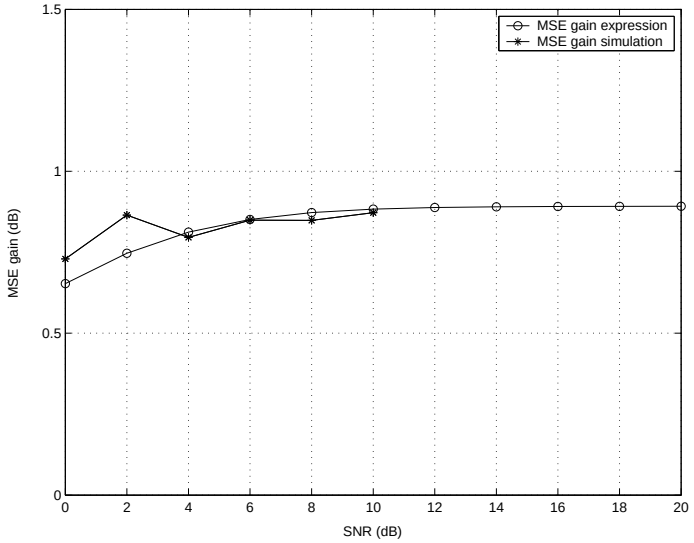

Fig. 21. MSE gain obtained from MSE expression for (3,2) SPC code and from the simulations

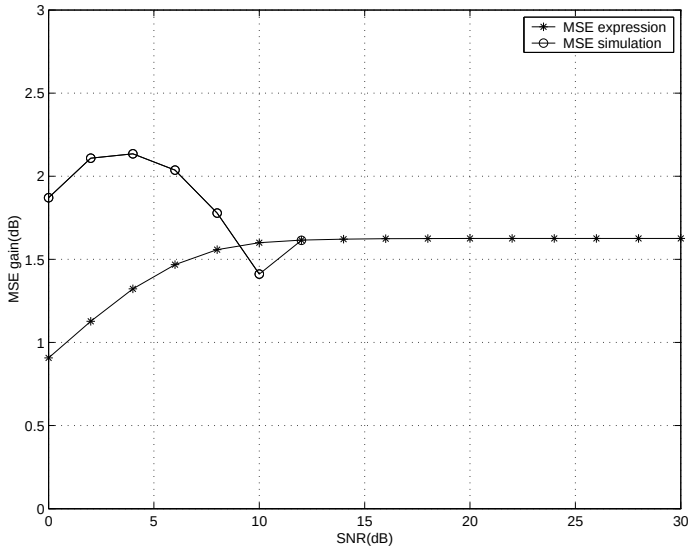

Fig. 22. MSE gain obtained from MSE expression for (4,3) SPC code and from simulations

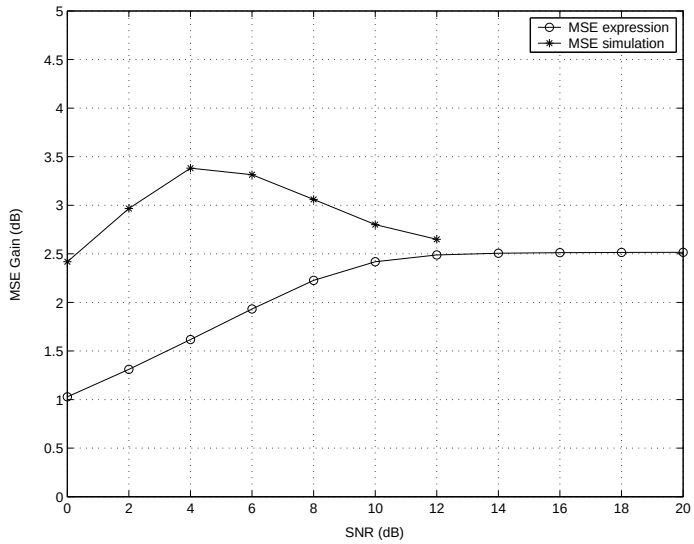

Fig. 23. MSE gain obtained from MSE expression for (5,4) SPC code and from simulations

MSE plot given in Figure 19 and the MSE gain is the same as the plot in Figure 22. For (5,4) code, MSE gain plot is shown in Figure 23. When only the d_{min} codewords are considered in the MSE expression, the power profile that is found is close to the power profile found from the optimization of (4.20). Actually as SNR increases, the power profile from the optimization of the expression with d_{min} terms gets even more close to the power profile from the optimization of (4.20). This power profile is given in Figure 24. The MSE gain plot for (5,4) is nearly the same as the plot shown in Figure 23 so it is not shown again. This suggests that d_{min} approximation is a very robust one in spite of being much simpler.

2. Results for (7,4) Hamming Code

Similar results for the Hamming code is obtained for (7,4) Hamming code by optimizing the expression given in (4.21). The power profile obtained from the optimization

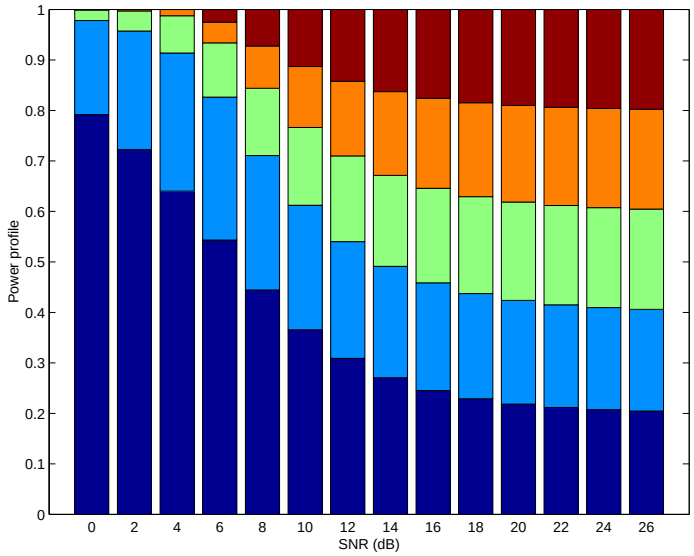

Fig. 24. Power profile for (5,4) SPC code from MSE expression with d_{min} codewords

of (4.21) is plotted in Figure 25.

The system is simulated with the optimum power profile given in Figure 25. The actual MSE found from the simulation of the system is plotted with the MSE obtained from the optimization of (4.21) in Figure 26.

As can be seen from Figure 26, as SNR increases the upper bound determined from the expression given in (4.21) becomes a tighter bound on the actual MSE of the system. MSE gain of the system is also plotted in Figure 27 against the MSE gain determined from expression (4.21). Using a power profile different from the uniform power profile on the simulation of the system gives a gain up to 3 dB.

