Coding Techniques for Noisy Channels with Gain and/or Offset Mismatch

Bu, R.

DOI
10.4233/uuid:18ab40eb-5336-4498-a20b-b5d9a1663003

Publication date
2021

Document Version
Final published version

Citation (APA)
Bu, R. (2021). *Coding Techniques for Noisy Channels with Gain and/or Offset Mismatch*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:18ab40eb-5336-4498-a20b-b5d9a1663003

Coding Techniques for Noisy Channels with Gain and/or Offset Mismatch

Coding Techniques for Noisy Channels with Gain and/or Offset Mismatch

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Monday 6 December 2021 at 10:00 o'clock

by

Renfei BU

Master of Engineering in Information and Communication Engineering,
National University of Defense Technology, China,
born in Changsha, Hunan, China.

This dissertation has been approved by the promotors:

Dr.ir. J.H. Weber
Prof.dr.ir. K.I. Aardal

Composition of the doctoral committee:

Rector Magnificus,	chairman
Dr.ir. J.H. Weber,	Delft University of Technology, promotor
Prof.dr.ir. K.I. Aardal,	Delft University of Technology, promotor
Independent members:	
Prof.dr.ir. R.E. Kooij,	Delft University of Technology
Prof.dr.ir. R. Heusdens,	Netherlands Defence Academy
	Delft University of Technology
Prof.dr.ir. L. Van der Perre,	KU Leuven
Prof.dr. K. Cai,	Singapore University of Technology and Design
Prof.dr. D.C. Gijswijt,	Delft University of Technology, reserve member
Other member:	
Dr.ir. K.A. Schouhamer Immink,	Turing Machines Inc.

This research was financially supported by China Scholarship Council (CSC).

Keywords: Coding techniques, Pearson distance, Euclidean distance, maximum likelihood decoding, offset, gain, fading, mismatch.

Printed by: GVO drukkers & vormgevers

Front & Back: Xiumeng Sheng

ISBN 978-94-6332-812-8

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To my family

Contents

SUMMARY

Data transmission is ubiquitous in all walks of life, ranging from basic home and office appliances like compact disc players and hard disk drives to deep space communication. More often than not, the communication and storage channels are *noisy*, and data might be distorted during transmission. However, noise is not the only disturbance during the data transmission, and information can sometimes be seriously distorted by the phenomena of unknown channel *gain or offset (drift) mismatch*. The conventional minimum Euclidean distance based detection, where the receiver picks a codeword from the codebook to minimize the Euclidean distance with the received word, has a poor performance under the gain and/or offset mismatch. Recently, a Pearson distance based detection was introduced, which is immune to unknown offset and/or gain mismatch, but the drawback is that it is pretty sensitive to errors caused by the noise.

This thesis investigates possible coding techniques to improve decoders' performance in noisy channel conditions while maintaining the resistance against the gain and/or offset mismatch. The results discussed in the thesis are divided into four parts, based on different assumptions on the gain and/or offset mismatch. We describe each of the parts in further detail below.

We start with a fundamental model, where the offset mismatch is constant within a codeword length. A method called maximum likelihood (ML) decoding outputs the codeword that has the highest likelihood to the received word. Firstly, an ML decoding criterion is derived when assuming bounded distributions for both noise and offset mismatch. Most importantly, we investigate, for the proposed decoder, under which conditions zero word error rate performance can be achieved. Moreover, assuming Gaussian distributions for both noise and offset mismatch, we show that an ML decoding criterion is, in fact, to minimize a weighted average of Euclidean distance and Pearson distance. Based on this, we propose a concatenated scheme and its corresponding decoding algorithm. The concatenation is between a Reed-Solomon (RS) code and a certain coset of a block code (Coset). To maintain the immunity to offset mismatch, decoding on the inner Coset code is based on the Pearson distance while decoding the outer RS code adapts an efficient two-stage decoding algorithm. The proposed scheme achieves significant coding gain while it is simultaneously immune to offset mismatch.

Next, we look at a different model of the offset mismatch, where the offset mismatch is a *signal dependent* parameter. The signal dependency of the offset signifies that it may differ for distinct signal levels. We investigate an ML decoding criterion, assuming uniform distributions for both the noise and the signal dependent offset. In particular, for the proposed ML decoder, specific constraints on the standard deviations of the noise and the offset can lead to a zero error performance. Later, we derive an ML decoding criterion for signals suffering from Gaussian noise and signal dependent offset. Besides the ML criterion itself, an option to reduce the complexity is also considered. A brief performance analysis demonstrates the superiority of the newly developed ML decoder over classical decoders based on the Euclidean or Pearson distances.

We then direct our attention to a channel model in which the retrieved data is corrupted by Gaussian noise, *gain*, and offset mismatch. The intervals from which the gain and offset values are taken are known, but no further assumptions on their distributions are made. We derive a general framework of ML decoding criteria for such channels based on finding a codeword with the closest Euclidean distance to a specified set defined by the received vector and the gain and offset parameters. In addition, we give a geometric interpretation of the proposed ML criteria. It is shown that certain known criteria, including the gain-only case and the offset-only case, appear as special cases.

Lastly, we consider an even more complicated model, where data is transmitted over noisy channels with unknown gain and *varying* offset mismatch. We present a scheme for channels with unknown gain and varying offset, where Pearson distance based detection is used in conjunction with a difference operator. Pair-constrained codes are proposed for unambiguous decoding, where, in each codeword, certain adjacent symbol pairs must appear at least once. These codes significantly reduce redundancy compared to previously proposed mass-centered codes, making the new scheme an attractive alternative for practical applications. We also propose a systematic encoding algorithm of pair-constrained codes, and its redundancy is analyzed for memoryless uniform sources.

Samenvatting

Gegevensoverdracht is alomtegenwoordig, variërend van basistoestellen voor thuis en op kantoor, zoals compact disc-spelers en harde schijven, tot communicatie in de ruimte. De communicatie- en opslagkanalen hebben vaak last van ruis en de gegevens kunnen tijdens de transmissie worden vervormd. Ruis is echter niet de enige verstoring tijdens de gegevensoverdracht, en de informatie kan soms ernstig worden vervormd door onbekende kanaalverschijnselen, zoals mismatch van de gain of de offset (drift). De conventionele detectie op basis van de minimale Euclidische afstand, waarbij de ontvanger een codewoord uit het codeboek kiest om de Euclidische afstand met het ontvangen woord zo klein mogelijk te houden, levert slechte prestaties bij mismatch in gain en/of offset. Onlangs is een Pearson afstandsgebaseerde detectie geïntroduceerd, die immuun is voor onbekende offset en/of gain, maar het nadeel is dat deze erg gevoelig is voor fouten veroorzaakt door ruis.

Deze dissertatie onderzoekt mogelijke coderingstechnieken om de prestaties van decoders in ruisige kanaalomstandigheden te verbeteren, terwijl de weerstand tegen de mismatch van de gain en/of de offset behouden blijft. De resultaten die in het proefschrift worden besproken zijn verdeeld in vier delen, gebaseerd op verschillende aannames over de gain en/of offset mismatch. We beschrijven elk van de delen hieronder in meer detail.

We beginnen met een fundamenteel model, waarbij de offset mismatch constant is binnen een codeword lengte. Een methode genaamd maximum likelihood (ML) decoding geeft het codewoord dat de hoogste waarschijnlijkheid heeft voor het ontvangen woord. Eerst wordt een ML-decodeercriterium afgeleid wanneer wordt uitgegaan van begrensde verdelingen voor zowel ruis als offset mismatch. Het belangrijkste is dat we onderzoeken, voor de voorgestelde decoder, onder welke voorwaarden een woordfoutenkans van nul kan worden bereikt. Bovendien tonen we aan, uitgaande van Gaussische verdelingen voor zowel ruis als offset mismatch, dat een ML-decoderingscriterium in feite het minimaliseren is van een gewogen gemiddelde van Euclidische afstand en Pearson-afstand. Op basis hiervan stellen wij een aaneengeschakeld schema en het bijbehorende decoderingsalgoritme voor. De aaneenschakeling is tussen een Reed-Solomon (RS) code en een bepaalde coset van een blokcode (Coset). Om de immuniteit tegen offset-mismatches te behouden, is de decodering van de binnenste Coset-code gebaseerd op de Pearson-afstand, terwijl voor de decodering van de buitenste RS-code een efficiënt tweefasig decoderingsalgoritme wordt gebruikt. Het voorgestelde schema bereikt een aanzienlijke coderingswinst terwijl het tegelijkertijd immuun is voor offset mismatch.

Vervolgens bekijken we een ander model van de offset-mismatch, waarbij de offset-mismatch een signaal afhankelijke parameter is. De signaalafhankelijkheid van de offset betekent dat deze kan verschillen voor verschillende signaalniveaus. Wij onderzoeken een ML-decodeercriterium, uitgaande van uniforme verdelingen voor zowel de ruis

als de signaal afhankelijke offset. In het bijzonder kunnen voor de voorgestelde ML-decoder specifieke beperkingen op de standaardafwijkingen van de ruis en de offset leiden tot een nul-fout prestatie. Later leiden we een ML-decodeercriterium af voor signalen met Gaussische ruis en signaal afhankelijke offset. Naast het ML-criterium zelf, wordt ook een optie overwogen om de complexiteit te verminderen. Een beknopte prestatie-analyse toont de superioriteit aan van de nieuw ontwikkelde ML-decoder ten opzichte van klassieke decoders gebaseerd op de Euclidische of Pearson-afstanden.

Vervolgens richten wij onze aandacht op een kanaalmodel waarin de opgehaalde gegevens worden gecorrumpeerd door Gaussische ruis, gain en offset mismatch. De intervallen waaruit de gain- en offset-waarden worden genomen zijn bekend, maar er worden geen verdere aannames gedaan over hun verdelingen. Wij leiden een algemeen kader af van ML-decodeercriteria voor dergelijke kanalen, gebaseerd op het vinden van een codewoord met de Euclidische afstand die het dichtst ligt bij een gespecificeerde reeks, gedefinieerd door de ontvangen vector en de gains- en offset-parameters. Bovendien geven wij een geometrische interpretatie van de voorgestelde ML-criteria. Er wordt aangetoond dat bepaalde bekende criteria, met inbegrip van het geval van alleen de gain en alleen de offset, als speciale gevallen voorkomen.

Tenslotte beschouwen wij een nog gecompliceerder model, waarbij gegevens worden verzonden over ruiskanalen met onbekende gain en een variërende offset-mismatch. Wij presenteren een schema voor kanalen met onbekende gain en variërende offset, waarbij detectie op basis van Pearson-afstand wordt gebruikt in combinatie met een verschil-operator. Voor ondubbelzinnige decodering worden paarsgewijze codes voorgesteld, waarbij in elk codewoord bepaalde aangrenzende symboolparen ten minste eenmaal moeten voorkomen. Deze codes verminderen de redundantie aanzienlijk in vergelijking met eerder voorgestelde massa-gecentreerde codes, waardoor het nieuwe schema een aantrekkelijk alternatief wordt voor praktische toepassingen. We stellen ook een systematisch coderingsalgoritme voor paarsgewijze codes voor, en de redundantie wordt geanalyseerd voor geheugenloze uniforme bronnen.

1

INTRODUCTION

This chapter provides an introduction to this thesis. We first introduce the background and the basics of error-correcting codes. Then we talk about the research status of coding techniques over channels with noise, gain and/or offset mismatch, and present the motivation of this work. Afterward, we shortly describe the contribution of each chapter.

1.1. BACKGROUND

With the rapid development of the Internet and personal consumer electronics, the amount of data created and replicated experienced unusually high growth in 2020 due to the dramatic increase in the number of people working, learning, and entertaining themselves from home. According to the International Data Corporation [1], the 'Global Datasphere' in 2020 reached 64.2 zettabytes, defying the systemic downward pressure asserted by the COVID-19 pandemic on many industries. The explosive growth of data caused a great deal of pressure. The rapid growth in the amount of data has an increasingly high demand for storage capacity. The increasing value of the data itself also requires fewer risks of error when transmitting it. As a result, the ability to transmit and store large-scale data reliably is of great importance.

It is usually found that noise is an important issue during data transmission. However, other physical factors may also hamper the reliability of the stored data. Tackling the problem of data distortions such as noise, intersymbol interference, gain and/or offset mismatch, fading, clock jitter, etc., is a fundamental and challenging topic in the theory of channel coding. We are interested in data transmission and storage over channels with noise and gain and/or offset mismatch. Perhaps the best-known example of these channels is Flash memory, which we will discuss in detail in the following section.

1.1.1. FLASH MEMORY AND ITS ISSUE OF CHARGE LEAKAGE

Storage media can be divided into volatile memory and non-volatile memory according to whether they can keep the information even after power is removed. The most typical volatile memory includes dynamic random access memory and static random access memory. Although volatile memory can only save data temporarily, its high-speed access feature makes it widely used in processing units, which undertakes the vital task of caching data and instructions in both big data systems and embedded systems. Besides, the non-volatile memory is usually for external and long-term uses because it can stably keep the data for a long time.

As a non-volatile memory with high reliability, high capacity, and low cost, Flash memories [2, 3] are widely used for storage and data transfer in consumer electronics, enterprise systems, and industrial applications. Flash memory programs several bytes of data on a page and rewrites data at byte level. Several pages form a block, and the erasing process is based on blocks. In addition to the information, each page has a relatively small redundant area for storing the file system and error correction codes. Structure of the storage array inside MT29F64G08CBAB[A/B] Flash memory chip of the Micron company [4] is shown in Figure 1.1 as an example. The storage capacity of 64 Gb is divided into two sides, each of which has 2048 blocks. Each block contains 256 pages, and the data area and redundancy area of each page are 8192 bytes and 744 bytes, respectively. Double-sided design is a common way to improve the read-write efficiency in advanced Flash memory. When one side is in the read or write state, the other side can execute new read-write instructions.

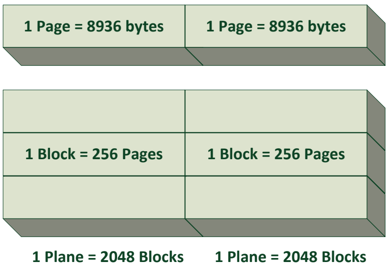


Figure 1.1: Structure of storage array inside Flash.

The most direct way to increase the storage capacity of Flash memories is to introduce multi-level cell technology. According to the number of bits stored in each memory cell, Flash memories can be divided into single-level cell (SLC) and multi-level cell (MLC). Flash memories are built from programmable and erasable floating-gate transistors. The information is represented by the amount of charge on a gate [5], and this charge within a specific voltage range gives a possible value of a cell. Figure 1.2 describes the voltage distributions for an SLC, an MLC, and a triple-level cell (TLC). The x-axis and y-axis are voltages and the distribution of voltages that correspond to different charge states, respectively. Charge voltage in an SLC is only divided by high and low levels, storing only one bit per memory cell, while an MLC is generally divided into four intervals from the highest level to the lowest level. Each storage cell can store two bits of information [6, 7]. There is a TLC with even more voltage intervals that can store three-dimensional information to pursue a large capacity further. For instance, each value level $(0, 1, \ldots, 7)$ in a TLC represents a 3-bits information that is stored in the Flash cell, e.g. 111, 110, 100, 000, 010, 011, 001, or 101. It is clear that, as the number of levels increases, MLC and TLC significantly enhance the storage capacity of Flash memories compared with SLC. However, the ranges of voltage that each value may occupy decrease. Smaller intervals in the TLC, for instance, have a higher overlap, resulting in a higher error possibility when the voltage may deviate from the nominal values due to the physical effect. Thus, multi-level cell technology also reduces Flash memories reliability and introduces higher error risk.