For (7,4) Hamming code, the MSE expression (4.21) has 15 terms. In order to simplify (4.21) only the codewords with weight d_{min} are taken into account and a simpler expression is optimized. This expression has only 7 terms as it is given in (4.22). The power profile obtained from the optimization of (4.22) is plotted in Figure 28. It can be seen that this power profile is a close approximation to the power

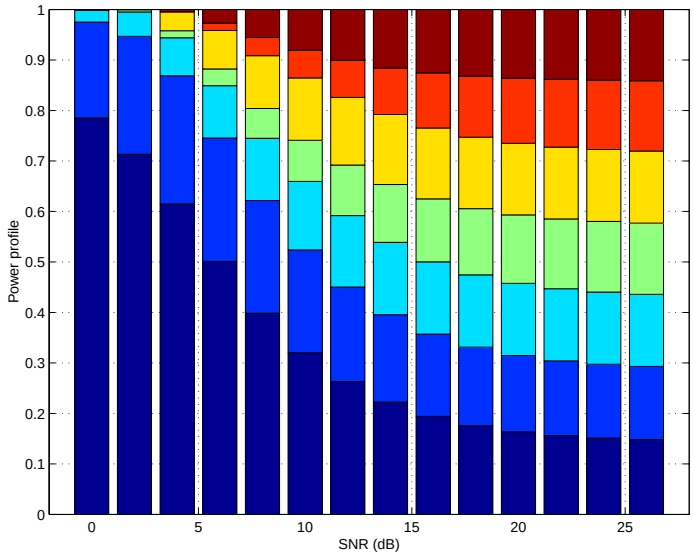

Fig. 25. Power profile for (7,4) Hamming code

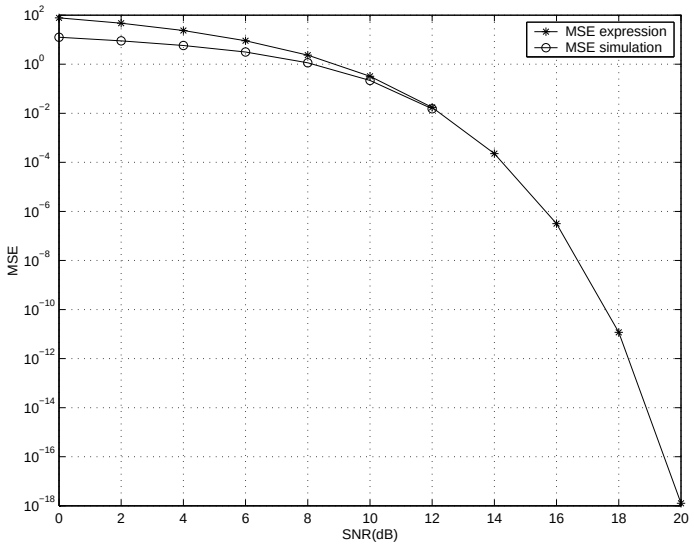

Fig. 26. MSE from MSE expression for (7,4) Hamming code and simulation of the system

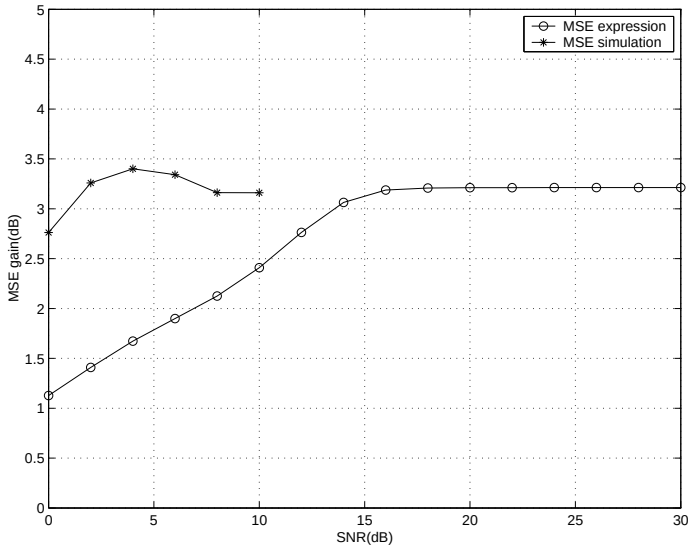

Fig. 27. MSE gain from MSE expression for (7,4) Hamming code and simulation of the system over the uniform power profile

profile given in Figure 25 for SNRs higher than 10 dB. This is due to the fact that as SNR increases the MSE terms with the d_{min} codewords contribute more to the total MSE than the other terms. The MSE obtained from optimizing (4.22) are plotted in Figure 29 with the MSE obtained from the simulation of the actual system.

As can be seen from Figure 29, (4.22) is an upper bound on the actual MSE obtained from the simulation for low SNRs. As SNR increases, the bound becomes tighter and it is very close to the actual value at 12 dB. MSE gain over the uniform power profile is given in Figure 30. It can be seen that the MSE gain obtained from using d_{min} error codewords is around 3 dB for SNRs around 10 dB. This shows that using only the error codewords with weight d_{min} gives reasonable MSE gain over the uniform power profile.

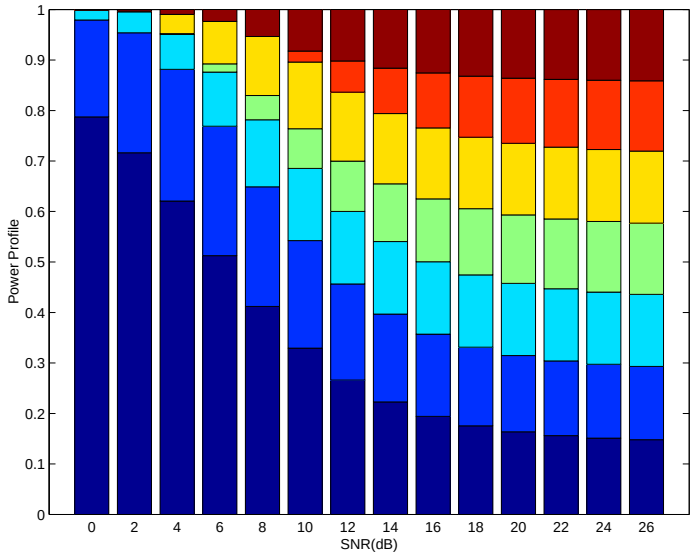

Fig. 28. Power profile for (7,4) Hamming code from MSE expression with d_{min} codewords

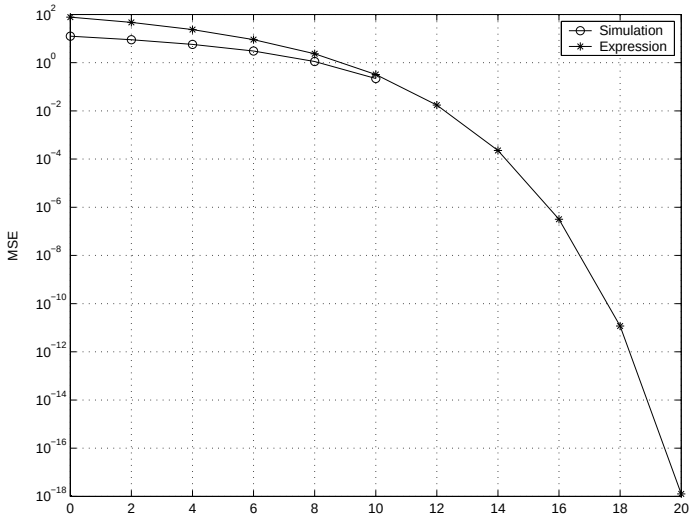

Fig. 29. MSE from bounding expression with d_{min} codewords and simulation results for the system

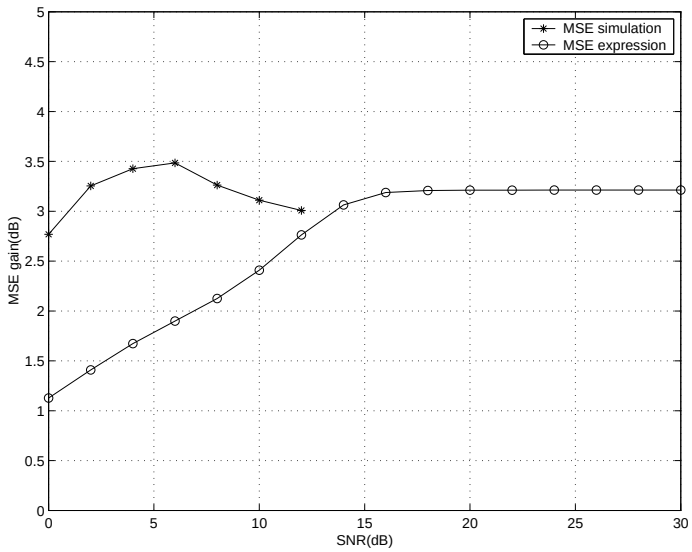

Fig. 30. MSE gain of MSE expression with d_{min} codewords over the uniform power profile

CHAPTER V

GENERALIZATION OF MSE EXPRESSION TO ALL LINEAR BLOCK CODES

In the previous section, we discussed how to derive MSE expressions for SPC and Hamming codes and looked into a possible way to simplify the MSE expressions. In this section, we will try to generalize and simplify the MSE expressions for the coded case further. One possible way to achieve this end is to use the input-output weight enumerating function of the code. We also assume that the parity bits have the same energy and all the information bits have the same energy. At the encoder the mapping is assumed to be natural binary. The approximations made to the MSE expression give a way to find the power profiles without the knowledge of all the codewords in the code. We will show that with this method you can still get reasonable MSE gain over uniform power profile.