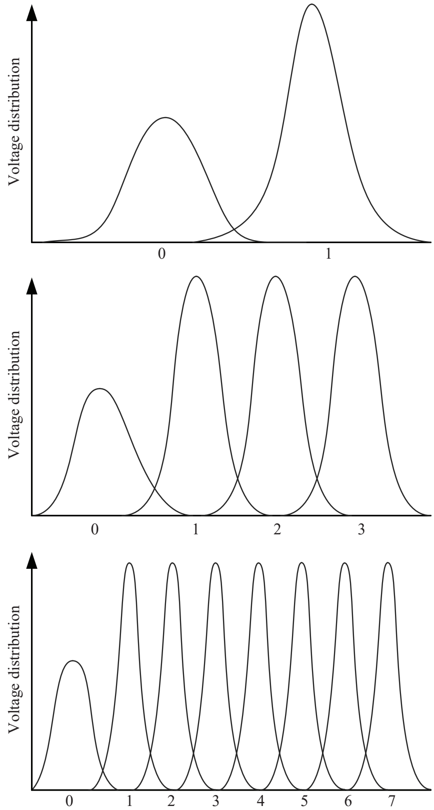


Figure 1.2: The distributions of the voltage in a cell for various values are shown for the SLC (one bit per cell) case on top, the MLC (two bits per cell) case in the middle, and the TLC (three bits per cell) case on the bottom. Note that the distribution in the TLC case must be much thinner to fit the same voltage interval and, even so, have a much larger potential for error.

The reliability of multi-level cell memories experiences a diverse set of short-term and long-term deviations. Short-term variations exacerbate unpredictable stochastic errors. For example, random errors occur under a program/erase (P/E) cycling process, where the data was read out right after programming.

As for the long term, *charge leakage* may be one of critical issues of multi-level cell memories. As documented in [8–13], voltage of a cell decreases and some cells even become defective over time. The amount of charge leakage, which can be modeled as gain and/or offset mismatch, depends on various physical parameters, such as the device temperature, the magnitude of the charge, and the time elapsed between writing and reading data [11]. Importantly, the charge leakage leads to a severe shift in the voltage distribution over time. Techspot reports an unfortunate degradation of the read performance of Samsung's solid-state drive (SSD) 840 [14], which happens in older data blocks. Reading old files is always slower than normal (about 30MB/s) due to its inherent voltage drift, while newly written files (such as those used in benchmarks) are as fast as new (about 500MB/s for the well-regarded SSD 840 EVO). Figure 1.3 shows the threshold voltage distribution of Flash memory at different retention ages for 8k P/E cycles [12]. If a predetermined fixed threshold detection is used, these offsets will increase the potential for errors.

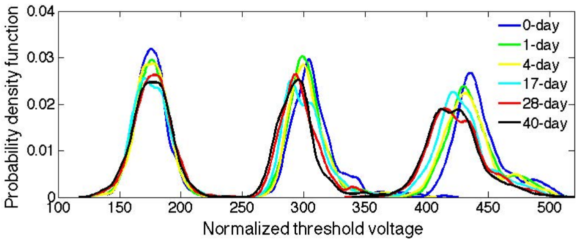


Figure 1.3: Threshold voltage distribution of MLC NAND Flash memory versus retention age [12].

Flash consumption on disk drives or SD cards has exploded over the last ten years, and new products, such as SSDs, are now making a significant introduction into personal electronics, mobile computing, intelligent vehicles, enterprise storage, data warehouse, and data-intensive computing systems. It is a crucial task to improve and expand nonvolatile Flash technologies. However, Flash also faces significant challenges, which can be overcome to a certain extent through innovative coding and signal processing technologies. We will give a brief introduction to coding theory in the following subsection. The results in this thesis are not limited to only Flash memory but also support other data transmission processes facing the same problem.

1.1.2. ERROR-CORRECTING CODES

Any communication *channel* refers to the transmission of information, either in a spatial domain from one place to another or a time domain from one time to another. In these two types of communication channels, information may not be transmitted correctly. Various factors may cause these distortions. If information is transmitted in the spatial domain, then any natural source, such as weather conditions, radiation, thermal effects, etc., will cause errors. In the time domain, information is stored on a memory device, and any physical defect or reduction in memory reliability will damage the stored data. For example, a scratch on a CD will damage the stored information. In both channels, the reliability issue of information transmission needs to be solved with proper signal processing and coding techniques.

Over half a century ago, the concept of *coding theory* was proposed by Claude E. Shannon in a seminal paper entitled "A Mathematical Theory of Communication". In his paper, the coding theory is subdivided into source coding theory and channel coding theory. Source coding theory reduces the number of bits needed to describe the data before transmitting them, which provides an efficient representations of data. On the other hand, by proper channel coding of the information, we can reduce the errors caused by the channel to any desired level without sacrificing the information transformation rate, as long as the information rate is less than the channel capacity. In this research, we focus on improving the error-correction ability of data transmission by channel coding.

Error-correcting codes provide a way of protecting the information from corruption in the channel. A simple block diagram of the data transmission in which error-correcting coding is applied is shown in Figure 1.4. The idea is to add some redundancy into a message $\mathbf{u}$ such that the original message can be obtained from received signals even if it is corrupted. We call this encoding, and an encoder gives the encoded sequence $\mathbf{x}$, which is called the codeword. The set of codewords forms a code. We transmit the codeword over the channel. The process is then to decode the received sequence $\mathbf{r}$ to a message $\mathbf{u}_o$ by a decoder. Successful decoding is not always guaranteed because of channel distortions.

Figure 1.4: Coding block diagram.

A simple method to add redundancy to protect our message is to transmit multiple copies of the message. Instead of transmitting only the message itself, at least three copies should be transmitted. A decoder decodes the received word to a message based on a simple decoding rule – the most frequently occurred message is the transmitted one. Then the decoder can output the correct message if only one copy is wrong. This rule is not guaranteed to catch all errors. However, the probability of error is reduced. We call this a repetition code. The Hamming code is another error-correcting code, which entails redundant parity check bits in the original message. Each check bit calculates the parity for some of the bits in the codeword. If the number of 1's among these bits are even, then the parity is 0; if the number of 1's among these bits are odd, then the parity is 1. On the receiver, calculating the parity of the received word can reveal whether there are any errors and where the errors are located. We will introduce some well-known error-correcting codes in detail in the next chapter.

1.2. RELATED RESEARCH

As we discussed before, there are various distortions, and we are interested in dealing with noise and gain and/or offset mismatch. One would like to understand how these channels with gain and/or offset mismatch differ from the classical noisy ones. There are conceptual connections, as well as significant differences between the noise distortion and the offset mismatch. Both of them are considered to have negative effects on the transmitted or stored signals. However, noise is a symbol-wise distortion, which is usually independently distributed for each symbol, and thus its value changes symbol by symbol. On the one hand, considering continuous inputs to the channel, the noise is usually modeled as an additive Gaussian variable. Decoders based on Euclidean distance are widely used as they have been shown to be optimal for additive white Gaussian noise [15]. On the other hand, we take a look at a binary discrete memoryless channel. When we wish to transmit a single bit (0 or 1), the bit may be flipped to the incorrect value ($0 \to 1$ or $1 \to 0$). A binary symmetric channel, where $0 \to 1$ and $1 \to 0$ errors occur with equal probability, is the simplest model for information transmission via a discrete channel, where the only distortion is additive white Gaussian noise. Most classes of codes are designed for adopting in symmetric channels, such as BCH codes [16], Reed-Solomon codes [17], LDPC codes [18], trellis coded modulation [19], and so on.

Gain and/or offset mismatch, on the contrary, is a type of block-wise distortion. It remains constant within one codeword length, then may change to another value and remains constant for another codeword length, and so on. As a result, while Euclidean distance-based decoding is known to be optimal if the transmitted or stored signals are only disturbed by Gaussian noise, it may perform poorly if there is a channel mismatch as well. We now turn to information transmission via a discrete channel. The offset mismatch has a high possibility to cause asymmetric or unidirectional errors since it is constant within a codeword length. A binary word is said to suffer from unidirectional errors if all errors are of the same type when sending a certain codeword [20], even though both $1 \to 0$ and $0 \to 1$ errors are possible. If we have *a priori* knowledge of what the types will be, they are called asymmetric errors [21]. In many practical applications, we can observe such asymmetric or unidirectional errors, for example, static random access memory cell [22], on-chip buses [23], and phase change memory [24, 25].

Clearly, to ensure reliable transmission over channels with noise and gain and/or offset mismatch, the use of some error detecting or/and correcting techniques has become mandatory. However, handling the problem of the unknown channel mismatch is a great challenge since coding techniques designed for noise may not be suitable for dealing with the channel mismatch due to the differences between them. Much of the research pays particular attention to gain and/or offset mismatch issues. These studies can be classified into the following three broad categories: (i) reference/pilot based methods, (ii) code constructions, and (iii) efficient detecting schemes.

(i) A few proposals overcome the negative effect of the gain and/or offset mismatch by using previous sequences or reference symbols.

A straightforward method applied in many practical transmission systems is an automatic gain (AGC) and offset control [26]. AGC modifies values of received signals depending on the weighted average offset of previously received sequences. Nevertheless, if the offset changes very rapidly, an AGC may be sub-optimal. A similar approach is accomplished by using training sequences or reference memory cells to estimate the unknown channel offset [27] and then adjusting the detector settings to match the actual values. Estimated values may be inaccurate because they lag behind actual values. Inserting reference symbols more frequently can improve the estimation; however, this comes at the cost of higher redundancy. A method is proposed in [13] for threshold calibration that is based on a small amount of pilot data per Flash page. Depending on the observed number of errors in the pilot data, corrections of voltage shifts are taken from a simple look-up table. However, the look-up tables are calculated using measurements of only a small number of scenarios, which can not cover all possible life-cycle states.

Recently, machine learning has developed rapidly, and it has shown superior performance in many aspects of communication systems. Employing a machine learning based technique in the improvement of the estimation of channel parameters is investigated in [28–31]. With these machine learning based frameworks, all the unknown offset or unpredictable variations in channels can be learned from the training data, thus avoiding the difficult task of modeling the practical channels. However, a large amount of training data are required to establish an unknown input-output relationship of the estimation model. The estimation may not be accurate without adequate training data.

All of the studies reviewed here show a strong resistance for not only the mismatch issue but also other unpredictable variations. This benefit comes at the cost of complexity as they need plenty of training data or reference symbols. Accuracy of estimated channel parameters might be another issue, especially for high-speed applications.

(ii) Up to now, various coding techniques have been applied to alleviate the detection in case of channel mismatch, such as, rank modulation [32], balanced codes [33–37], and composition check codes [38].

In rank modulation [32], data is carried by the relative charge levels of many cells and not by the charge level in a single cell. Assume a sequence of the charge levels in 5 cells is $(6,1,3,2,10)$. A codeword in this scheme is a permutation of cells induced by the charge levels of the cells, that is, $(5,1,3,4,2)$. This coding by ranking of charge levels eliminates the problem of charge leakage in aging devices. Research on the rank modulation scheme since its introduction more than ten years ago is developed extensively [39–43].

The notion of dynamic thresholds based on balanced codes is introduced in [33] for the reading of binary sequences. It is further shown to be highly effective against errors caused by voltage drift in Flash memories [34–36]. A balanced code consists of the sequences where the number of ones equals the number of zeros. With the dynamic thresholds scheme, there are no thresholds for comparison when reading signals if we use the balanced code since it has a fixed distribution of levels. However, this method adds complexity in encoding and decoding balanced codes, and the size of the codebook is relatively small. The generating function offers a tool for enumerating the balanced codes [44, 45]. Encoding/decoding of balanced codes has attracted a considerable amount of research and engineering attention [46, 47].

A balanced code is a constant composition code. The error performance of optimal detection of codewords that are drawn from a single constant composition code is immune to offset mismatch as showed by Slepian [37]. To further increase the size of the codebook, a composition check code is proposed [38]. Composition check codes have the virtues of Slepian's optimal detection method. A label is added that informs the receiver regarding the constant composition code to which the sent main data word belongs such that the encoding of the main data into a constant composition code is avoided.

The research reviewed here suggests a pertinent role in using proper coding schemes to solve the offset mismatch issue. However, the disadvantages of these methods, which have limited applicability, are the high redundancy and complexity. For example, the redundancy of a full set of balanced codewords is $O(\log m)$, where m is the number of user bits [48].

(iii) A promising decoding technique with asymptotic zero redundancy as the codeword length increases is proposed in [49], where it is shown that decoders using the Pearson distance have immunity to offset and/or gain mismatch. A study [50] shows that a digital modulation transceiver based on Pearson distance detection provides excellent error performance for noisy channels with Rayleigh fading. The use of the Pearson distance requires that the set of codewords satisfies several specific properties. Such sets are called Pearson codes, which have attracted a lot of interest [51–55]. In [51], optimal Pearson codes are presented, in the sense of having the largest number of codewords and thus minimum redundancy among all q-ary Pearson codes of fixed length n. Properties of binary Pearson codes are discussed in [52, 53], where the Pearson noise distance is compared to the well-known Hamming distance. A simple systematic Pearson coding scheme, that maps sequences of information symbols generated by a q-ary source to q-ary code sequences, is proposed in [54]. Construction of a particular kind of Pearson codes, i.e., T-constrained codes [49], using a finite state machine, is introduced in [55].

Furthermore, a considerable amount of literature has grown around the theme of Pearson distance that tackles the offset mismatch issues. In [56], a decoder is proposed based on minimizing a weighted sum of Euclidean and Pearson distances. A dynamic threshold detection scheme is proposed in [57], where the gain and offset are first estimated based on Pearson distance detection. The estimates of the gain and offset are used to re-scale the received signal within its normal range. Then, the re-scaled signal, brought into its standard range, can be forwarded to the final detection/decoding system, where the distance properties of the code can be optimally utilized by applying, for example, the Chase algorithm [58]. A detection scheme for channels with gain and such varying offset is investigated in [59, 60], where, for the binary case, minimum Pearson distance based detection is used in conjunction with mass-centered codewords.

1.3. RESEARCH QUESTIONS

The above decoding methods have improved the resilience to gain and/or offset mismatch, or even established immunity to it. However, the price paid for this benefit is a higher noise sensitivity. There is a natural decoding technique, known as maximum likelihood (ML) decoding, where the decoder outputs the codeword with the highest likelihood of being the one that was actually transmitted.

Minimum Euclidean distance detection is an ML criterion for an additive white Gaussian noise channel[15], but it may perform poorly against gain and/or the offset mismatch. Minimum Pearson distance detection is an ML criterion for the gain and/or offset mismatch channel without noise [49]. It is crucial and challenging to study the ML decoding solutions considering both noise and offset issues. Blackburn [61] investigates an ML criterion for channels with Gaussian noise and unknown gain and offset mismatch. In a subsequent study, ML decoding criteria are derived for Gaussian noise channels when assuming various distributions for the offset in the absence of gain mismatch [62]. This research aims to investigate possible coding techniques for noisy channels with gain and/or offset mismatch.

Main Research Question: what are possible ways to improve decoders' error correction performances with noisy channel conditions while maintaining the resistance against gain and offset mismatch?

Investigating this research question, we consider four models of offset mismatch from simple to complex. The offset mismatch, which is constant within a codeword length, serves as the most fundamental model. The next part concerns the dependency between offset mismatch and signal levels. Later we discuss channels with gain and offset mismatch. A gain and slowly varying offset model is our last consideration. The various forms of the channel mismatch provide different ways to externalize physical distortions in practical systems, but they are not limited only by those shown in this work.

- *How can an ML decoding criterion be established for noisy channels with offset mismatch?*

  In storage and communication systems, noise is not the only disturbance during data transmission. Sometimes the error performance can also be seriously degraded by other physical factors, such as offset mismatch. We focus on noisy channels with unknown offset mismatch, where an offset is constant within a codeword length and may vary word by word. An ML decoding, which outputs that the codeword has the highest likelihood to the received word, is carried out for such channels.

- *How to establish an ML decoding criterion for noisy channels with signal dependent offset mismatch?*

  With the changes in the environment, different signal levels may suffer from distinct offset mismatch values. The signal dependency of the offset signifies that it may differ for different signal levels. This thesis explores an ML decoding criterion for noisy channels with signal dependent offset mismatch and discusses two situations – uniform and Gaussian distributions for noise and offset mismatch.

- *What are an ML decoding criterion for noisy channels with gain and offset mismatch?*

  An ML decoding is considered for channels with gain and offset mismatch. We discuss an ML decoding for situations with different gain and/or offset and provide geometric interpretations of gain and offset.

- *What are possible detecting techniques for noisy channels with gain and slowly varying offset mismatch?*

  The basic premises considered here are that a codeword is received with an unknown gain, is offset by an unknown varying offset, and corrupted by additive Gaussian noise. We study minimum Pearson distance based detection in conjunction with a difference operator. It is independent of unknown channel gain and varying offsets. For such a detection scheme, constrained codes need to develop further.

1.4. THESIS OVERVIEW

The first two chapters introduce the research background and basic mathematics definitions used in this thesis. Related prior works are presented as well.

Chapter 3 mainly discusses the coding techniques for a simple and basic model of noisy channels with offset mismatch, where an unknown offset is assumed to be in the all-one direction. An ML decoding criterion is first derived for such channels assuming bounded distributions for both noise and offset mismatch. We also give the conditions under which the proposed decoder achieves a zero error performance. Further, an ML decoding criterion is presented for such channels with Gaussian noise and offset. A concatenated coding scheme is proposed in the case of Gaussian noise and offset mismatch. A decoding algorithm for concatenated codes is also proposed. Simulation results verify the algorithm's effectiveness in combatting against offset mismatch.

Chapter 4 takes a different look, and models the offset mismatch as a signal dependent parameter. We investigate an ML decoding criterion for the situation that the noise and the offset are uniformly distributed. A zero error performance is achieved when the standard deviations of the noise and the offset are small enough. Later, an ML decoding criterion is derived for such channels to improve and strengthen the resilience to Gaussian noise and signal dependent offset. For codebooks consisting of the union of constant weight sets, it shows that significant complexity reductions can be obtained.

The attention of Chapter 5 is put on a channel model in which the retrieved data is corrupted by Gaussian noise, gain, and offset mismatch. A general framework of maximum likelihood decoding criteria for such channels is summarized. We give geometric interpretations of gain and offset and show that certain known criteria appear as special cases of our general setting.

Chapter 6 considers an even more complicated channel model, where data is transmitted over noisy channels with unknown gain and varying offset mismatch. A minimum Pearson distance detection is used in cooperation with a difference operator, which offers immunity to such mismatch. Novel pair-constrained codes are proposed for such a decoding scheme, where cardinality and redundancy of these codes are derived. A simple systematic encoding algorithm of pair-constrained codes is presented, and its redundancy is analyzed for memoryless uniform sources. Simulation results are given for the proposed scheme when the channel is corrupted with additive white Gaussian noise.

Finally, we summarize this thesis and give our recommendations on future research in Chapter 7.

2

Basic Concepts and Methods

In Galois Fields, full of flowers
primitive elements dance for hours
climbing sequentially through the trees
and shouting occasional parities.

S.B. Weinsteine

This chapter aims to review the basic definitions relating to error-correcting codes and standardize some notation. We then give a brief description of the channel model that will be dealt with and used in this work. Finally, we discuss the fundamental decoding techniques and the main results established in the prior art, explaining how the various techniques perform.

2.1. Basic Concepts

2.1.1. Basic Definitions for Codes

In order to avoid introducing too much formalism and notation this early on, we only discuss the most fundamental definitions. We will defer a formal treatment of further definitions until they are needed.

Block Codes

For an integer $q \geq 2$, let $[q] = \{0, 1, \ldots, q-1\}$.

- A *block code* S is a subset of $[q]^n$, which is the n-dimensional vector space over the *alphabet* $[q]$.

- The integer q is referred to as the *alphabet size* of the code, or alternatively we say that S is the q-ary code. We say S is a *binary* code when $q = 2$; if $q = 3$, then it is called *ternary* block code.

- The elements of S are called the *codewords* in S. The length of every codeword, n, is called *codeword length*.

The number of codewords, also called the *cardinality* or *size* of the code, is denoted by $|S|$. The *code rate* of a q-ary code S of size $|S|$, denoted $r_S(n)$, is defined to be the normalized quantity

$$r_S(n) = \frac{\log_q |S|}{n}. \tag{2.1}$$

The code rate determines the amount of redundancy. Since the cardinality of a q-ary code S is at most q^n, the code rate is a real number between 0 and 1. The closer it is to 1, the less redundantly the data is encoded by the block code.

It is convenient to view a code $S \subseteq [q]^n$ of size $|S|$ as a function where messages $\mathbf{u}$ generated by the source are one-to-one mapped to codewords $\mathbf{x} \in S$. Often we will take $|S|$ to be a power of q, that is $|S| = q^k$, where k is the *dimension* of the code. Note that the dimension k is at most n. In this way, an error-correcting code offers a method to add some redundancy to a q-ary message vector $\mathbf{u}$ of length k and encode it into a longer vector, or called a codeword, $\mathbf{x}$ of length n.

HAMMING DISTANCE OF CODE

Hamming distance is a notion in a formal mathematical sense that captures how close-by two vectors $\mathbf{u}$ and $\mathbf{v}$ are. Given any two vectors $\mathbf{u},\mathbf{v} \in [q]^n$ of the same length, where $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, the *Hamming distance* is defined as the number of coordinates where these two vectors differ, that is,

$$d_H(\mathbf{u},\mathbf{v}) = |\{i : u_i \neq v_i\}|. \tag{2.2}$$

The *weight*, denoted as $w(\mathbf{u})$, of a vector $\mathbf{u}$ is the Hamming distance between this vector and the all-zero vector $\mathbf{0}$.

Now, we introduce more operations related to the Hamming distance in the binary case. A binary block code is a subset of $[2]^n$, where $[2]$ is the field containing only two elements 0 and 1. The following modulo-2 addition and multiplication on this field are defined:

$\oplus$	0	1
0	0	1
1	1	0

$\odot$	0	1
0	0	0
1	0	1

The number of coordinaties that equal 1 in a binary vector $\mathbf{u}$ is the weight $w(\mathbf{u})$ of $\mathbf{u}$:

$$w(\mathbf{u}) = |i\,|u_i = 1\,|.$$

The Hamming distance between two binary vectors $\mathbf{u}$ and $\mathbf{v}$ is equal to the weight of modulo-2 addition of $\mathbf{u}$ and $\mathbf{v}$, that is, $d_H(\mathbf{u},\mathbf{v}) = w(\mathbf{u} \oplus \mathbf{v})$. It is also equal to

$$d_H(\mathbf{u},\mathbf{v}) = \sum_{i=1}^{n} |u_i - v_i|. \tag{2.3}$$

The *minimum Hamming distance* (or simply *distance*) of a code S, denoted $d(S)$, is the minimum Hamming distance between any two distinct codewords of S. Formally,

$$d(S) = \min\{d_H(\mathbf{u},\mathbf{v}) : \mathbf{u},\mathbf{v} \in S, \mathbf{u} \neq \mathbf{v}\}. \tag{2.4}$$

We represent a q-ary block code of length n, dimension k, and minimum Hamming distance d, as an $(n,k,d)_q$ code. The distance parameter can be omitted, and an $(n,k)_q$ code refers to a q-ary code of length n and dimension k. We will often omit the alphabet size subscript if it is clear from the context.

LINEAR CODES

Let q be a prime power. We denote a finite field with q elements by $\mathbb{F}(q)$ or $GF(q)$ interchangeably. We assume, when necessary, that the field $GF(q)$ can be identified with $[q]$ in some canonical way.

If a block code S of length n is a linear *subspace* of $(GF(q))^n$, then S is called a *linear* (n,k) block code. In other words, an (n,k) block code is linear if

– *component-wise addition of two codewords is another codeword, i.e., if* $\mathbf{x}_1, \mathbf{x}_2 \in S$, *then* $a_1\mathbf{x}_1 + a_2\mathbf{x}_2 \in S$ *for any* a_1 *and* a_2 *from* $GF(q)$.

Clearly, a linear code over $GF(q)$ has q^k elements, where k is the dimension of the code, and so the code rate is $r_S(n) = k/n$. For linear codes, the all-zero vector is always a codeword. Thus the minimum Hamming distance of a linear code equals the minimum weight of all codewords except the all-zero codeword.

Any (n,k) linear code S satisfies the following properties. The full proof can be found in any standard linear algebra textbook.

1. There exists $\mathbf{v}_1,\ldots,\mathbf{v}_k \in S$ called basis elements, which need not be unique, such that every $\mathbf{x} \in S$ can be expressed as $\mathbf{x} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_k\mathbf{v}_k \in S$, where $a_i \in GF(q)$ for $1 \leq i \leq k$. In other words, there exists a full rank $k \times n$ matrix G, also known as a generator matrix, with entries from $GF(q)$ such that every $\mathbf{x} \in S$,

$$\mathbf{x} = (a_1, a_2, \ldots, a_k) \cdot G, \tag{2.5}$$

   where $G = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k)^T$.

2. There exists a full rank $(n-k) \times n$ matrix H (called a parity check matrix) such that $H\mathbf{x}^T = 0$ for every $\mathbf{x} \in S$.

3. G and H are orthogonal, that is, $G \cdot H^T = 0$.

The above properties give two alternate ways to specify an (n,k) linear code S. It can be generated by its $k \times n$ generator matrix G and also characterized by an $(n-k) \times n$ parity check matrix H. Encoding is a process by which a message $\mathbf{x} \in (GF(q))^k$ is encoded into its corresponding codeword by multiplying it with the generator matrix of the code.

COMMONLY USED NOTATION

Much of the notation we use is standard. Throughout the thesis, both $\log x$ and $\lg x$ will denote the logarithm of x to the base 2. We denote the natural logarithm of x by $\ln x$. For bases other than 2 and e, we explicitly include the base in the notation. For example, the logarithm of x to the base q will be denoted by $\log_q x$.

For a real number x, $\lfloor x \rfloor$ will denote the largest integer at most x, and $\lceil x \rceil$ will denote the smallest integer at least x. $\mathbb{R}$ denotes the set of real numbers.

2.1.2. Some Code Families

We describe some code families that will be studied in this work. Several of these will also be used as building blocks for the new code constructions that we present.

Single Parity Check Codes

Parity check is a simple way to add redundancy bits so that even if some of the information is lost or corrupted, it will still be possible to detect or even recover the message at the receiver. One example is a single parity check code [63]. Formally, an $(n, n-1)$ single parity check code, denoted by S_p, with $q = 2$, is defined as follows. Starting with a block of $n-1$ information bits $(x_1, x_2, \ldots, x_{n-1})$, we can combine them so that a single extra bit checks whether there is an error in the information bits. The n-th bit is chosen such that

$$\left(\sum_{i=1}^{n} x_i\right) \mod 2 = a, \tag{2.6}$$

where $a \in \{0,1\}$ is a pre-set integer. For instance, we can choose the n-th bit so that the parity of the entire block is 0, that is, the number of '1's in n bits is even. Then if an odd number of errors occurs during the transmission, the receiver will detect the error since the parity has changed. The code size equals $|S_p| = 2^{n-1}$ and the rate of the single parity check code is $1 - 1/n$. Single parity check code is a simple example of error-detecting codes.

Hamming Codes

The class of Hamming codes [64] is one of the oldest families of binary error-correcting codes. Hamming codes extend the idea of parity checks in two perspectives: (1) it allows for more than a single parity check bit, and (2) the parity checks depend on various subsets of the information bits. Formally, Hamming codes are most commonly defined from the parity check matrix.

The parity check matrix H has the property that for every codeword $\mathbf{x}$, $\mathbf{x}H^T = 0$. If the sum of any two columns is another column, then other than the all-zero codeword, the weight of $\mathbf{x}$ in order to satisfy this equation is at least 3. Hence, the Hamming distance of the code is at least equal to 3. Denote the number of the rows in H by $m \geq 2$. The maximum number of columns satisfying the requirements is $2^m - 1$. The H matrix with this maximum number of columns leads to an $(2^m - 1, 2^m - 1 - m, 3)$ code, called a Hamming code, denoted by $\mathscr{H}_m$. A generator matrix G can be easily derived from H. Thus the code length of Hamming code is $n = 2^m - 1$ and the dimension is $k = 2^m - 1 - m$. Hamming codes with the minimum distance of 3 are the simplest of a class of algebraic error-correcting codes that can correct one error or detect two errors in a block of bits.

To show an example Hamming code, we consider a binary code of block length 7. All operations will be done modulo 2. Arrange the set of all nonzero binary vectors of length 3 in columns to form a matrix:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{2.7}$$

From the properties of linear spaces, it also produces the 4-by-7 generator matrix G that corresponds to the parity-check matrix H.

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.8}$$

The set of codewords therefore forms a linear subspace of dimension 4 in the vector space of length 7. These 2^4 codewords are

0000000	0110100	1101000	1011100
1010001	1100101	0111001	0001101
1110010	1000110	0011010	0101110
0100011	0010111	1001011	1111111

Looking at the codewords, we notice that other than the all-zero codeword, the minimum number of '1's in any codeword is 3. This number is called the minimum weight of the code. The encoding is explicit so that the first $n-k$ bits in each codeword are parity check bits, and the last k bits represent the message. Such a code is called a *systematic code*. The above code is called a (7,4,3) Hamming code, denoted as $\mathscr{H}_3$, i.e., $n=7$, $k=4$, and $d=3$.

2.2. CHANNEL MODEL

We consider transmitting a codeword $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ from a code $S \subseteq [q]^n$. In this work, we focus on the situation that the received vector may not only be hampered by noise $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, but also by offset mismatch. Hence, the received symbols read

$$r_i = x_i + v_i + b, \tag{2.9}$$

for $i = 1, \ldots, n$, where b represents an unknown offset mismatch. We assume that the offset mismatch is constant within one codeword block length and may vary block by block, i.e.,

$$\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}, \tag{2.10}$$

where $\mathbf{1}$ is the all-one vector. Values of the offset mismatch are unknown to both the sender and the receiver.

This model (2.10) as our most primary and straightforward assumption of the channel mismatch is investigated in Chapter 3. However, our research is not limited to only one situation. Other models of the channel mismatch are included based on this blockwise model. The signal-dependent offsets, $b_{\mathbf{x}}$, are investigated in Chapter 4, and later Chapter 5 discusses the channels with gain and offset mismatch, where the gain is presented by a parameter a. We also consider the gain and slowly varying offset model in Chapter 6, where an unknown slope, c, of varying offset is introduced. The various forms of the channel mismatch provide different ways to externalize physical distortions in practical systems, but they are not limited only by those shown in this work.