A. Approximation to the General MSE Expression

As it is given in (4.15) and (4.16), the probability of a codeword error depend on the positions that the two codewords differ. Since we are looking at linear block codes, each error pattern corresponds to a codeword in the code. Therefore to calculate (2.7), each codeword can be treated as an error pattern and the terms of the MSE distortion expression can be grouped in terms of these $l = 1, 2, \ldots, 2^K - 1$ error patterns. This method of grouping the terms with the same error pattern leads to

$$\mathrm{D} = \sum_{l=1}^{2^K-1} C_l P_l \tag{5.1}$$

Now, assume that a_1 is the energy of the information bits (they all have same energy) and a_2 is the energy of each of the parity bits (same energy). Further, the input-

output weight enumerating function (IOWEF) for a (N,K) linear block code can be written as

$$A(X,Y) = \sum_{b=1}^{n} \sum_{a=1}^{k} A_{a,b} X^a Y^b \tag{5.2}$$

where $A_{a,b}$ is the number of codewords in the code with an output weight b that correspond to an input weight a. The IOWEF of a given code can be easily obtained.

(4.16) can be easily written in terms of the input and the output weight. Let's assume that for each error pattern level m (which corresponds to one of 2^K codewords in the code), n_{mi} is the input weight and n_{mo} is the output weight of the codeword. Since it is a systematic code, n_{mi} represents the number of information bits that have the value 1 and $n_{mo} - n_{mi}$ is the number of parity bits that have the value 1. Then (4.16) can be expressed as

$$\mathcal{Q}\Big(\sqrt{\frac{2E_s(n_{mi}a_1 + (n_{mo} - n_{mi})a_2)}{N_0}}\Big) \tag{5.3}$$

As stated earlier, there are $l = 1, 2, \ldots, 2^K - 1$ error patterns for the case with distinct energy for each of the bits. If this requirement is relaxed and information bits have the same energy and parity bits have the same energy, then the number of distinct $\Pr[j|i]$ terms further comes down to the number of distinct terms in the IOWEF. This is so because with this energy constraint, only the input and output weights of a codeword are significant. The coefficients of the new expression C'_h with a given input weight and output weight is then the sum of the coefficients of error patterns which have these parameters. That is, (5.1) reduces to:

$$D = \sum_{h=1}^{H} C'_h P'_h \tag{5.4}$$

where H is the number of distinct terms in IOWEF and P'_h is of a similar form as

(5.3). However, looking at the coefficients of these distinct terms (definition of C_l) carefully, it becomes evident that they are dependent only on the information bits or in other words the mapping of the quantizer. We will group all the error patterns which have an input weight a and different output weights. Assume that there are n_a distinct error patterns of input weight a in the IOWEF. Then the distortion can be approximated as:

$$D \approx \sum_{a=1}^{k} C^{(a)} (\sum_{t=1}^{n_a} A_{a,b_t} P'_{h_t})$$
$$C^{(a)} = \frac{\sum_{t=1}^{n_a} C'_{h_t}}{\sum_{t=1}^{n_a} A_{a,b_t}} \tag{5.5}$$

where b_ts correspond to different output weights possible for the same input weight a and P'_{h_t}s correspond to the probability of error expression for error patterns with those output weights and input weight a. $C^{(a)}$ is the sum of the coefficients C_l of error patterns (codewords) of input weight a divided by the total number of error patterns of input weight a. For mappings such as natural binary mapping, $C^{(a)}$ is a sequence with some good properties which make them easily determinable for any input length k.

B. Coefficients for Natural Binary Mapping

It is known that the mapping at the transmitter is natural binary. To calculate the coefficients without going through all the codewords, a more general way is sought. Since all the information bits have the same energy, this gives us a good way to find an approximate MSE expression which can be generalized. To calculate the coefficients, $C^{(a)}$, the following can be done. Table IV shows how the coefficients relate to the input weight. In this table, a row represents a fixed K which is the number of information bits, and a column represents a fixed t which is the input weight(number

Table IV. Coefficients from natural binary mapping for any (N,K) code

K	input weight 1	input weight 2	input weight 3	input weight 4
1	1	-	-	-
2	$1[1^2+2^2]$	$\frac{1^2+3^2}{2}$	-	-
3	$1[1^2+2^2+4^2]$	$\frac{1^2+3^2}{2}[1^2+2^2]+\frac{5^2+3^2}{2}$	$\frac{1^2+3^2+5^2+7^2}{4}$	-
4	$1[1^2+2^2$ $+4^2+8^2]$	$\frac{1^2+3^2}{2}[1^2+2^2+4^2]$ $+\frac{5^2+3^2}{2}[1^2+2^2]+\frac{7^2+9^2}{2}$	$\frac{1^2+3^2+5^2+7^2}{4}[1^2+2^2]$ $+\frac{11^2+9^2+7^2+5^2}{4}+\frac{13^2+11^2+3^2+5^2}{4}$	$\frac{1^2+3^2+5^2+7^2+9^2+11^2+13^2+15^2}{8}$

of 1s in the information bits). For a fixed input weight, as the number of information bits increases there is a relationship between the subsequent coefficients. That is, the coefficients of a fixed input weight t for K information bits is related to the coefficients of the same input weight t for $K-1$ information bits and input weight $t-1$ for $K-1$ input weight. A input weight t codeword with K information bits can result from

1. Appending 0s either to the left (more significant side) or to the right (less significant side) of a weight t codeword with information bits $K-1$.

2. Appending a 1 to the right (less significant side) of the weight $t-1$ codeword with information bits $K-1$.

Using these facts, for any length K information bits the coefficients can be calculated for any fixed given input weight , t where $t = 1, 2, \ldots, K$. Once the IOWEF of a code is also known, these can be put together in (5.5) to obtain the approximate expression for MSE distortion.

The coefficients can be found by using a simple algorithm. Finding the coefficients for a certain length mapping requires calculation of all the coefficients starting from length 1. In this algorithm, the input weight is fixed and the length of the mapping varies. You can generate a matrix $\mathcal{B}$ which has the known length of the mapping (number of information bits) as its rows and the input weight as its columns. In this algorithm let ii represent a row of the matrix and jj represent a column of the matrix and K the fixed length of the mapping for which the coefficients need to be found.

1. Initialize a coefficient matrix of K rows, K columns, $\mathcal{B}$ and set all the elements to 0. Rows correspond to number of information bits and columns to weight of the word.

2. Initialize $B(1,1) = 1$.

3. For $jj = 2$ to K, $B(jj, 1) = 2^2 * B(jj-1, 1) + 1$ (wt. 1 coefficients) End Loop

4. For $ii = 2$ to K, Initialize wt_{prev} to wt_{pres} and wt_{pres} to empty set. Equate $numc = 1$ (number of different coefficients) and $num_{prev} = 2^{ii-2}$

   (a) for $jj = ii$ to K, Set $B(jj, ii) = B(jj-1, ii) * 2^2$ and $num_{pres} = 2^{jj-1}$

      i. for $kk = 1$ to $numc$, $wt_{pres} = [wt_{pres} num_{pres} - wt_{prev}((kk-1) * num_{prev} + 1 : kk * num_{prev}) num_{pres} + wt_{prev}((kk-1) * num_{prev} + 1 : kk * num_{prev})]$ Endloop

      ii. for $kk = 1$ to $length(wt_{pres})/(2 * num_{prev})$, $B(jj, ii) = B(jj, ii) + sum([wt_{pres}((kk-1)*2*num_{prev}+1 : kk*2*num_{prev})]^2)/(num_{prev}*2)$ Endloop

   (b) $numc = numc + ii - 2$ End Loop

5. End Loop

In the next section, the above explained method is used and explained in detail for (7,4) Hamming code and the results are given for (3,2), (4,3) and (5,4) SPC code as well as the Hamming code.

C. Approximate Expression for (7,4) Hamming Code

In Table III the mapping of the Hamming codewords are given. The input-output weight enumerating function for (7,4) Hamming code can be written as

$$A(X, Y) = 3XY^3 + XY^4 + 3X^2Y^4 + 3X^2Y^3 + X^3Y^3 + 3X^3Y^4 + X^4Y^7 \qquad (5.6)$$

From (5.6), the number of codewords with a given input-output weight is known. If the exact coefficients are calculated by going through all the different input and output

weight codewords, the distortion expression can be written as a sum of following terms:

1. $A_{1,3} = 3$

$$(1^2 + 2^2 + 8^2)\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1 + 2a_2)}{N_0}}\Big) \tag{5.7}$$

2. $A_{1,4} = 1$

$$4^2\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1 + 3a_2)}{N_0}}\Big) \tag{5.8}$$

3. $A_{2,3} = 3$

$$\Big(\frac{5^2 + 3^2}{2} + \frac{6^2 + 2^2}{2} + \frac{12^2 + 4^2}{2}\Big)\mathcal{Q}\Big(\sqrt{\frac{2E_s(2a_1 + a_2)}{N_0}}\Big) \tag{5.9}$$

4. $A_{2,4} = 3$

$$\Big(\frac{9^2 + 7^2}{2} + \frac{10^2 + 6^2}{2} + \frac{3^2 + 1^2}{2}\Big)\mathcal{Q}\Big(\sqrt{\frac{2E_s(2a_1 + 2a_2}{N_0}}\Big) \tag{5.10}$$

5. $A_{3,3} = 1$

$$\Big(\frac{11^2 + 9^2 + 7^2 + 5^2}{4}\Big)\mathcal{Q}\Big(\sqrt{\frac{2E_s(3a_1)}{N_0}}\Big) \tag{5.11}$$

6. $A_{3,4} = 3$

$$\Big(\frac{7^2 + 5^2 + 3^2 + 1^2}{4} + \frac{13^2 + 11^2 + 5^2 + 3^2}{4} + \frac{14^2 + 10^2 + 6^2 + 2^2}{4}\Big) \mathcal{Q}\Big(\sqrt{\frac{2E_s(3a_1 + a_2)}{N_0}}\Big) \tag{5.12}$$

7. $A_{4,7} = 1$

$$\Big(\frac{15^2 + 13^2 + 11^2 + 9^2 + 7^2 + 5^2 + 3^2 + 1^2}{8}\Big)\mathcal{Q}\Big(\sqrt{\frac{2E_s(4a_1 + 3a_2)}{N_0}}\Big) \tag{5.13}$$

The approximation discussed earlier to this expression can be written as follows:

$$\text{For input weight1} : C^{(1)}\Big(3\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+2a_2)}{N_0}}\Big)+\mathcal{Q}\Big(\sqrt{\frac{2E_s(a_1+3a_2)}{N_0}}\Big)\Big)$$

$$\text{For input weight 2} : C^{(2)}\Big(3\mathcal{Q}\Big(\sqrt{\frac{2E_s(2a_1+2a_2)}{N_0}}\Big)+3\mathcal{Q}\Big(\sqrt{\frac{2E_s(2a_1+a_2)}{N_0}}\Big)\Big)$$

$$\text{For input weight 3} : C^{(3)}\Big(\mathcal{Q}\Big(\sqrt{\frac{2E_s(3a_1)}{N_0}}\Big)+3\mathcal{Q}\Big(\sqrt{\frac{2E_s(3a_1+a_2)}{N_0}}\Big)\Big)$$

$$\text{For input weight 4} : C^{(4)}\mathcal{Q}\Big(\sqrt{\frac{2E_s(4a_1+3a_2)}{N_0}}\Big) \tag{5.14}$$

where $C^{(1)}$, $C^{(2)}$, $C^{(3)}$, $C^{(4)}$ are coefficients that come from Table IV. These coefficients are the total of the coefficients for the given input. Thus, they need to be averaged over the total number of codewords with the given input weight. Therefore from Table IV

$$\begin{aligned} C^{(1)} &= \frac{(1^2+2^2+4^2+8^2)}{(A_{1,3}+A_{1,4})} \\ &= \frac{(1^2+2^2+4^2+8^2)}{4} \end{aligned} \tag{5.15}$$

$$\begin{aligned} C^{(2)} &= \frac{\Big(\frac{1^2+3^2}{2}[1^2+2^2+4^2]+\frac{5^2+3^2}{2}[1^2+2^2]+\frac{7^2+9^2}{2}\Big)}{(A_{2,3}+A_{2,4})} \\ &= \frac{\Big(\frac{1^2+3^2}{2}[1^2+2^2+4^2]+\frac{5^2+3^2}{2}[1^2+2^2]+\frac{7^2+9^2}{2}\Big)}{6} \end{aligned} \tag{5.16}$$

$$\begin{aligned} C^{(3)} &= \frac{\Big(\frac{1^2+3^2+5^2+7^2}{4}[1^2+2^2]+\frac{11^2+9^2+7^2+5^2}{4}+\frac{13^2+11^2+3^2+5^2}{4}\Big)}{(A_{3,3}+A_{3,4})} \\ &= \frac{\Big(\frac{1^2+3^2+5^2+7^2}{4}[1^2+2^2]+\frac{11^2+9^2+7^2+5^2}{4}+\frac{13^2+11^2+3^2+5^2}{4}\Big)}{4} \end{aligned} \tag{5.17}$$

$$C^{(4)} = \frac{1^2+3^2+5^2+7^2+9^2+11^2+13^2+15^2}{8} \tag{5.18}$$

A simple MATLAB program can be used to generate the coefficients starting with $k = 1$ going up to $k = 4$ for this case by using the algorithm described.

D. Generalization of the MSE Expression

Table IV gives a good way to generalize the approximate expression for all linear block codes. If the IOWEF of the code is known and the assumption that all the information bits have the same energy and parity bits have the same energyis made, then the above described method can be used to get the MSE expression which can be optimized to find the energy profile of the information and parity bits.

E. Results

Results obtained from MSE expression for the (3,2), (4,3) and (5,4) SPC codes with the above method are shown in Figure 31. Since all the information bits are constrained to have the same energy and all the parity bits are constrained to have the same energy, this scheme cannot perform as well as the scheme with power allocation available for each of the information bits.

The power profile graphs are shown in Figure 32, Figure 33 and Figure 34. As it can be seen from these graphs, the power profile converges close to the uniform power profile fast.

Similar results for the (7,4) Hamming code are shown in Figure 35. Figure 36 shows the power profiles of information and parity bits for different SNRs. The magnitude of the gain is smaller compared to the MSE gain that can be obtained by changing the power profiles of each bit.