The receiver would like to pick a codeword from the code that he believes is transmitted based on some criteria. Various traditional methods will be introduced in the following sections. Before that, we define a useful statistics function, *Q-function*, which will be used in the error performance analysis of detection schemes.

2

The *Q-function* is a convenient way to express right-tail probabilities for normal (Gaussian) random variables. For $x \in \mathbb{R}$, $Q(x)$ is defined as the probability that a standard normal random variable (zero mean, unit variance) will obtain a value larger than x, that is,

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt.$$

If we have a normally distributed variable $X \sim \mathcal{N}(\mu, \sigma^2)$, i.e., X has a probability density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

the probability that $X > x$ is

$$\mathbb{P}(X > x) = Q\left(\frac{x-\mu}{\sigma}\right). \tag{2.11}$$

CHANNEL RAW BIT ERROR RATE (BER)

Here we show the asymmetric or unidirectional errors caused by the offset mismatch by analyzing the channel raw bit error rate (BER). Consider transmitting a codeword $\mathbf{x}$, from a binary code $S \subseteq [2]^n$ through a channel as shown in (2.10). We derive the raw BER theoretically for such a channel. The symbols of the received vector, r_i, can be straightforwardly quantized to a hard decision bit, $x_i^d \in \{0,1\}$, with a conventional fixed threshold detection. The threshold function is denoted by $x_i^d = \vartheta(r_i)$, where a threshold V_m is used such that

$$\vartheta(r_i) = \begin{cases} 1 & \text{if } r_i \geq V_m, \\ 0 & \text{otherwise.} \end{cases}$$

In the following, we will show the influence of offset mismatch on the bit error performance of the channel hard detected bits x_i^d threshold detection. The BER P_b of x_i^d is computed as

$$\begin{aligned} P_b &= \Pr(x_i = 0)\Pr(x_i^d \neq x_i \,|\, x_i = 0) + \Pr(x_i = 1)\Pr(x_i^d \neq x_i \,|\, x_i = 1) \\ &= \Pr(x_i = 0)\Pr(r_i \geq V_m \,|\, x_i = 0) + \Pr(x_i = 1)\Pr(r_i < V_m \,|\, x_i = 1). \end{aligned}$$

Assuming independent and uniformly distributed channel inputs, we thus have

$$\begin{aligned} P_b &= \frac{1}{2}\Pr(r_i \geq V_m \,|\, x_i = 0) + \frac{1}{2}\Pr(r_i < V_m \,|\, x_i = 1) \\ &= \frac{1}{2}\int_{V_m}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r-0-b)^2}{2\sigma^2}} dr + \frac{1}{2}\int_{-\infty}^{V_m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r-1-b)^2}{2\sigma^2}} dr \\ &= \frac{1}{2}Q\left(\frac{V_m - b}{\sigma}\right) + \frac{1}{2}\left(1 - Q\left(\frac{V_m - 1 - b}{\sigma}\right)\right). \end{aligned} \tag{2.12}$$

In Figure 2.1, we have drawn the probability (2.12) assuming a threshold $V_m = 0.5$ as a function of signal to noise ratio (SNR). The SNR is defined by

$$SNR(dB) = -20\log_{10}\sigma. \tag{2.13}$$

The offset mismatch reduces system reliability, and the greater offset values cause the larger error probabilities P_b. Moreover, it has a higher chance of occurring $0 \to 1$ error than $1 \to 0$ error when the values of offset are positive, as we have shown in Figure 2.2; chance of occurring $1 \to 0$ error is higher than $0 \to 1$ error when the values of offset are negative. That is to say, offset mismatch causes asymmetric errors or unidirectional errors.

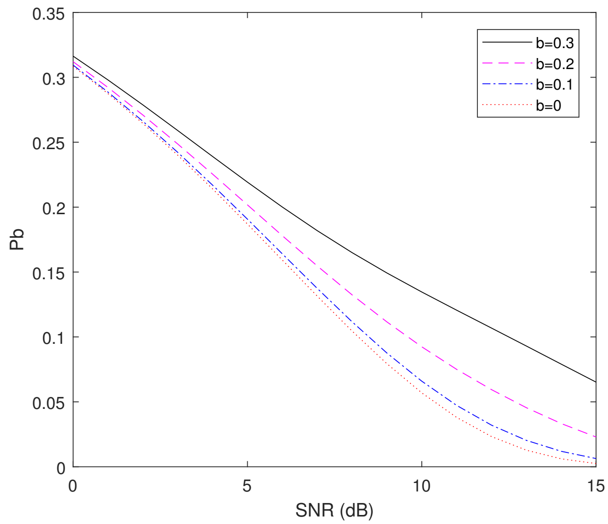


Figure 2.1: Channel raw BER P_b as a function of SNR assuming different values of the offset b.

In a channel with only Gaussian noise, the probabilities of occurring $0 \to 1$ error and $1 \to 0$ error are the same, called symmetric errors. Then the channel raw BER is

$$P_b^g = \frac{1}{2}\left[Q\left(\frac{V_m}{\sigma}\right) + Q\left(\frac{1-V_m}{\sigma}\right)\right],$$

which can be obtained by letting $b = 0$ in (2.12) and $Q(-x) = 1 - Q(x)$ for $x > 0$. Note that the raw BER of channels with offset mismatch is always larger than a channel with only Gaussian noise, i.e., $P_b \geq P_b^g$.

2.3. MINIMUM EUCLIDEAN DISTANCE BASED DETECTION (MED)

A well-known decoding criterion upon receipt of the vector $\mathbf{r}$ is to choose a codeword that minimizes the (squared) Euclidean distance between the received vector $\mathbf{r}$ and the candidate codeword $\hat{\mathbf{x}}$. Minimum Euclidean distance based detection (MED) is known

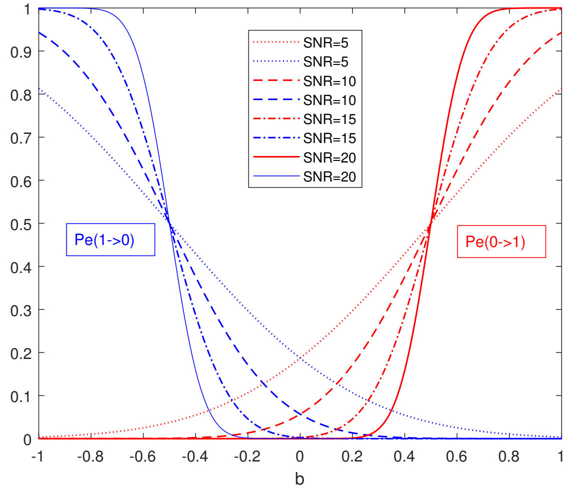


Figure 2.2: Probabilities of $0 \to 1$ and $1 \to 0$ errors as a function of SNR assuming different values of the offset b, where $Pe(0 \to 1)$ and $Pe(1 \to 0)$ correspond to $\Pr(r_i \geq V_m \,|x_i = 0)$ and $\Pr(r_i < V_m \,|x_i = 1)$, respectively.

to be optimal with regard to handling Gaussian noise without offset mismatch. That is, we assume that the channel (2.10) is ideally matched, i.e., $b = 0$, so the received vector after transmitting a codeword $\mathbf{x}$ is

$$\mathbf{r} = \mathbf{x} + \mathbf{v}, \tag{2.14}$$

where the noise vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is such that the v_i are identically independently distributed (i.i.d.) Gaussian random variables with zero mean and variance σ^2, i.e., $v_i \sim \mathcal{N}(0, \sigma^2)$. The probability density function of $\mathbf{v}$ is

$$\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-v_i^2/(2\sigma^2)}. \tag{2.15}$$

The squared Euclidean distance between $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^n$ is defined by

$$\delta_E(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} (u_i - v_i)^2. \tag{2.16}$$

Upon receipt of a vector $\mathbf{r}$, the receiver decides that the codeword $\mathbf{x}_o$ was sent if (2.16) attains its least value for $\hat{\mathbf{x}} = \mathbf{x}_o$, that is

$$\mathbf{x}_o = \arg\min_{\hat{\mathbf{x}} \in S} \delta_E(\mathbf{r}, \hat{\mathbf{x}}). \tag{2.17}$$

We consider four codewords $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, and $\mathbf{x}_4$ in Figure 2.3. Points in the Euclidean space are separated into four subsets by surfaces being perpendicular to the line segments formed by each pair of codewords. Each subset contains only one codeword, thus if a received vector belongs to a certain subset, MED will decode it as the codeword within this subset.

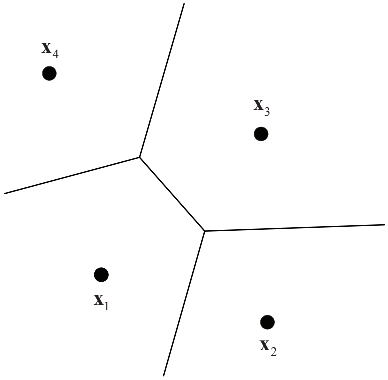


Figure 2.3: Graphic illustration of minimum Euclidean distance based detection of four codewords.

2.3.1. PERFORMANCE ANALYSIS

We now examine the theoretical word error rate upper bound of block codes in the minimum Euclidean distance based detection scheme. Word error rate (WER) is defined as the ratio of the number of incorrectly decoded words to the number of words transmitted. The receiver uses the Euclidean distance (2.17) for the evaluation of the received word, where we assume that $\mathbf{x} \in S$ is sent, and received as $\mathbf{r} = \mathbf{x} + \mathbf{v}$. The Euclidean detector errs if there is at least one codeword $\hat{\mathbf{x}} \neq \mathbf{x}$, $\hat{\mathbf{x}} \in S$, such that

$$\delta_E(\mathbf{r}, \hat{\mathbf{x}}) < \delta_E(\mathbf{r}, \mathbf{x})$$

or

$$\delta_E(\mathbf{r}, \hat{\mathbf{x}}) - \delta_E(\mathbf{r}, \mathbf{x}) < 0. \tag{2.18}$$

Using the definition of the squared Euclidean distance (2.16), the left side of (2.18) is

$$\begin{aligned}
\delta_E(\mathbf{r}, \hat{\mathbf{x}}) - \delta_E(\mathbf{r}, \mathbf{x}) &= \sum_{i=1}^{n} (r_i - \hat{x}_i)^2 - \sum_{i=1}^{n} (r_i - x_i)^2 \\
&= \sum_{i=1}^{n} (x_i + v_i - \hat{x}_i)^2 - \sum_{i=1}^{n} v_i{}^2 \\
&= \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 + 2\sum_{i=1}^{n} (x_i - \hat{x}_i) v_i,
\end{aligned}$$

where the second equality follows from (2.14). The left side of (2.18) is a stochastic variable with distribution $\mathcal{N}(\alpha_E, \beta_E \sigma^2)$, where $\alpha_E = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$ and $\beta_E = 4\sum_{i=1}^{n} (x_i - \hat{x}_i)^2$. Thus, the probability that $\delta_E(\mathbf{r}, \hat{\mathbf{x}}) < \delta_E(\mathbf{r}, \mathbf{x})$ equals

$$\mathbb{P}(\delta_E(\mathbf{r}, \hat{\mathbf{x}}) < \delta_E(\mathbf{r}, \mathbf{x})) = Q\left(\frac{\alpha_E}{\sqrt{\beta_E}\sigma}\right) = Q\left(\frac{\sqrt{\delta_E(\mathbf{x}, \hat{\mathbf{x}})}}{2\sigma}\right). \tag{2.19}$$

WER over all coded sequences can be upper bounded by using a union bound type of argument. If MED is used as the decoding criterion, then it is well known that

$$WER \leq \frac{1}{|S|} \sum_{\mathbf{x} \in S} \sum_{\hat{\mathbf{x}} \in S, \mathbf{x} \neq \hat{\mathbf{x}}} Q\left(\frac{\sqrt{\delta_E(\mathbf{x}, \hat{\mathbf{x}})}}{2\sigma}\right). \tag{2.20}$$

Define the squared of the minimum Euclidean distance between any possible pair of codewords in S by

$$\delta_{E,min} = \min_{\mathbf{x},\hat{\mathbf{x}}\in S,\mathbf{x}\neq\hat{\mathbf{x}}} \delta_E(\mathbf{x},\hat{\mathbf{x}}). \tag{2.21}$$

2

Then, for asymptotically small standard deviations of the noise, i.e., for $\sigma \ll 1$, the WER is upper bounded by

$$WER \leq N_e \times Q\left(\frac{\sqrt{\delta_{E,min}}}{2\sigma}\right), \tag{2.22}$$

where N_e is the average number of pairs of codewords at minimum Euclidean distance $\delta_{E,min}$.

2.4. MINIMUM PEARSON DISTANCE BASED DETECTION (MPD)

We start by introducing some notation. For any vector $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{R}^n$, let

$$\bar{\mathbf{u}} = \frac{1}{n}\sum_{i=1}^{n} u_i$$

denote the average symbol value, let

$$\sigma_{\mathbf{u}} = \sqrt{\sum_{i=1}^{n}(u_i - \bar{\mathbf{u}})^2}$$

denote the unnormalized symbol standard deviation, and let

$$\|\mathbf{u}\| = \sqrt{\sum_{i=1}^{n}|u_i|^2}$$

denote the (Euclidean) norm. We write $\langle\mathbf{u},\mathbf{v}\rangle$ for the standard inner product (the dot product) of two vectors $\mathbf{u}$ and $\mathbf{v}$, i.e.,

$$\langle\mathbf{u},\mathbf{v}\rangle = \sum_{i=1}^{n} u_i v_i = \|\mathbf{u}\|\,\|\mathbf{v}\|\cos\theta,$$

where θ is the angle between $\mathbf{u}$ and $\mathbf{v}$. Note that $\langle\mathbf{u},\mathbf{u}\rangle = \|\mathbf{u}\|^2$.

The error performance of the traditional MED scheme degrades in the case that there exists a channel mismatch. Luckily, Immink and Weber [49] showed that detectors that use the Pearson distance offer immunity to offset and/or gain mismatch. The Pearson distance between two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined as

$$\delta_P(\mathbf{u},\mathbf{v}) = 1 - \rho_{\mathbf{u},\mathbf{v}}, \tag{2.23}$$

where

$$\rho_{\mathbf{u},\mathbf{v}} = \frac{\sum_{i=1}^{n}(u_i - \bar{\mathbf{u}})(v_i - \bar{\mathbf{v}})}{\sigma_{\mathbf{u}}\sigma_{\mathbf{v}}}$$

is the well-known Pearson correlation coefficient. As the Pearson correlation coefficient lies in the interval $[-1,1]$, the Pearson distance $0 \le \delta_p(\mathbf{u},\mathbf{v}) \le 2$. Pearson correlation coefficient $\rho_{\mathbf{u},\mathbf{v}} = 1$ indicates a perfect positive relationship, $\rho_{\mathbf{u},\mathbf{v}} = -1$ indicates a perfect negative relationship, and a $\rho_{\mathbf{u},\mathbf{v}} = 0$ indicates that no relationship exists. Thus the Pearson distance of two strongly related vectors is close to 2, and its value decreases when the relationship goes weaker.

2

It is assumed that a randomly chosen codeword $\mathbf{x} \in S$ is sent, and the received vector is given by

$$\mathbf{r} = a(\mathbf{x}+\mathbf{v}) + b\mathbf{1}, \tag{2.24}$$

where the scaling factor, $a \ne 1$, $a > 0$, is called *gain*, and $b \ne 0$ is an *offset*.