These system have also been simulated to see how much the actual MSE gain over the uniform power profile for (7,4) Hamming code is. The MSE graph given in

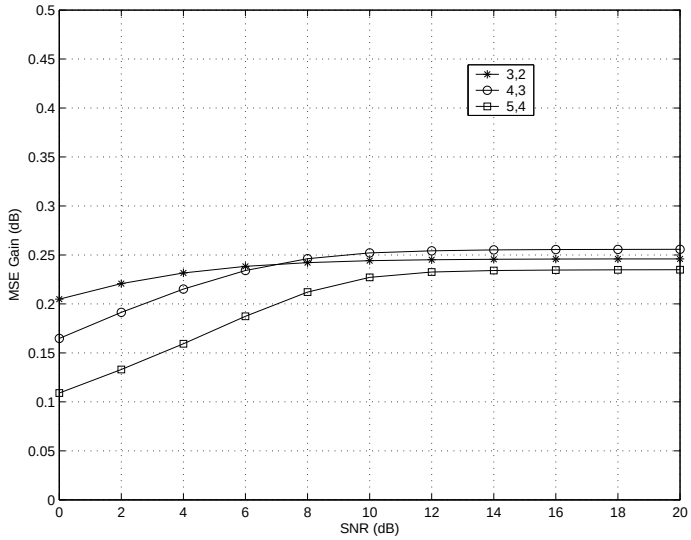

Fig. 31. Optimum MSE found by using the DE method for the SPC coded cases (3,2), (4,3), (5,4)

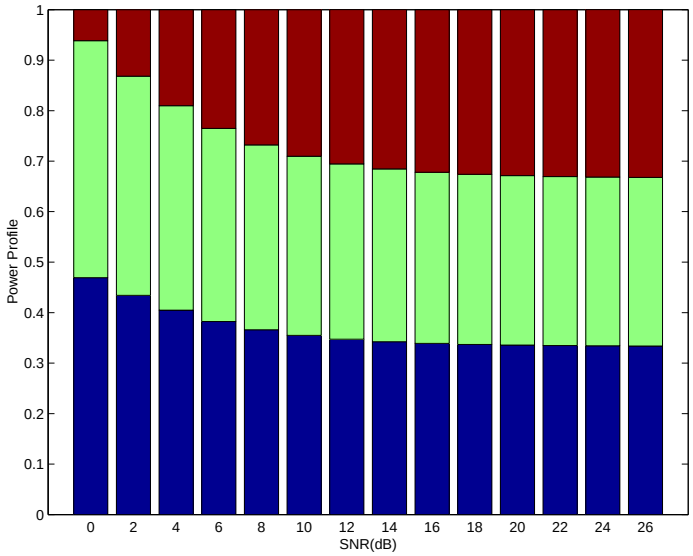

Fig. 32. Power profile for (3,2) SPC code found from DE

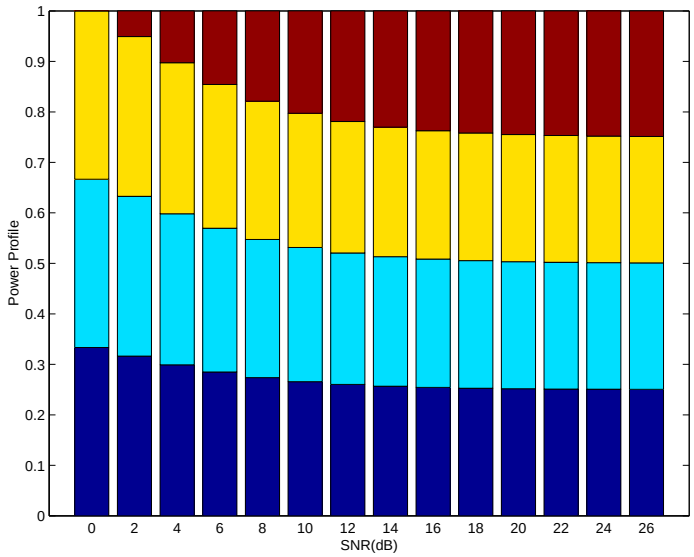

Fig. 33. Power profile for (4,3) SPC code found from DE

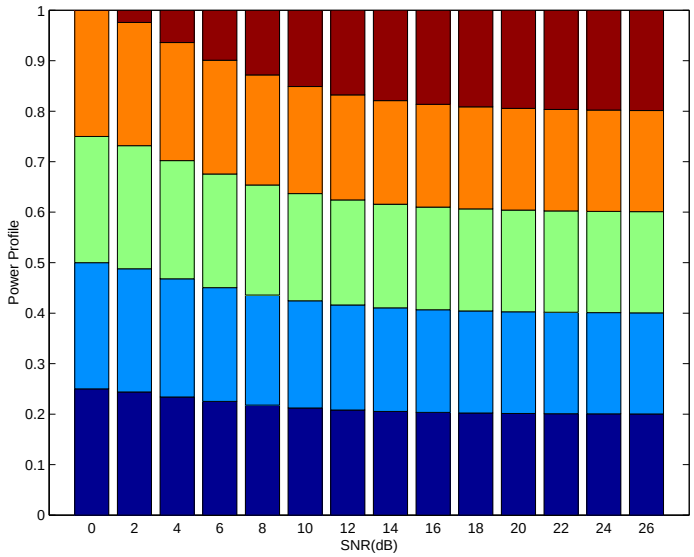

Fig. 34. Power profile for (5,4) SPC code found from DE

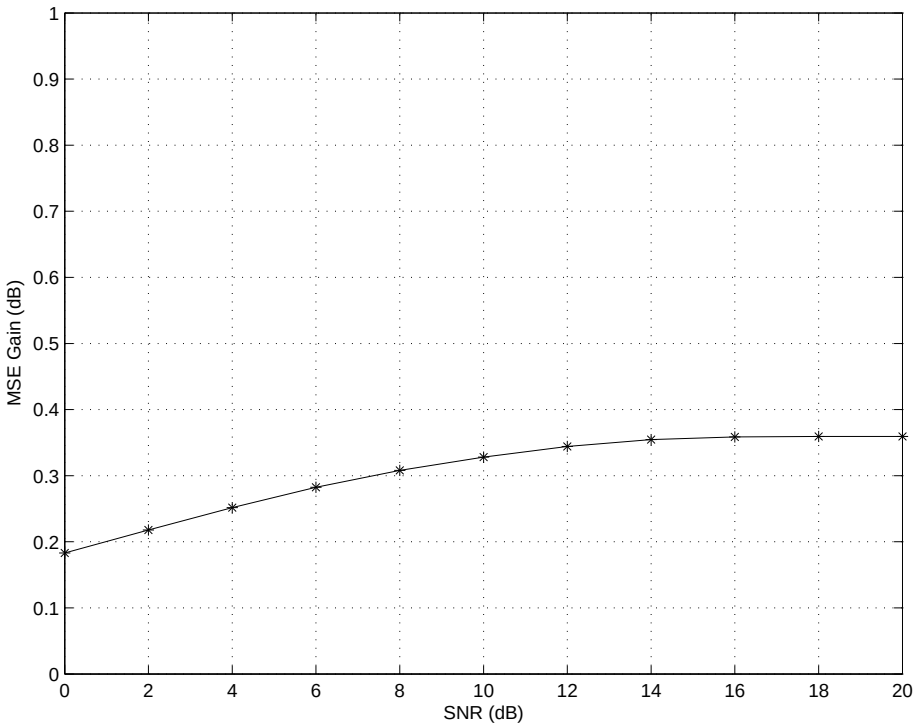

Fig. 35. Optimum MSE found by using the DE method for the (7,4) Hamming coded case