In Figure 2.4, we show an example of the channel model (2.24) when there is no gain mismatch, i.e., $a = 1$. Assume transmitting the codeword $\mathbf{x}_1$, the received vector (red dot) is corrupted by the noise (blue dash line) and distorted by the offset mismatch (red dash line). The MED will incorrectly decode the received vector into the codeword $\mathbf{x}_3$ since the received vector is within the subset where $\mathbf{x}_3$ is. In the next section, a detector based on the modified Pearson distance is shown to decode the received vector in this example successfully.

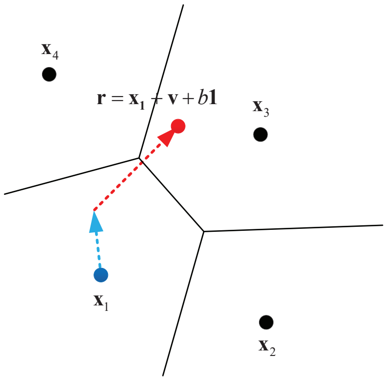


Figure 2.4: Graphic illustration of channel model (2.24) of four codewords.

An MPD chooses a codeword minimizing the Pearson distance between the received vector $\mathbf{r}$ and candidate codeword $\hat{\mathbf{x}} \in S$, that is,

$$\mathbf{x}_o = \arg\min_{\hat{\mathbf{x}} \in S} \delta_P(\mathbf{r},\hat{\mathbf{x}}). \tag{2.25}$$

An essential property of the Pearson distance is that it is invariant under separate changes in location and scale in the two variables. That is, we may transform $\mathbf{r}$ to $a + b\mathbf{r}$ and transform $\hat{\mathbf{x}}$ to $c + d\hat{\mathbf{x}}$, where a, b, c, and d are constants with $b, d > 0$, without changing the correlation coefficient, that is,

$$\delta_P(\mathbf{r},\hat{\mathbf{x}}) = \delta_P(a + b\mathbf{r}, c + d\hat{\mathbf{x}}). \tag{2.26}$$

Clearly, this property ensures that the detection outcome based on the Pearson distance (2.23) is intrinsically resistant to offset and gain mismatch.

An equivalent of the Pearson distance measure is

$$\begin{aligned}\delta_P(\mathbf{r},\hat{\mathbf{x}}) &= 1-\rho_{\mathbf{r},\hat{\mathbf{x}}}\\ &= 1-\frac{\sum_{i=1}^{n}(r_i-\bar{\mathbf{r}})(x_i-\bar{\hat{\mathbf{x}}})}{\sigma_{\mathbf{r}}\sigma_{\hat{\mathbf{x}}}}\\ &\equiv -\frac{\sum_{i=1}^{n} r_i(x_i-\bar{\hat{\mathbf{x}}})}{\sigma_{\hat{\mathbf{x}}}}+\frac{\bar{\mathbf{r}}\sum_{i=1}^{n}(x_i-\bar{\hat{\mathbf{x}}})}{\sigma_{\hat{\mathbf{x}}}}\\ &= -\frac{\sum_{i=1}^{n} r_i(x_i-\bar{\hat{\mathbf{x}}})}{\sigma_{\hat{\mathbf{x}}}},\end{aligned}$$

where the third equality has removed an irrelevant term in the minimization process and the last equation follows from $\bar{\hat{\mathbf{x}}} = \frac{1}{n}\sum_{i=1}^{n}\hat{x}_i$.

2.4.1. PEARSON CODES

The use of MPD requires that the code satisfies certain conditions since an MPD cannot distinguish between the words $\hat{\mathbf{x}}$ and $c_1\hat{\mathbf{x}}+c_2\mathbf{1}$, $c_1, c_2 \in \mathbb{R}$, $c_1 > 0$. It is further immediate, see (2.23), that the Pearson distance is undefined for codewords $\mathbf{x}$ with $\sigma_{\mathbf{x}} = 0$, for example, the all-one vector. Thus some codewords must be taken from a codebook $S \subseteq [q]^n$ that guarantees unambiguous detection, which type of codes are named *Pearson codes.*

The Pearson code introduced by Immink and Weber is defined as a set of codewords that can be uniquely decoded by an MPD. Let S be a codebook of chosen codewords $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ over the alphabet $[q]$, $q \geq 2$. Thus the codewords in a Pearson code must have these two properties:

- *Property 1:* If $\mathbf{x} \in S$ then $c_1\mathbf{x}+c_2\mathbf{1} \notin S$ for all $c_1, c_2 \in \mathbb{R}$ with $(c_1, c_2) \neq (1,0)$ and $c_1 > 0$.
- *Property 2:* $\mathbf{x} = (c, c, \ldots, c) \notin S$ for all $c \in \mathbb{R}$.

One example of Pearson codes called T-constrained codes [49], and the optimal Pearson codes [51] will be included in this subsection.

T-CONSTRAINED CODES

For an integer T, $1 \leq T \leq q$, the T*-constrained code* consists of q-ary n-length codewords where T reference symbols must appear at least once in a codeword. Denote a set of T-constrained codes as S_T. The cardinality of T-constrained codes equals

$$|S_T| = \sum_{i=0}^{T}(-1)^i\binom{T}{T-i}(q-i)^n, n \geq T. \tag{2.27}$$

We present two example sets of T-constrained codes: a binary 1-constrained code, denoted by S_1 and a binary 2-constrained code, denoted by S_2. The set S_1 has binary codewords where the symbol ‘0’ appears at least once, and the set S_2 contains binary codewords where both the symbols ‘0’ and ‘1’ appear at least once. The size of $|S_1|$ and $|S_2|$, respectively, equal

$$|S_1| = 2^n - 1 \tag{2.28}$$

and

$$|S_2| = 2^n - 2. \tag{2.29}$$

S_2 contains all sequences with all-one and all-zero codewords excluded, and S_1 contains all sequences with all-one codewords excluded. The 2-constrained code S_2 is a Pearson code as it satisfies Properties 1 and 2. Here we define the modified block code as the regular block code with all-one and all-zero codewords removed. The 1-constrained code S_1, excluding the all-one codeword, is not a Pearson code but can be used in minimum modified Pearson distance detection, which will be discussed in the next section.

OPTIMAL PEARSON CODES

A Pearson code with maximum possible size given the alphabet size q and length n is said to be *optimal*. Let $m(\mathbf{x})$ and $M(\mathbf{x})$ denote the smallest and largest value, respectively, among $x_i, 1 \le i \le n$ of $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Furthermore, in case $\mathbf{x}$ is not the all-zero word, let $GCD(\mathbf{x})$ denote the greatest common divisor of the x_i. For any $n, q \ge 2$, all sequences $\mathbf{x}$ in an optimal Pearson code, denoted by $P_{q,n}$, satisfy the following properties:

- $m(\mathbf{x}) = 0$;
- $M(\mathbf{x}) > 0$;
- $GCD(\mathbf{x}) = 1$.

It is evident that the binary 2-constrained code S_2 of size $2^n - 2$ is the optimal Pearson code. However, for $q > 3$, the 2-constrained sets such that reference symbols '0' and '1' appear at least once are not optimal Pearson codes when $n > 2$.

The cardinality (number) of a q-ary optimal Pearson code of length n equals

$$\left|P_{q,n}\right| = \sum_{d=1}^{q-1} \mu(d) \left(\left(\left\lfloor \frac{q-1}{d} \right\rfloor + 1 \right)^n - \left\lfloor \frac{q-1}{d} \right\rfloor^n - 1 \right), \tag{2.30}$$

where $\mu(d)$ is the Mobius function. For a positive integer d, $\mu(d)$ is defined to be 0 if d is divisible by the square of a prime, otherwise $\mu(d) = (-1)^k$ where k is the number of (distinct) prime divisors of d.

2.4.2. PERFORMANCE ANALYSIS

We will now examine the error performance of the Pearson distance based detection scheme. The receiver uses the Pearson distance (2.23) for the evaluation of the received word, where we assume that $\mathbf{x} \in S_2$ is sent, and received as $\mathbf{r} = \mathbf{x} + \mathbf{v}$. S_2 is a codebook wherein both the two symbols '0' and '1' appear at least once. The receiver errs if there is at least one codeword $\hat{\mathbf{x}} \neq \mathbf{x}, \hat{\mathbf{x}} \in S_2$, such that

$$\delta_P(\mathbf{r}, \hat{\mathbf{x}}) < \delta_P(\mathbf{r}, \mathbf{x}). \tag{2.31}$$

2

After some manipulation of (2.23), we may use an equivalent of the Pearson distance measure and obtain

$$-\sum_{i=1}^{n} r_i\left(\frac{\hat{x}_i - \bar{\hat{x}}}{\sigma_{\hat{\mathbf{x}}}}\right) < -\sum_{i=1}^{n} r_i\left(\frac{x_i - \bar{x}}{\sigma_{\mathbf{x}}}\right).$$

We substitute $r_i = x_i + v_i$, and obtain

$$\sum_{i=1}^{n} (x_i + v_i)(m_i - \hat{m}_i) < 0, \tag{2.32}$$

where

$$m_i = \frac{x_i - \bar{x}}{\sigma_{\mathbf{x}}}$$

and

$$\hat{m}_i = \frac{\hat{x}_i - \bar{\hat{x}}}{\sigma_{\hat{\mathbf{x}}}}.$$

The left-hand side of inequality (2.32) is a stochastic variable with distribution $N(\alpha_P, \beta_P\sigma^2)$, where

$$\alpha_P = \sum_{i=1}^{n} x_i(m_i - \hat{m}_i)$$

and

$$\beta_P = \sum_{i=1}^{n} (m_i - \hat{m}_i)^2.$$

Since

$$\sum_{i=1}^{n} m_i = \sum_{i=1}^{n} \hat{m}_i = 0,$$

we have

$$\begin{aligned} \alpha_P &= \sum_{i=1}^{n} x_i(m_i - \hat{m}_i) = \sum_{i=1}^{n} (x_i + \bar{x})(m_i - \hat{m}_i) \\ &= \sum_{i=1}^{n} (x_i + \bar{x})\left(\tfrac{x_i - \bar{x}}{\sigma_{\mathbf{x}}} - \tfrac{\hat{x}_i - \bar{\hat{x}}}{\sigma_{\hat{\mathbf{x}}}}\right) = \sigma_{\mathbf{x}}(1 - \rho_{\mathbf{x},\hat{\mathbf{x}}}). \end{aligned} \tag{2.33}$$

In a similar fashion, we find

$$\beta_P = 2(1 - \rho_{\mathbf{x},\hat{\mathbf{x}}}). \tag{2.34}$$

Define the square of the distance, $d_P^2(\mathbf{x}, \hat{\mathbf{x}})$, between the codewords $\mathbf{x}$ and $\hat{\mathbf{x}}$ by

$$d_P^2(\mathbf{x}, \hat{\mathbf{x}}) = \frac{4\alpha_P^2}{\beta_P} = 2\sigma_x^2(1 - \rho_{\mathbf{x},\hat{\mathbf{x}}}). \tag{2.35}$$

The WER over all coded sequences $\mathbf{x}$ is, due to the union bound, bounded from above by

$$WER < \frac{1}{|S_2|} \sum_{\mathbf{x} \in S_2} \sum_{\mathbf{x} \neq \hat{\mathbf{x}}} Q\left(\frac{d_P(\mathbf{x}, \hat{\mathbf{x}})}{2\sigma}\right). \tag{2.36}$$

Then, for asymptotically large signal-to-noise-ratio's, i.e. for $\sigma \ll 1$, the word error rate is over-bounded by

$$WER < N_P Q\left(\frac{d_{\min,P}}{2\sigma}\right), \tag{2.37}$$

where N_P is the average number of nearest neighbors at minimum distance

$$d^2_{\min,P} = \min_{\mathbf{x},\hat{\mathbf{x}}\in S_2,\mathbf{x}\neq\hat{\mathbf{x}}} d^2_P(\mathbf{x},\hat{\mathbf{x}}) \tag{2.38}$$

for every codeword.

2.5. MINIMUM MODIFIED PEARSON DISTANCE BASED DETECTION (MMPD)

As shown in [49], a simpler Pearson distance-based criterion leading to the same result in the minimization process reads

$$\delta'_P(\mathbf{r},\hat{\mathbf{x}}) = \sum_{i=1}^{n}(r_i - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2, \tag{2.39}$$

if there is no gain mismatch, $a = 1$, in channel model (2.24). That is,

$$\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}, \tag{2.40}$$

where the offset mismatch is constant within one codeword length and may vary word by word. We can calculate the δ'_p distance to an arbitrary codeword $\hat{\mathbf{x}}$ in the code as

$$\begin{aligned}\delta'_P(\mathbf{r},\hat{\mathbf{x}}) &= \sum_{i=1}^{n}(x_i + v_i + b - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2 \\ &= \sum_{i=1}^{n}(x_i + v_i - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2 + 2b\sum_{i=1}^{n}(x_i + v_i - \hat{x}_i + \bar{\hat{\mathbf{x}}}) + b^2 \\ &= \sum_{i=1}^{n}(x_i + v_i - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2 + 2b\sum_{i=1}^{n}(x_i + v_i) + b^2.\end{aligned} \tag{2.41}$$

Note that the last two parts in (2.41) are independent of the choice of $\hat{\mathbf{x}}$. So it is shown that the MMPD is intrinsically resistant to the channel offset $b\mathbf{1}$.

After some manipulation of (2.39), we may write down an equivalent of the modified Pearson distance measure, namely

$$\sum_{i=1}^{n}((r_i - \bar{\mathbf{r}}) - (x_i - \bar{\hat{\mathbf{x}}}))^2. \tag{2.42}$$

The equivalence of (2.42) follows from

$$\begin{aligned}\sum_{i=1}^{n}((r_i - \bar{\mathbf{r}}) - (x_i - \bar{\hat{\mathbf{x}}}))^2 &= \sum_{i=1}^{n}(r_i - (x_i - \bar{\hat{\mathbf{x}}}))^2 - \sum_{i=1}^{n}2\bar{\mathbf{r}}(r_i - (x_i - \bar{\hat{\mathbf{x}}})) + n\bar{\mathbf{r}}^2 \\ &= \sum_{i=1}^{n}(r_i - (x_i - \bar{\hat{\mathbf{x}}}))^2 - n\bar{\mathbf{r}}^2 \\ &\equiv \delta'_P(\mathbf{r},\hat{\mathbf{x}}),\end{aligned}$$

where the second equation follows from $\bar{\hat{\mathbf{x}}} = \frac{1}{n}\sum_{i=1}^{n}\hat{x}_i$, and in the last equivalence we have ignored an irrelevant term $n\bar{\mathbf{r}}^2$ in the minimization process. Note that δ'_P can not distinguish two candidate codewords with the same normalized values by definition (2.39).

Our example of four codewords is illustrated for MMPD in Figure 2.5. Actually, an interpretation of MMPD from (2.42) is that the vectors in $\mathbb{R}^n$ are mapped to vectors in the hyperplane $\{\mathbf{y} \in \mathbb{R}^n : \bar{\mathbf{y}} = 0\}$ by orthogonal projection

$$\begin{bmatrix} 1-\frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & \cdots & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \cdots & 1-\frac{1}{n} \end{bmatrix},$$

i.e., in the direction $\mathbf{1}$, and that the squared Euclidean distance between these projections is calculated. As a consequence, the transmitted codeword $\mathbf{x}_1$ is successfully decoded by MMPD, since its projection is the closest to the projection of the received vector.