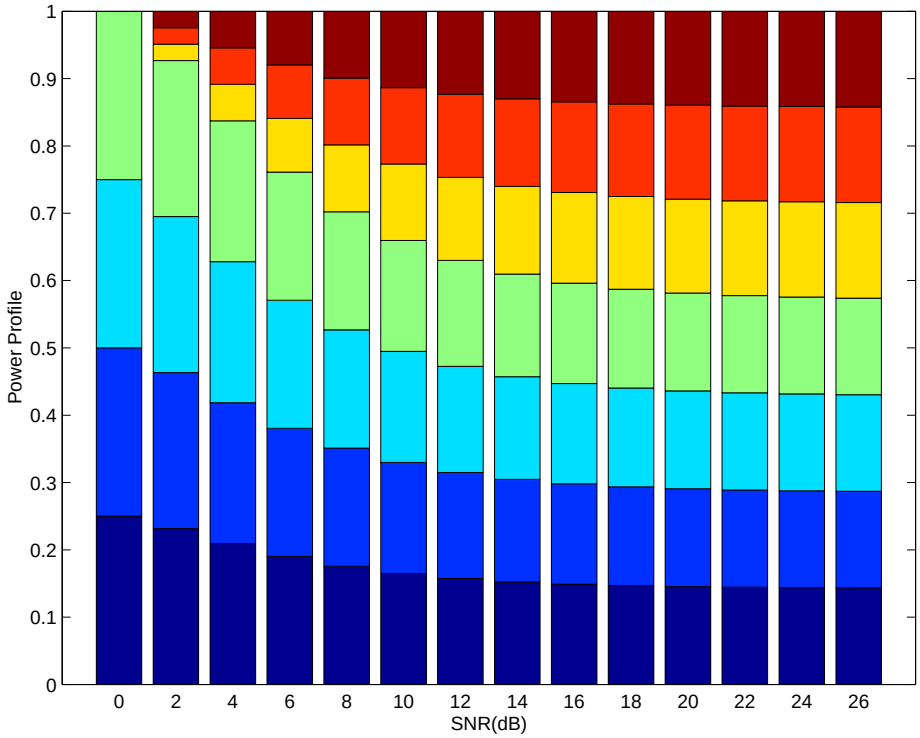

Fig. 36. Optimum power profiles found by using the DE method for the (7,4) Hamming coded case with the approximation

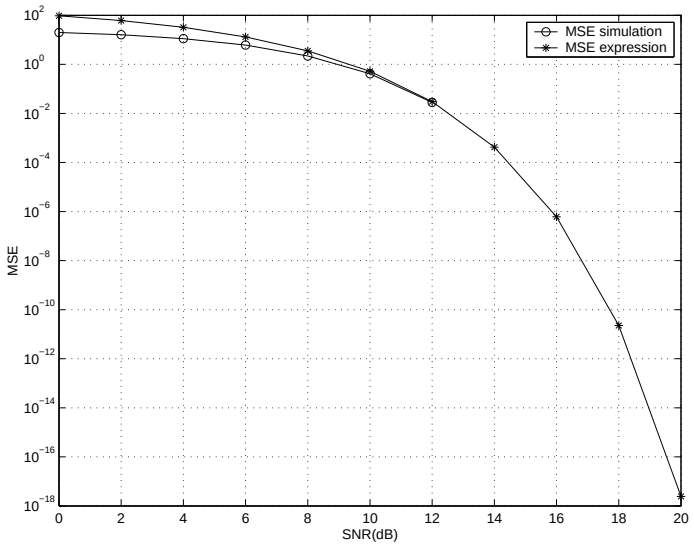

Fig. 37. Actual MSE from the simulation and the MSE from the expression

Figure 37 shows that the actual MSE is very close to the MSE calculated from the expression and as SNR increases it becomes nearly equal. Since the power profile approaches the uniform power profile faster than the cases when all the bits are assigned different energies, MSE of the actual system approaches the MSE found from the expressions faster.

In Figure 38, actual MSE obtained from the simulation of the system for uniform power profile and the MSE obtained from the expression is shown. As can be seen from this figure, the two MSE values are very close for SNRs higher than 10 dB.

The MSE gain of this system is shown in Figure 39. The simulation of the system shows 0.6-0.8 dB for low SNRs and 0.2 dB for higher SNRs (10 dB).

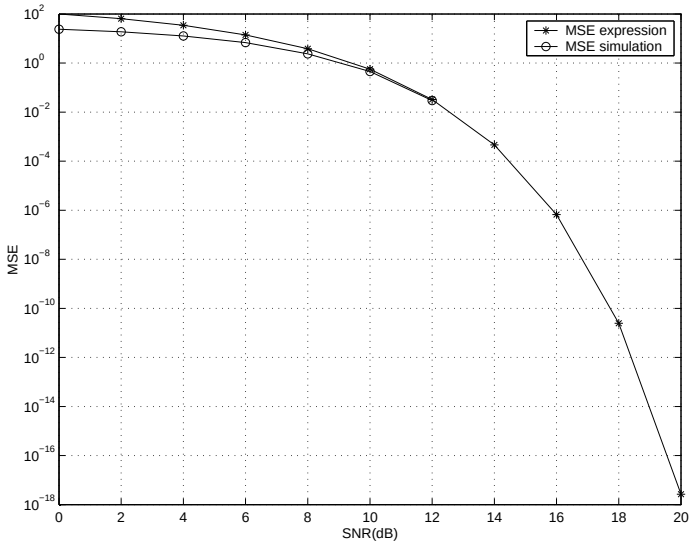

Fig. 38. Actual MSE from the simulation and the MSE from the expression for uniform power profile

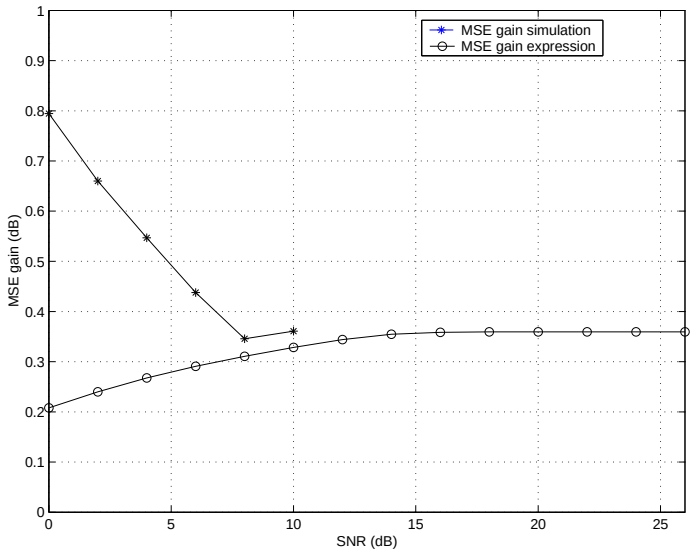

Fig. 39. MSE gain determined from the expression and the simulation of the system for (7,4) Hamming code

CHAPTER VI

CONCLUSION

In this thesis, we studied how to determine the optimum bit power profile in order to minimize the MSE of a basic communication system. At transmitter, the signal generated by the analog signal source was sampled, passed through a uniform quantizer where the signal samples were naturally mapped to 2^k levels and BPSK modulated and transmitted through the AWGN channel. At the receiver, the received signal was demodulated and the transmitted signal was reconstructed. We looked into two cases:

- uncoded
- coded

For the uncoded system, the communication system was the same as described above, but for the coded case, the system was modified by adding channel encoding before BPSK modulation at the transmitter and channel decoding at the receiver after demodulation. The MSE of the system was derived between the output of the quantizer at the transmitter and the reconstructed signal levels at the input of the dequantizer at the receiver for both cases. By using Chernoff bound we were able to obtain a closed form expression for the power profiles which gives minimum MSE. The analytical results for the uncoded case showed that by changing the power of each bit, the MSE of the system can be minimized. This was also shown by using computer-based optimization. Actually for low SNRs (less than 10 dB), the less significant bit is assigned negligible amount of power compared to the most significant bit. This means that for low SNRs using different power allocation for each bit and not transmitting the less significant bits gives nearly optimum MSE for the system. Therefore, for

very noisy channels only the more important bits need to be transmitted with more power instead of transmitting all the bits and still obtain gain in terms of MSE in dB. It was also shown that as SNR increases the power profile approaches the uniform power profile. The MSE gain results obtained from the closed form expression and the computer-based optimization also showed that as SNR increases, a constant gain in dB is achieved even though the power profile approaches uniform. The simulation results could only be obtained up to 12 dB, but these results closely matched the MSE gain obtained from MSE expressions.