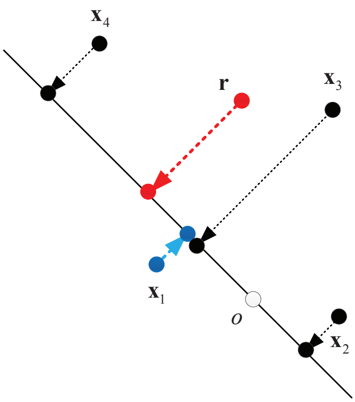


Figure 2.5: Graphic illustration of the minimum modified Pearson distance based detection of four codewords.

2.5.1. PERFORMANCE ANALYSIS

We will now examine the error performance of the modified Pearson distance based detection scheme. The receiver uses the modified Pearson distance (2.39) for the evaluation of the received word, where we assume that $\mathbf{x} \in S_1$ is sent, and received as $\mathbf{r} = \mathbf{x} + \mathbf{v}$. S_1 is a codebook wherein the symbol '0' appears at least once. Note that S_1 is not a Pearson code, but it is feasible enough to the MMPD. The receiver errs if there is at least one codeword $\hat{\mathbf{x}} \neq \mathbf{x}, \hat{\mathbf{x}} \in S_1$, such that

$$\delta'_P(\mathbf{r}, \hat{\mathbf{x}}) < \delta'_P(\mathbf{r}, \mathbf{x}). \tag{2.43}$$

Define $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$ and $\bar{e} = \bar{x} - \bar{\hat{x}}$, so that

$$2\sum_{i=1}^{n} v_i(e_i - \bar{e}) + \sum_{i=1}^{n} (e_i - \bar{e})^2 < 0. \tag{2.44}$$

The left-hand side of inequality (2.44) is a stochastic variable with distribution $N(\alpha_{MP}, \beta_{MP}\sigma^2)$, where

$$\alpha_{MP} = \sum_{i=1}^{n} (e_i - \bar{e})^2$$

and

$$\beta_{MP} = 4\sum_{i=1}^{n} (e_i - \bar{e})^2.$$

We define the square of the distance, $d^2_{MP}(\mathbf{x}, \hat{\mathbf{x}})$, between the codewords $\mathbf{x}$ and $\hat{\mathbf{x}}$ by

$$d^2_{MP}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{4\alpha^2_{MP}}{\beta_{MP}} = \sum_{i=1}^{n} (e_i - \bar{e})^2. \tag{2.45}$$

The word error rate (WER) over all coded sequences $\mathbf{x}$ is, due to the union bound, bounded from above by

$$WER < \frac{1}{|S_1|}\sum_{\mathbf{x}\in S_1}\sum_{\mathbf{x}\neq\hat{\mathbf{x}}} Q\left(\frac{d_{MP}(\mathbf{x}, \hat{\mathbf{x}})}{2\sigma}\right). \tag{2.46}$$

Then, for asymptotically large signal-to-noise-ratios, i.e. for $\sigma \ll 1$, the word error rate is upper-bounded by

$$WER < N_{MP} Q\left(\frac{d_{\min,MP}}{2\sigma}\right), \tag{2.47}$$

where N_{MP} is the average number of nearest neighbors at minimum distance

$$d^2_{\min,MP} = \min_{\mathbf{x},\hat{\mathbf{x}}\in S_1, \mathbf{x}\neq\hat{\mathbf{x}}} d^2_{MP}(\mathbf{x}, \hat{\mathbf{x}}) = \min_{\mathbf{e}\neq 0} \sum_{i=1}^{n} (e_i - \bar{e})^2 \tag{2.48}$$

for every codeword.

As we introduced, MED and MPD are two ideal methods for channels with additive white Gaussian noise and gain and/or offset mismatch, respectively. We compare these two schemes for channel model (2.10), $\mathbf{r} = a(\mathbf{x}+\mathbf{v}) + b\mathbf{1}$, considering both noise and offset mismatch with the same code. Because Pearson detection is dedicated to Pearson code, we use the modified (7,4,3) Hamming code. The modified (7,4,3) Hamming code is the Hamming code $\mathcal{H}_3$ with all-zero and all-one codewords excluded.

We first show the WER theoretical upper bounds and simulation curves of MED and MPD in Figure 2.6. Simulation curves are obtained for the ideally matched channels with only Gaussian noise. Each result is the average value after 10,000 times simulation. The error performances are computed by upper bounds (2.22) and (2.37).

Two factors of a codebook play crucial roles in detectors' performance. One is the minimum distance, and another is the number of nearest neighbors at this distance. Though the former is of utmost importance to the WER performance, the latter could also play an important role. For modified (7,4,3) Hamming code, the average number of nearest neighbors N_E is 6, and the minimum squared distance $d^2_{\min,E}$ is 3. We can see that the computer simulation (red line with diamonds) matches theoretical computation (red dashed line) very well. For MPD, the average number of nearest neighbors N_P is 6, and the minimum squared distance $d^2_{\min,P}$ is 2.86. The simulation results (blue line with asterisks) match the upper bound (blue dashed line) in high SNR.

2

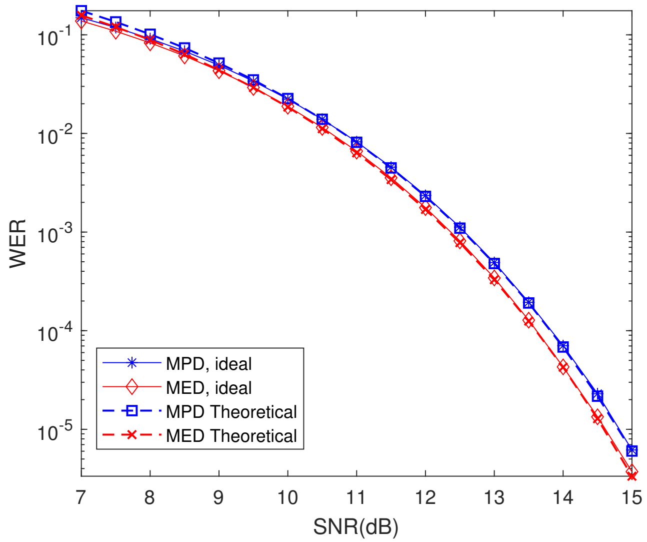


Figure 2.6: Theoretical upper bounds and simulated word error rate (WER) as a function of signal to noise ratio (SNR) for the modified (7,4,3) Hamming code. The theoretical performances are computed by upper bounds (2.22) and (2.37), and simulation curves are obtained for the ideally matched channels ($a = 1, b = 0$) with only Gaussian noise.

Figure 2.7 shows the computer simulation comparison of MPD and MED in the ideally matched case and the situation with $a = 1.1$ and $b = 0.3$. Results show that the MED outperforms the MPD in the ideally matched case. The reason is that the minimum Euclidean squared distance $d^2_{\min,E}$ of the modified (7,4,3) Hamming code is larger than its minimum Pearson squared distance $d^2_{\min,P}$ with the same average number of nearest neighbors. When there is the gain and offset mismatch, we see that the performance of MED drops dramatically, while the performance of MPD almost stays the same. It supports our argument on the intrinsic immunity of MPD to the gain and offset mismatch.

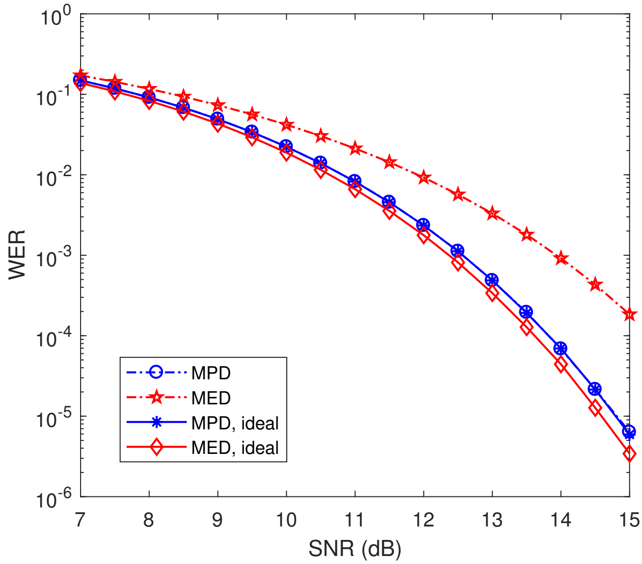


Figure 2.7: Simulated word error rate (WER) as a function of signal to noise ratio (SNR) for the modified (7,4,3) Hamming code in the ideally matched case ($a = 1, b = 0$) and the situation with $a = 1.1$ and $b = 0.3$.

3

Noisy Channels with Unknown Offset Mismatch

We have the duty of formulating, of summarizing, and of communicating our conclusions, in intelligible form, in recognition of the right of other free minds to utilize them in making their own decisions.

R. Fisher

In storage and communication systems, noise is not the only disturbance during data transmission. Sometimes the error performance can also be seriously degraded by offset mismatch. In this chapter, we focus on noisy channels with unknown offset mismatch, where an offset is in the all-one direction, i.e., $b\mathbf{1}$. *The maximum likelihood (ML) decoding strategy for such channels is considered in two situations, where both noise and offset mismatch are: (1) bounded; and (2) Gaussian distributed.*

Firstly, we present an ML criterion when assuming bounded noise and offset. In particular, for Euclidean distance-based decoding, modified Pearson distance-based decoding, and

The material in this chapter has appeared in

- Section 3.2

  R. Bu and J. H. Weber, Maximum likelihood decoding for channels with uniform noise and offset, *Proc. of the Thirty-ninth WIC Symposium on Information Theory and Signal Processing in the Benelux*, Enschede, the Netherlands, May 2018.

  R. Bu and J. H. Weber, Decoding criteria and zero wer analysis for channels with bounded noise and offset, *Proc. of IEEE International Conference on Communications (ICC)*, Shanghai, China, May, 2019.

- Section 3.4

  R. Bu and J. H. Weber, Decoding of concatenated codes for noisy channels with unknown offset, *Proc. of the Forty-first WIC Symposium on Information Theory and Signal Processing in the Benelux*, the Netherlands, May, 2021.

ML decoding, bounds are determined on the magnitudes of the noise and offset intervals, leading to a word error rate equal to zero.

Secondly, for Gaussian distributed noise and offset, the ML criterion is shown to be a weighted average of the well-known Euclidean and Pearson norms, where the weighting coefficients depend on the ratio of the noise and offset variances. Based on this work, we further propose a concatenated scheme and its corresponding decoding algorithm in case of Gaussian noise and offset mismatch. Simulation results demonstrate the effectiveness of the proposed codes and the error correction ability of the decoding algorithm in the presence of both noise and offset mismatch.

3.1. Introduction

It is usually found that noise, which leads to unpredictable stochastic errors, is a critical issue for data transmission and storage systems. However, that also other physical factors may hamper the reliability of the stored data. For example, in Flash memories, the number of electrons of a cell decreases with time, and some cells become defective over time [9]. In the digital optical recording, fingerprints and scratches on the surface of discs result in offset variations of the retrieved signal [65].

In this chapter, a simple and basic model of the offset mismatch is considered. We assume that the unknown offset mismatch is constant within one codeword block length and may vary block by block. That is,

$$\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}, \tag{3.1}$$

where $\mathbf{x} = (x_1, ..., x_n)$ is the transmitted codeword from a codebook $S \subseteq \mathbb{R}^n$, and a uniform distribution is assumed, i.e., all codewords are equally likely to be transmitted. $\mathbf{v} = (v_1, ..., v_n)$ is the noise vector, where v_i are identically independently distributed with mean 0 and standard deviation σ, and $\mathbf{r}$ is the received vector. We denote the probability density function of the noise vector as $\chi(\mathbf{v})$. We can also assume that b has a specified probability density function ζ with mean μ and variance β^2. Since a receiver can subtract $\mu\mathbf{1}$ from $\mathbf{r}$ in case the expected offset value μ is not equal to zero, we may assume $\mu = 0$ without loss of generality.

There are usually two approaches to address the physical-related offset issues. One approach uses pilot sequences to estimate the unknown channel offset [27], which is often considered too expensive concerning redundancy. Other approaches are error-correcting techniques. Up to now, various coding techniques have been applied to alleviate the detection in case of channel mismatch, specifically rank modulation [32], balanced codes [34], and composition check codes [38]. These methods are often considered too expensive in terms of redundancy and complexity.

The so-called *maximum likelihood (ML) principle* says that the decoder should decode a word $\mathbf{r}$ as a codeword $\mathbf{x}_o$ that has the maximum likelihood to $\mathbf{r}$. A minimum Euclidean distance based detection (MED) is an ML decoder for channels with Gaussian noise. Since the retrieved data value has been offset in channels, the MED will be biased or grossly inaccurate. Immink and Weber [49] showed that detectors minimizing the modified Pearson distance (MMPD) have immunity to offset mismatch. The MMPD is an ML decoder when there exists the offset but not the noise in a channel. Further, in

[56] and [62] a decoder was proposed based on minimizing a weighted sum of Euclidean and Pearson distances, which is proved to be optimal for channels with Gaussian noise and offset mismatch.

This chapter explores decoding criteria for channels with bounded noise and bounded offset mismatch. Specifically, we consider MED, MMPD, and ML decoding. Most importantly, we investigate, for each of these decoders, under which constraints zero Word Error Rate (WER) performance can be achieved.

In addition, for Gaussian distributed noise and offset mismatch, we derive the ML criterion considering successive channel outputs, which includes the results in [56, 62] as its particular case. A concatenated coding scheme is proposed in the case of Gaussian noise and offset mismatch. A novel decoding algorithm for the concatenated scheme is proposed, aiming to exploit its error correction potential better. The concatenation is between a Reed-Solomon (RS) code and a certain coset of a block code proposed in [66]. The modified Pearson distance detection is used to decode the inner code. Its output will be given to a two-stage hybrid decoding algorithm for the outer RS code. Simulation results show that a considerable coding gain is achieved with an even higher code rate compared with non-concatenated codes.

The remainder of this chapter is organized as follows. We explore an ML decoding criterion for channels with bounded noise and offset mismatch in Section 3.2. Conditions to achieve zero WER for MED, MMPD, and ML detectors are derived in Section 3.2.3. In Section 3.3, the ML decoding method is investigated when the noise and offset are Gaussian distributed. Further, a concatenated coding scheme and its decoding algorithm are proposed in Section 3.4 for channels with Gaussian noise and offset. Simulation results for each situation are given, confirming the results.

3.2. Maximum Likelihood Decoding for Channels with Bounded Noise and Offset

An increasing number of studies focus on an essential class of non-Gaussian stochastic processes: bounded noise, which is motivated by the fact that the Gaussian stochastic process is an inadequate mathematical model of the physical world because it is unbounded [67, 68]. Moreover, in many relevant cases, especially in Flash memory, the impact of parameters (such as charge leakage) on the retrieved data value should not be arbitrarily large. Consequently, not taking into account the bounded nature of stochastic variations may lead to impracticable model-based inferences.

In this section, we explore ML decoding criterion for channels (3.1) with bounded noise and offset mismatch. Most importantly, we investigate, for MED, MMPD, and ML decoders, under which constraints zero WER performance can be achieved. We should stress that zero WER performance is achieved without assumptions of specific distributions for the bounded noise and offset. If more information is available about these distributions, other decoding methods might be appropriate. For example, with the assumption of uniformly distributed noise and offset, subtraction can be used naturally instead of integral in the decoding as shown in Subsection 3.2.2.

3.2.1. DECODING CRITERION

The values v_i in the noise vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ are independently and identically distributed with probability density function ϕ, leading to a probability density function $\chi(\mathbf{v}) = \Pi_{i=1}^{n} \phi(v_i)$ for $\mathbf{v}$. We assume that the noise values are restricted to a certain range. More specifically, ϕ only takes non-zero values on an interval $(-\sigma, \sigma)$, where $\sigma > 0$. Hence, $-\sigma < v_i < \sigma$ for all i. For a codeword $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n) \in S$, we define its noise environment as

$$U_{\hat{\mathbf{x}}} = \{\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{R}^n : \hat{x}_i - \sigma < u_i < \hat{x}_i + \sigma\}. \tag{3.2}$$

For the offset b, we assume that it has a probability density function ζ, which only takes non-zero values on an interval (γ, η). Hence, $\gamma < b < \eta$. Since the receiver can subtract $\frac{\eta+\gamma}{2}\mathbf{1}$ from $\mathbf{r}$ if the offset range is not symmetric around zero, we may assume without loss of generality that the offset is within the range $(-\beta, \beta)$, where $\beta = (\eta - \gamma)/2$, which we will do throughout the rest of this section. We define

$$L_{\mathbf{r}} = \{\mathbf{r} - t\mathbf{1} : t \in (-\beta, \beta)\} \tag{3.3}$$

for a vector $\mathbf{r} \in \mathbb{R}^n$.

In order to achieve ML decoding, we need to choose the codeword of maximum likelihood given the received vector. Assuming all codewords are equally likely, this is equivalent to maximizing the probability density value of the received vector $\mathbf{r}$ given the candidate codeword $\hat{\mathbf{x}}$. With (3.1), we should thus maximize

$$\psi(\mathbf{r} - \hat{\mathbf{x}}) = \int_{-\infty}^{\infty} \chi(\mathbf{r} - \hat{\mathbf{x}} - b\mathbf{1})\zeta(b)\,db \tag{3.4}$$

over all candidate codewords $\hat{\mathbf{x}}$, where χ and ζ are the probability density functions of the noise and offset, respectively. χ and ζ can be any distribution as long as they are restricted to the indicated intervals. In Subsection 3.2.2, we will show the results assuming specific distributions. Because both ϕ and ζ are bounded between a minimum and a maximum, the integral range of (3.4) is the intersection of the line segment $L_{\mathbf{r}}$ to the noise environment $U_{\hat{\mathbf{x}}}$ of a codeword $\hat{\mathbf{x}}$. Thus ML decoding is equivalent to choosing codeword $\hat{\mathbf{x}}$ which has the largest integral value over the intersection interval.

Note from (3.2) and (3.3) that a point $\mathbf{r} - t\mathbf{1}$ of $L_{\mathbf{r}}$ is in $U_{\hat{\mathbf{x}}}$ if and only if t satisfies

$$\begin{cases} r_i - \hat{x}_i - \sigma < t < r_i - \hat{x}_i + \sigma, \forall i = 1, \ldots, n, \\ -\beta < t < \beta. \end{cases} \tag{3.5}$$

From this observation, we find that (3.4) equals

$$\begin{cases} \int_{t_1(\mathbf{r},\hat{\mathbf{x}})}^{t_0(\mathbf{r},\hat{\mathbf{x}})} \chi(\mathbf{r} - \hat{\mathbf{x}} - b\mathbf{1})\zeta(b)\,db & \text{if } t_0(\mathbf{r},\hat{\mathbf{x}}) > t_1(\mathbf{r},\hat{\mathbf{x}}), \\ 0 & \text{otherwise}, \end{cases} \tag{3.6}$$

where

$$\begin{aligned} t_0(\mathbf{r},\hat{\mathbf{x}}) &= \min\left(\{r_i - \hat{x}_i + \sigma \,|\, i = 1, \ldots, n\} \cup \{\beta\}\right), \\ t_1(\mathbf{r},\hat{\mathbf{x}}) &= \max\left(\{r_i - \hat{x}_i - \sigma \,|\, i = 1, \ldots, n\} \cup \{-\beta\}\right). \end{aligned} \tag{3.7}$$

$t_0(\mathbf{r},\hat{\mathbf{x}})$ and $t_1(\mathbf{r},\hat{\mathbf{x}})$ represent the maximum and the minimum values of a point in intersecting line segment, respectively. If there is an intersection, it corresponds to the parametric interval $t_1(\mathbf{r},\hat{\mathbf{x}}) < t_0(\mathbf{r},\hat{\mathbf{x}})$. Note that $\psi(\mathbf{r}-\hat{\mathbf{x}})$ is immediately set to 0 without further computation if $t_1(\mathbf{r},\hat{\mathbf{x}}) > t_0(\mathbf{r},\hat{\mathbf{x}})$. For instance, when the noise and the offset are uniformly distributed, the ML decoding is maximizing the length of the intersection between the line segment $L_{\mathbf{r}}$ and the noise environment $U_{\hat{\mathbf{x}}}$ given the candidate codeword $\hat{\mathbf{x}}$.

3.2.2. CASE STUDIES

Here, we consider several noise and offset distributions. Simulated WER results are shown for the (3,2) parity check code

$$S^* = \{(0,0,0),(1,1,0),(1,0,1),(0,1,1)\} \tag{3.8}$$

in combination with different decoders. This simple codebook is used to demonstrate some important WER characteristics. Code book construction as such is referred to [54].

UNIFORM NOISE AND OFFSET

The uniform distribution is the most-commonly used for bounded random variables. Let the probability density function of a random variable which is uniformly distributed on the interval (τ_1,τ_2) be denoted by $\mathcal{U}(\tau_1,\tau_2)$. The uniform distribution $\mathcal{U}(\tau_1,\tau_2)$ has probability density function

$$\mathcal{U}(x) = \begin{cases} \frac{1}{\tau_2-\tau_1} & \text{if } \tau_1 < x < \tau_2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

Hence, for the noise we assume $v_i \sim \mathcal{U}(-\sigma,\sigma)$ and for the offset $b \sim \mathcal{U}(-\beta,\beta)$.

Because of the uniform nature of both ϕ and ζ, ML criterion (3.6) is tantamount to choosing a codeword $\hat{\mathbf{x}}$ for which the noise environment $U_{\hat{\mathbf{x}}}$ has the largest intersection with the line segment $L_{\mathbf{r}}$. Therefore, define the intersection distance, $\mathrm{ISD}(\mathbf{r},\hat{\mathbf{x}})$, between $\mathbf{r}$ and $\hat{\mathbf{x}}$ as the length of the intersection between the noise environment $U_{\hat{\mathbf{x}}}$ and the line segment $L_{\mathbf{r}}$. The most likely candidate codeword $\mathbf{x}_o$ for a received vector has the largest intersection distance, that is

$$\mathbf{x}_o = \arg\max_{\hat{\mathbf{x}}\in S} \mathrm{ISD}(\mathbf{r},\hat{\mathbf{x}}). \tag{3.10}$$

We can express the intersection distance between $\mathbf{r}$ and $\hat{\mathbf{x}}$ as

$$\mathrm{ISD}(\mathbf{r},\hat{\mathbf{x}}) = \sqrt{n(\max\{t_0(\mathbf{r},\hat{\mathbf{x}})-t_1(\mathbf{r},\hat{\mathbf{x}}),0\})^2}. \tag{3.11}$$

Note that maximizing $\mathrm{ISD}(\mathbf{r},\hat{\mathbf{x}})$ is equivalent to maximizing a simplified measure

$$\mathrm{ISD}'(\mathbf{r},\hat{\mathbf{x}}) = \max\{t_0(\mathbf{r},\hat{\mathbf{x}})-t_1(\mathbf{r},\hat{\mathbf{x}}),0\}, \tag{3.12}$$

i.e., choosing the codeword $\hat{\mathbf{x}}$ for which the part of the line segment $L_{\mathbf{r}}$ that is within $U_{\hat{\mathbf{x}}}$ is largest.

Simulated WER results for the (3,2) parity check code and various values of σ and β are shown in Figures 3.1-3.3 for MED, MMPD, and ML decoders, respectively.

In Figure 3.1, we observe that the performance of the MED gets worse with increasing values of σ and/or β. In Figure 3.2, the curves for different values of β overlap because

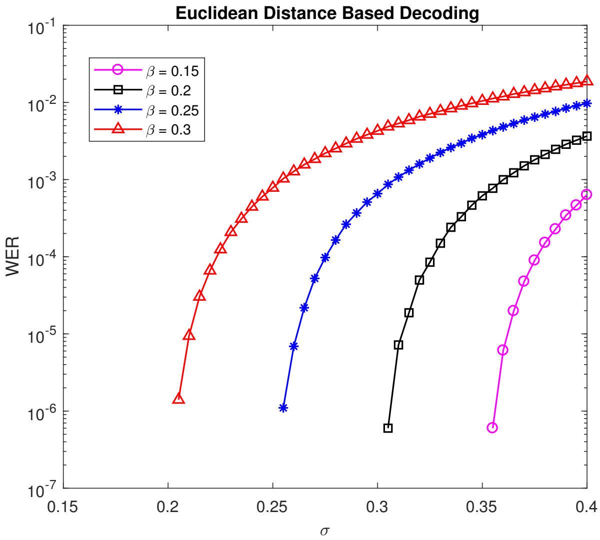


Figure 3.1: Simulated WER of Euclidean distance based decoding for codebook S^* on channels with uniform noise $v_i \sim \mathscr{U}(-\sigma,\sigma)$ and uniform offset $b \sim \mathscr{U}(-\beta,\beta)$.

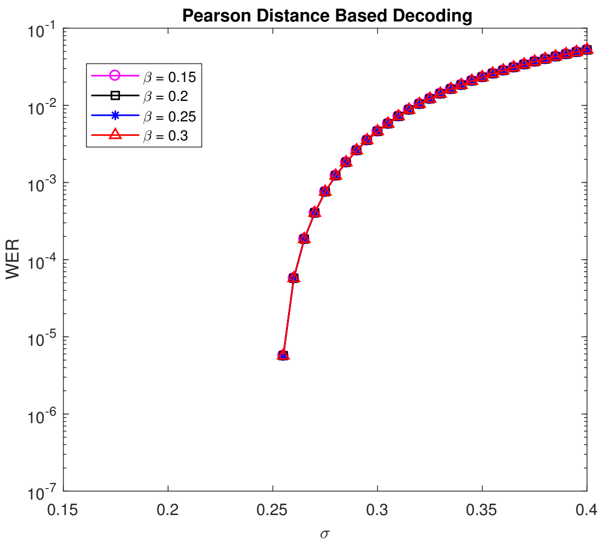


Figure 3.2: Simulated WER of Pearson distance based decoding for codebook S^* on channels with uniform noise $v_i \sim \mathscr{U}(-\sigma,\sigma)$ and uniform offset $b \sim \mathscr{U}(-\beta,\beta)$.

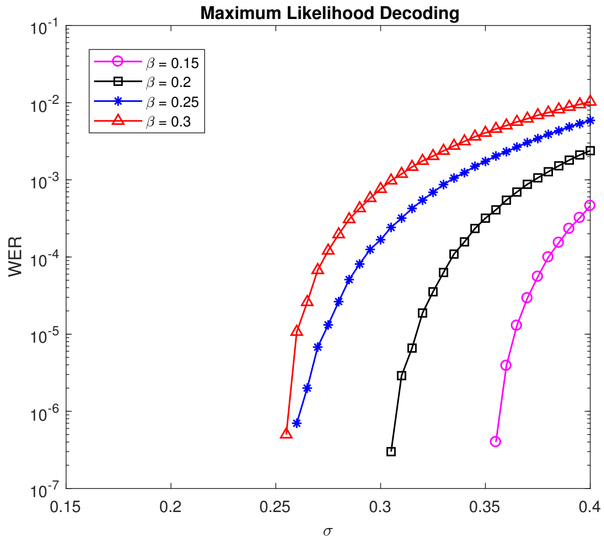


Figure 3.3: Simulated WER of ML decoding for codebook S^* on channels with uniform noise $v_i \sim \mathcal{U}(-\sigma,\sigma)$ and uniform offset $b \sim \mathcal{U}(-\beta,\beta)$.

of the MMPD's intrinsic immunity to offset mismatch. In Figure 3.3, we can observe that the performance of the MED is close to ML performance when $\beta = 0.15$ The performance of the MMPD is close to ML performance when $\beta = 0.30$.

Most interestingly, for MED, WER approaches zero if $\sigma \leq 1/2 - \beta$, while for MMPD, it happens when $\sigma < 1/4$. WER approaches zero for the ML decoding if $\sigma \leq 1/4$ or $\sigma \leq 1/2 - \beta$, i.e., $\sigma \leq \max\{1/4, 1/2-\beta\}$. Indeed, we observe in Figure 3.3 a zero WER for $\sigma \leq 0.35$ if $\beta = 0.15$, for $\sigma \leq 0.30$ if $\beta = 0.20$, and for $\sigma \leq 0.25$ if $\beta = 0.25$ or $\beta = 0.30$. We will show in Section 3.2.3 that, for all decoders under consideration, a WER of zero is achieved if the magnitudes of the noise and offset intervals satisfy certain conditions.

UNIFORM NOISE AND VARIOUS OFFSET DISTRIBUTIONS

Here, we consider uniform noise again, but various options for the offset distribution. In particular, $v_i \sim \mathcal{U}(-0.3, 0.3)$, while the offset is (i) uniform, i.e., $b \sim \mathcal{U}(-\beta,\beta)$, as in the previous case, (ii) triangular, i.e., $b \sim \mathcal{T}(-\beta, 0, \beta)$, as specified next, or (iii) Gaussian with mean zero and variance β^2, i.e., $b \sim \mathcal{N}(0, \beta^2)$. The last option is included for comparison purposes. The triangular distribution $\mathcal{T}(-\beta, 0, \beta)$ has probability density function

$$\mathcal{T}(x) = \begin{cases} \frac{1}{\beta}(1 - \frac{1}{\beta}|x|) & \text{if } -\beta < x < \beta, \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

In Figures 3.4-3.6 we present WER results for the example code S^* for the three offset options under consideration. For a fair comparison, we present the WER as a function of the standard deviation of the offset.

In general, note that the WER of MMPD has the same constant value for all cases

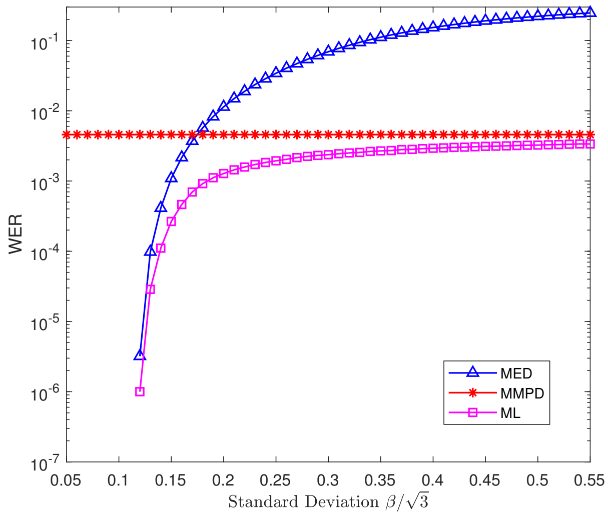


Figure 3.4: Simulated WER for codebook S^* on channels with uniform noise $v_i \sim \mathscr{U}(-0.3, 0.3)$ and uniform offset $b \sim \mathscr{U}(-\beta, \beta)$.