For the coded case, it is not possible to obtain an exact expression for MSE. Instead a bound was derived from the soft-decision decoding rule and computer-based optimization was used to see how power was allocated between the information bits and the parity bits. The codes considered in this work were linear block codes such as (3,2), (4,3) and (5,4) SPC codes and (7,4) Hamming code. This work can be extended to longer length codes too. The optimization of the approximate MSE expressions gave power profiles where the parity bits had negligible power for low SNRs (less than 8 dB). For very noisy channels, the information bits need to be protected more than the parity bits. As SNR increases all the power profiles tend to the uniform power profile. The simulations of the actual system with these power profiles showed a positive MSE gain in dB over the uniform power profile in all the codes considered here.

The MSE expressions determined for the coded case had the same number of terms as the codewords in the code. Therefore, as the number of codewords in the code increased, the MSE expression became intractable. Simplifications to the MSE expression were sought. One of them was to consider the terms with d_{min} weight codewords. This gave a good approximation for higher SNRs since the d_{min} weight codewords dominate the probability of bit error for high SNRs. However, it is also

observed that the power profiles obtained from d_{min} expression for lower SNRs are also close to the power profiles obtained from exact MSE expression. We also looked into ways to find a generalized MSE expression where we do not need to go through all the codewords of the code. We assumed that input-output weight enumerating function (IOWEF) of the code is known and that the information bits have the same energy whereas the parity bits also have the same energy. Natural binary mapping was used at the quantizer. These assumptions led to a generalized expression which was used for the SPC codes and the Hamming code. Since all the information bits are allocated the same power and all the parity bits are allocated the same power, the MSE gain of the system decreased drastically compared to allocating different power to each bit, but still gave reasonable gain over the uniform power allocation.

REFERENCES

[1] K.W. Cattermole, *Principles of Pulse Code Modulation*. New York: American Elsevier Publishing Co., 1969.

[2] E. Bedrosian, "Weighted PCM," *IRE Trans. Inform. Theory*, vol. IT-4, pp.45-49, Mar 1958.

[3] R. Bellman and R. Kalaba, "On Weighted PCM and Mean-Square Deviation," *IRE Trans. Inform. Theory*, vol.4, no. 1, pp. 58-59, Mar 1958.

[4] C-E. Sundberg, "Optimum Weighted PCM for Speech Signals," *IEEE Trans. on Communications*, vol. COM-26, no. 6, pp. 872-881, June 1978.

[5] N. Rydbeck and C-E. Sundberg, "Analysis of Digital Errors in Nonlinear PCM Sytems," *IEEE Trans. on Communications*, vol. COM-24, no. 1, pp. 59-65, Jan 1976.

[6] N. C. Griswold, "An Optimized Weighing Algorithm for Variations in PCM Energy Levels,"*IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 1, pp. 180-187, Feb 1983.

[7] B. Masnick and J. Wolf, "On Linear Unequal Error Protection Codes," *IEEE Trans. on Inform. Theory*, vol. 13, no. 4, pp. 600-607, Oct 1967.

[8] A. Shiozaki, "Unequal Error Protection of PCM Signals by Self-Orthogonal Convolutional Codes," *IEEE Trans. on Communications*, vol.37, no. 3, pp. 289-290, Mar 1989.

[9] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, New York: Academic Press, 1982.

[10] J.G. Proakis, *Digital Communications*. Singapore: McGraw-Hill, 2000.

[11] R. Storn and K. Price, "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization, Kluwer Academic Publishers*, vol. 11, pp. 341-359, 1997.

APPENDIX A

CHERNOFF BOUND FOR UNCODED CASE

The problem is:

$$\min_{a_k} \frac{1}{2} \sum_{k=0}^{K-1} 2^{2k} e^{-\mathrm{SNR} a_k} \tag{A.1}$$

subject to:

$$\sum_{k=0}^{K-1} a_k = 1 \tag{A.2}$$

Lagrange multipliers can be applied (A.1). Therefore

$$J = \frac{1}{2} \sum_{k=0}^{K-1} 2^{2k} e^{-SNR a_k} + \lambda \sum_{k=0}^{K-1} a_k \tag{A.3}$$

Taking derivatives with respect to a_k

$$\frac{\partial J}{\partial a_k} = \frac{-1}{2} SNR 2^{2k} e^{-SNR a_k} + \lambda \tag{A.4}$$

From here a_k can be found to be

$$a_k = \frac{1}{SNR} \big(\ln SNR + 2k \ln 2 - \ln 2\lambda \big) \tag{A.5}$$

If this is substituted into the (A.2) equation λ can be found in terms of a_k . Therefore λ comes out to be

$$\lambda = e^{(\frac{-SNR}{K} + \ln SNR + (K-1)\ln 2)} \tag{A.6}$$

Substituting this back to (A.5) gives the solution to the minimization problem as

$$\hat{a_k} = \frac{(2k + 1 - K)\ln(2)}{\mathrm{SNR}} + \frac{1}{K} \quad k = 0, 1, \ldots, K-1 \tag{A.7}$$

APPENDIX B

MATLAB CODE

Here is the main MATLAB code used for this thesis. DE algorithm (devec3.m) has been modified from http://www.icsi.berkeley.edu/ storn/code.html#matl

%Uncoded case

%calculation of ak values by using the iterative method (Chernoff Bound)

%SNR is fixed

$format\ long$

$K = 4$

$for\ ss = 0 : 2 : 30$

$SNR = 10\hat{(}ss/10)$

$K1 = K;$

$flag = 1; c = [1 : 1 : K1]; b = 0;$

$while\ (flag == 1) K1 = length(c); b = b + 1;$

$for\ k = length(c) : -1 : 1$

$ak(c(k)) = ((2 * (k - 1) + 1 - K1) * log(2)/(SNR)) + (1/K1);$

$end;$

$c1 = find(ak > 0);$

$if\ (length(c1) == K1)$

$break;$

$else$

$ak(b) = 0;$

$end;$

$c2 = [b + 1 : K];$

$c = [\,]; c = c2;$

$end;$

$MSE(ss+1) = 0.5 * dot(2.^{(}2 * [0:K-1]), exp(-SNR * ak))$

$AK(ss+1,:) = ak$

$end;$

%Example of DE optimization for Hamming code for the approximate MSE expression

%DE optimization for Hamming code

format long;

$rand('state', sum(100 * clock));$

$K = 7;$

$XL = zeros(1,K); XU = ones(1,K);$

$fprintf('EXACT\ HAMMING\ EXPRESSION\ DE');$

$X = ones(1,K)/K$

% Initialization and run of differential evolution optimizer.

% A simpler version with fewer explicit parameters is in run0.m

%

% Here for Rosenbrock's function

% Change relevant entries to adapt to your personal applications

%

% The file ofunc.m must also be changed

% to return the objective function

%

% VTR "Value To Reach" (stop when ofunc < VTR)

% D number of parameters of the objective function

$D = K;$

% XVmin,XVmax vector of lower and bounds of initial population

% the algorithm seems to work well only if [XVmin,XVmax]

% covers the region where the global minimum is expected

% *** note: these are no bound constraints!! ***

$XVmin = [0, 0];$

$XVmax = [1, 1];$

% y problem data vector (remains fixed during optimization)

% NP number of population members

$NP = 30;$

% itermax maximum number of iterations (generations)

$itermax = 1.e3;$

% F DE-stepsize F ex [0, 2]

$F = 0.8;$

% CR crossover probability constant ex [0, 1]

$CR = 0.8;$

$\%strategy$

$\%1 - - > DE/best/1/exp$

$\%2 - - > DE/rand/1/exp$

$\%3 - - > DE/rand - to - best/1/exp$

$\%4 - - > DE/best/2/exp$

$\%5 - - > DE/rand/2/exp$

$\%6 - - > DE/best/1/bin$

$\%7 - - > DE/rand/1/bin$

$\%8 - - > DE/rand - to - best/1/bin$

$\%9 - - > DE/best/2/bin$

$\%else\ DE/rand/2/bin$

$strategy = 7$

% refresh intermediate output will be produced after "refresh"

% iterations. No intermediate output will be produced

% if refresh is < 1

$refresh = 1000;$

$MSE = zeros(1, 21);$

$for\ ss = 0 : 2 : 40$

$MSE(ss + 1) = inf;$

$AK(ss + 1, :) = zeros(1, K); VTR = 1.e - 1000;$

$y = [K\ ss];$

$[p, fval, fiter]= devec3('msehamming\_de1', VTR, D, XL, XU, y, \ldots$

$\ldots, NP, itermax, F, CR, strategy, refresh);$

$if\ (MSE(ss + 1)\ fval)$

$MSE(ss + 1) = fval$

$PP(ss + 1, :) = p$

end

$end;$

MSE, PP

$function\ solution = msehamming\_de1(p, y)$

$K = y(1); ss = y(2);$

$if\ (length(find(p < 0)) + length(find(p > 1)) < 1)$

solution=0.5*[erfc(10$\hat{(}$ss/20)*sqrt(p(4)+p(6)+p(7)))+4*erfc(10$\hat{(}$ss/20)*sqrt(p(3)
+p(5)+p(6)))+16*erfc(10$\hat{(}$ss/20)*sqrt(p(2)+p(5)+p(6)+p(7)))+64*erfc(10$\hat{(}$ss/20)
sqrt(p(1)+p(5)+p(7)))+5*erfc(10$\hat{(}$ss/20)*sqrt(p(3)+p(4)+p(5)+p(7)))+17*erfc

(10^(ss/20)*sqrt(p(2)+p(4)+p(5)))+20*erfc(10^(ss/20)*sqrt(p(2)+p(3)+p(7)))+21 erfc(10^(ss/20)*sqrt(p(2)+p(3)+p(4)+p(6)))+65*erfc(10^(ss/20)*sqrt(p(1)+p(4) +p(5)+p(6)))+68*erfc(10^(ss/20)*sqrt(p(1)+p(3)+p(6)+p(7)))+69*erfc(10^(ss/20) sqrt(p(1)+p(3)+p(4)))+80*erfc(10^(ss/20)*sqrt(p(1)+p(2)+p(6)))+81*erfc(10^ (ss/20)*sqrt(p(1)+p(2)+p(4)+p(7)))+84*erfc(10^(ss/20)*sqrt(p(1)+p(2)+p(3)+p(5)))+ 85*erfc(10^(ss/20)*sqrt(sum(p)))];

else

solution=1.e10;

end;

%simulation of the communication system with near optimum power profiles for Hamming code

%generation of the (7,4) Hamming code

$no\_bits = 1000;$

$quantizer(no\_bits, 7) = 0;$

$for\ ii = 0 : 15$

$bits(ii + 1, 1 : 4) = bitget(ii, 4 : -1 : 1);$

end

$for\ ii = 1 : 16$

$bits(ii, 5) = mod((bits(ii, 1) + bits(ii, 2) + bits(ii, 3)), 2);$

$bits(ii, 6) = mod((bits(ii, 2) + bits(ii, 3) + bits(ii, 4)), 2);$

$bits(ii, 7) = mod((bits(ii, 1) + bits(ii, 2) + bits(ii, 4)), 2);$

end

$codewords = 2 * bits - 1;$

%power allocation for Hamming code

$dummy = PP(1 : 2 : end, :); dummy(find(dummy < 0)) = 0;$ %generation of the random code words $ss = [0 : 2 : 32]; count1 = zeros(1, length(ss));$

```
count2 = zeros(1, length(ss)); count3 = zeros(1, length(ss));
count4 = zeros(1, length(ss)); MSE = zeros(1, length(ss));
for mm = 1 : length(ss)
p = dummy(mm, :);
frame = 0; frame1 = 0;
while   (frame<200)
frame1 = frame1 + 1;
samples = round(15 * rand(1, no_bits));
for ii = 1 : no_bits
quantizer(ii, 1 : 4) = bitget(samples(1, ii), 4 : -1 : 1);
end
for ii = 1 : no_bits
quantizer(ii, 5) = mod((quantizer(ii, 1) + quantizer(ii, 2) + quantizer(ii, 3)), 2);
quantizer(ii, 6) = mod((quantizer(ii, 2) + quantizer(ii, 3) + quantizer(ii, 4)), 2);
quantizer(ii, 7) = mod((quantizer(ii, 1) + quantizer(ii, 2) + quantizer(ii, 4)), 2);
end
quantizer_ = 2 * quantizer - 1;
for ii = 1 : no_bits
transmitted(ii, :) = sqrt(p(1, :)). * quantizer_(ii, :);
end
noise = (sqrt(0.5 * 10^( - ss(mm)/10)) * randn(no_bits, 7));
received = transmitted + noise;
for ii = 1 : no_bits
for jj = 1 : 16
temp(ii, jj) = sum(sqrt(p(1, :)). * codewords(jj, :). * received(ii, :));
%temp1(ii, jj) = sum(sqrt(p(1, :)). * bits(jj, :). * received1(ii, :));
```

```
%temp(ii,jj) = sum(codewords(jj,:).*received(ii,:));
end
end

(C,Y) = max(temp');
for ii = 1 : no_bits
decoded(ii,:) = bits(Y(ii),:);
end
MSE1 = 0;
for ii = 1 : no_bits
MSE1 = MSE1 + (samples(1,ii) - (8 * decoded(ii,1) + 4 * decoded(ii,2) + 2 *
decoded(ii,3) + decoded(ii,4)))^2;
end; MSE1;
MSE(mm) = MSE(mm) + MSE1/no_bits;
count11 = 0; count22 = 0; count33 = 0; count44 = 0;
count11 = length(find(decoded(:,1) = quantizer(:,1)));
count22 = length(find(decoded(:,2) = quantizer(:,2)));
count33 = length(find(decoded(:,3) = quantizer(:,3)));
count44 = length(find(decoded(:,4) = quantizer(:,4)));
if (count11 > 0&count22 > 0&count33 > 0&count44 > 0)
frame = frame + 1;
end;
count1(mm) = count1(mm) + count11;
count2(mm) = count2(mm) + count22;
count3(mm) = count3(mm) + count33;
count4(mm) = count4(mm) + count44;
```

save simhamming_de3 count1 count2 count3 count4 MSE frame frame1 no_bits ss;

end;

$count1(mm) = count1(mm)/(frame1 * no\_bits); count1(mm)$

$count2(mm) = count2(mm)/(frame1 * no\_bits); count2(mm)$

$count3(mm) = count3(mm)/(frame1 * no\_bits); count3(mm)$

$count4(mm) = count4(mm)/(frame1 * no\_bits); count4(mm)$

$MSE(mm) = MSE(mm)/frame1;$

save simhamming_de4 p count1 count2 count3 count4 MSE frame frame1 no_bits ss

end

VITA

Arzu Karaer was born in Istanbul, Turkey. She received her B.S. degree from Istanbul Technical University, Turkey in June 2002. Her permanent address is:

Dervis Pasa Sokak No:10 D:6

Capa ISTANBUL 34280

Turkey. She can also be reached via e-mail at *arzu_karaer@yahoo.com*.

The typist for this thesis was Arzu Karaer.