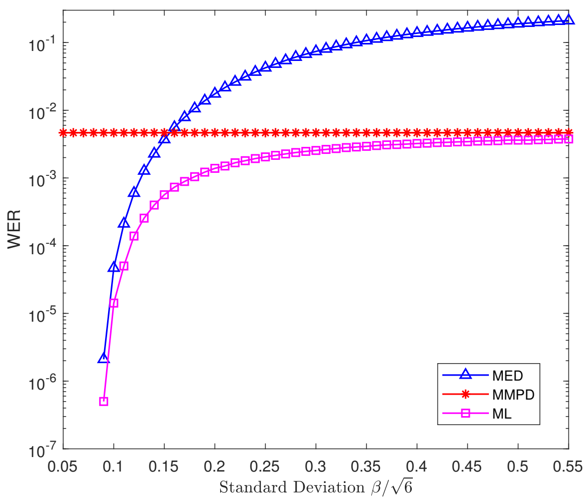


Figure 3.5: Simulated WER for codebook S^* on channels with uniform noise $v_i \sim \mathscr{U}(-0.3, 0.3)$ and triangular offset $b \sim \mathscr{T}(-\beta, 0, \beta)$.

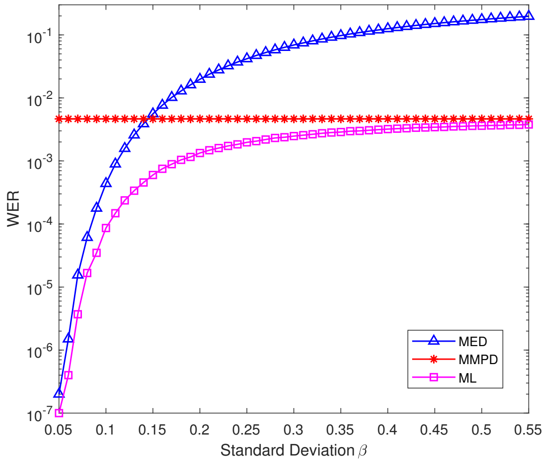


Figure 3.6: Simulated WER for codebook S^* on channels with uniform noise $v_i \sim \mathcal{U}(-0.3, 0.3)$ and Gaussian offset $b \sim \mathcal{N}(0, \beta^2)$.

since it does not depend on the offset, i.e., it is entirely determined by the noise boundary $\sigma = 0.3$. It is close to ML performance in case of large standard deviations. The performance of MED is close to ML performance for small standard deviations. For medium standard deviations, ML decoding outperforms both MED and MMPD in all three cases.

We also observe in Figure 3.4 that the WERs of MED and ML decoders approach zero if the standard deviation $\beta/\sqrt{3}$ of the uniform offset distribution is at most 0.12, and in Figure 3.5 that the WER approaches zero if the standard deviation $\beta/\sqrt{6}$ of the triangular offset distribution is at most 0.08. On the other hand, we see in Figure 3.6 that for Gaussian offset, zero WER can only be achieved by extremely small noise, as expected, due to the unbounded nature of the Gaussian distribution. In the following subsection, we will analyze the zero WER constraints for different detectors.

3.2.3. ZERO ERROR ANALYSIS

This subsection shows that, for all decoders under consideration, a WER of zero is achieved if the magnitudes of the noise and offset intervals satisfy certain conditions.

MINIMUM EUCLIDEAN DISTANCE BASED DETECTION

The Euclidean decoder can achieve zero WER for channels with bounded noise and offset when $\sigma + \beta$ is sufficiently small, as shown in the following result.

Theorem 1. *If the noise and offset are restricted to the intervals* $(-\sigma, \sigma)$ *and* $(-\beta, \beta)$, *respectively, with*

$$\sigma + \beta \leq \min_{\mathbf{s},\mathbf{c}\in S, \mathbf{s}\neq\mathbf{c}} \left(\frac{\sum_{i=1}^{n} (s_i - c_i)^2}{2\sum_{i=1}^{n} |s_i - c_i|} \right), \tag{3.14}$$

then the Euclidean decoder achieves a WER equal to zero.

Proof. Assume that $\mathbf{x} \in S$ is sent and $\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}$ is received. Then, for all codewords $\hat{\mathbf{x}} \neq \mathbf{x}$, it holds that

$$\begin{aligned}
&\delta_E(\mathbf{r},\hat{\mathbf{x}}) - \delta_E(\mathbf{r},\mathbf{x}) \\
&= \sum_{i=1}^{n} (r_i - \hat{x}_i)^2 - \sum_{i=1}^{n} (r_i - x_i)^2 \\
&= \sum_{i=1}^{n} (r_i - x_i - \hat{x}_i + x_i)^2 - \sum_{i=1}^{n} (r_i - x_i)^2 \\
&= \sum_{i=1}^{n} (\hat{x}_i - x_i)^2 - 2\sum_{i=1}^{n} (\hat{x}_i - x_i)(r_i - x_i) \\
&= \sum_{i=1}^{n} (\hat{x}_i - x_i)^2 - 2\sum_{i=1}^{n} (\hat{x}_i - x_i)(v_i + b) \\
&\geq 2(\sigma + \beta) \sum_{i=1}^{n} |\hat{x}_i - x_i| - 2\sum_{i=1}^{n} |\hat{x}_i - x_i||v_i + b| \\
&= 2\sum_{i=1}^{n} |\hat{x}_i - x_i|(\sigma + \beta - |v_i + b|) \\
&> 0,
\end{aligned}$$

where the fourth equality follows from $r_i = x_i + v_i + b$, the first inequality follows from (3.14) and the last inequality from the fact that $|v_i + b| \leq |v_i| + |b| < \sigma + \beta$ for all i. Hence, if decoding is based on minimizing the Euclidean distance, the transmitted codeword is always chosen as the decoding result, leading to a WER equal to zero. □

MINIMUM MODIFIED PEARSON DISTANCE BASED DETECTION

Since modified Pearson distance based decoding features its immunity to offset mismatch, zero WER performance only requires a limited value of σ, shown in the following theorem.

Theorem 2. *If the noise and offset are restricted to the intervals* $(-\sigma,\sigma)$ *and* $(-\beta,\beta)$, *respectively, with*

$$\sigma < \min_{\mathbf{s},\mathbf{c}\in S,\mathbf{s}\neq\mathbf{c}} \left(\frac{\sum_{i=1}^{n} (s_i - \bar{\mathbf{s}} - c_i + \bar{\mathbf{c}})^2}{\frac{n-1}{n} 4 \sum_{i=1}^{n} |s_i - \bar{\mathbf{s}} - c_i + \bar{\mathbf{c}}|} \right), \tag{3.15}$$

then the Pearson decoder achieves a WER equal to zero.

Proof. Assume that $\mathbf{x} \in S$ is sent, and $\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}$ is received. Then, for all codewords

$\hat{\mathbf{x}} \neq \mathbf{x}$, it holds that

$$
\begin{aligned}
&\delta'_P(\mathbf{r},\hat{\mathbf{x}}) - \delta'_P(\mathbf{r},\mathbf{x}) \\
&= \sum_{i=1}^{n} (r_i - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2 - \sum_{i=1}^{n} (r_i - x_i + \bar{\mathbf{x}})^2 \\
&= \sum_{i=1}^{n} (r_i - \hat{x}_i + \bar{\hat{\mathbf{x}}} - \bar{\mathbf{r}})^2 - \sum_{i=1}^{n} (r_i - x_i + \bar{\mathbf{x}} - \bar{\mathbf{r}})^2 \\
&= \sum_{i=1}^{n} (x_i - \bar{\mathbf{x}} - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2 \\
&\qquad + \sum_{i=1}^{n} 2(x_i - \bar{\mathbf{x}} - \hat{x}_i + \bar{\hat{\mathbf{x}}})(r_i - x_i + \bar{\mathbf{x}} - \bar{\mathbf{r}}) \\
&= \sum_{i=1}^{n} (x_i - \bar{\mathbf{x}} - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2 + \sum_{i=1}^{n} 2(x_i - \bar{\mathbf{x}} - \hat{x}_i + \bar{\hat{\mathbf{x}}})(v_i - \bar{\mathbf{v}}) \\
&> \frac{n-1}{n} 4\sigma \sum_{i=1}^{n} \left| x_i - \bar{\mathbf{x}} - \hat{x}_i + \bar{\hat{\mathbf{x}}} \right| \\
&\qquad - \sum_{i=1}^{n} 2 \left| x_i - \bar{\mathbf{x}} - \hat{x}_i + \bar{\hat{\mathbf{x}}} \right| |v_i - \bar{\mathbf{v}}| \\
&= \sum_{i=1}^{n} \left| x_i - \bar{\mathbf{x}} - \hat{x}_i + \bar{\hat{\mathbf{x}}} \right| \left(\frac{n-1}{n} 4\sigma - 2|v_i - \bar{\mathbf{v}}| \right) \\
&\geq 0,
\end{aligned}
\tag{3.16}
$$

where the fourth equality follows by substituting $r_i = x_i + v_i + b$ and $\bar{\mathbf{r}} = \bar{\mathbf{x}} + \bar{\mathbf{v}} + b$, the first inequality from (3.15), and the last inequality from the fact that $|v_i - \bar{\mathbf{v}}| < \frac{n-1}{n} 2\sigma$ for all i. Hence, if decoding is based on minimizing the modified Pearson distance, the transmitted codeword is always chosen as the decoding result, leading to a WER equal to zero. □

MAXIMUM LIKELIHOOD DECODING

Finally, we show that zero WER for ML decoding is achieved if σ or $\sigma + \beta$ is sufficiently small.

Theorem 3. *If the noise and offset are restricted to the intervals* $(-\sigma, \sigma)$ *and* $(-\beta, \beta)$*, respectively, with*

$$
\sigma \leq \min_{\mathbf{s},\mathbf{c} \in S, \mathbf{s} \neq \mathbf{c}} \left(\frac{\max_{1 \leq i,j \leq n} \{(s_i - c_i) - (s_j - c_j)\}}{4} \right) \tag{3.17}
$$

or

$$
\sigma + \beta \leq \min_{\mathbf{s},\mathbf{c} \in S, \mathbf{s} \neq \mathbf{c}} \left(\frac{\max_{i=1,\ldots,n} (|s_i - c_i|)}{2} \right) \tag{3.18}
$$

then the ML decoder achieves a WER equal to zero.

Proof. Assume that $\mathbf{x} \in S$ is sent and $\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}$ is received. We will show that if (3.17) or (3.18) holds, then $\psi(\mathbf{r}-\hat{\mathbf{x}})=0$ for all codewords $\hat{\mathbf{x}}\neq\mathbf{x}$. First of all, note that

$$\begin{aligned}
&t_0(\mathbf{r},\hat{\mathbf{x}})-t_1(\mathbf{r},\hat{\mathbf{x}})\\
&=\min\left(\{r_i-\hat{x}_i+\sigma\,|\,i=1,\ldots,n\}\cup\{\beta\}\right)\\
&\qquad-\max\left(\{r_i-\hat{x}_i-\sigma\,|\,i=1,\ldots,n\}\cup\{-\beta\}\right)\\
&=\min\left(\{r_i-\hat{x}_i+\sigma\,|\,i=1,\ldots,n\}\cup\{\beta\}\right)\\
&\qquad+\min\left(\{-(r_i-\hat{x}_i)+\sigma\,|\,i=1,\ldots,n\}\cup\{\beta\}\right)\\
&=\min(\{2\beta\}\cup\{\min_{i=1,\ldots,n}\{-|r_i-\hat{x}_i|\}+\sigma+\beta\}\\
&\qquad\cup\{\min_{1\le i,j\le n}\{(r_i-\hat{x}_i)-(r_j-\hat{x}_j)\}+2\sigma\}).
\end{aligned}\tag{3.19}$$

Next, we will show that if (3.17) or (3.18) holds, this expression is negative whenever $\hat{\mathbf{x}}\neq\mathbf{x}$.

If (3.17) holds, then

$$\begin{aligned}
&\min_{1\le i,j\le n}\{(r_i-\hat{x}_i)-(r_j-\hat{x}_j)\}+2\sigma\\
&=\min_{1\le i,j\le n}\{(r_i-\hat{x}_i)-(r_j-\hat{x}_j)\}-2\sigma+4\sigma\\
&<\min_{1\le i,j\le n}\{(r_i-\hat{x}_i)-(r_j-\hat{x}_j)-(v_i-v_j)\}+4\sigma\\
&=\min_{1\le i,j\le n}\{[(r_i-\hat{x}_i)-(r_j-\hat{x}_j)]\\
&\qquad\qquad-[(r_i-x_i-b)-(r_j-x_j-b)]\}+4\sigma\\
&=\min_{1\le i,j\le n}\{(x_i-\hat{x}_i)-(x_j-\hat{x}_j)\}+4\sigma\\
&=-\max_{1\le i,j\le n}\{(\hat{x}_i-x_i)-(\hat{x}_j-x_j)\}+4\sigma\\
&\le 0.
\end{aligned}\tag{3.20}$$

where the first inequality follows from the fact that $v_i-v_j\le|v_i|+|v_j|<2\sigma$ and the second inequality from (3.17).

If (3.18) holds, then

$$\begin{aligned}
&\min_{i=1,\ldots,n}\{-|r_i-\hat{x}_i|\}+\sigma+\beta\\
&=\min_{i=1,\ldots,n}\{-|r_i-\hat{x}_i|\}-\sigma-\beta+2(\sigma+\beta)\\
&<\min_{i=1,\ldots,n}\{-|r_i-\hat{x}_i|-|v_i+b|\}+2(\sigma+\beta)\\
&=\min_{i=1,\ldots,n}\{-|r_i-\hat{x}_i|-|r_i-x_i|\}+2(\sigma+\beta)\\
&\le\min_{i=1,\ldots,n}\{-|x_i-\hat{x}_i|\}+2(\sigma+\beta)\\
&=-\max_{i=1,\ldots,n}\{|x_i-\hat{x}_i|\}+2(\sigma+\beta)\\
&\le 0,
\end{aligned}\tag{3.21}$$

where the first inequality follows from the fact that $|v_i+b|\le|v_i|+|b|<\sigma+\beta$ and the last inequality from (3.18).

Combining (3.19), (3.20), and (3.21) with (3.4) and (3.6), we find that indeed $\psi(\mathbf{r}-\hat{\mathbf{x}})=0$ for all codewords $\hat{\mathbf{x}}\neq\mathbf{x}$, while the probability density value of the received vector $\mathbf{r}$ given the transmitted codeword $\mathbf{x}$ is larger than 0, i.e., $\psi(\mathbf{r}-\mathbf{x})>0$. This implies that if decoding is based on maximizing (3.6), the transmitted codeword is always chosen as the decoding result, leading to a WER equal to zero. □

For the codebook S^*, the bound on $\sigma + \beta$ for an MED in (3.14) is 1/2, the bound on σ for an MMPD in (3.15) is 3/16, and the bounds on σ and $\sigma + \beta$ for an ML decoder in (3.17) and (3.18) are 1/4 and 1/2, respectively.

Considering Figures 3.1-3.3, results from Theorems 1-3 are confirmed. The zero WER of MMPD is indeed achieved if $\sigma < 3/16$. However, the shown results suggest that this may not be the best upper bound for the code under consideration. In addition, for $\sigma = 0.3$ and the example code S^*, Theorems 1 and 3 give that, for both MED and ML decoding, the WER is equal to zero if the offset is restricted to the interval $(-\beta, \beta)$ with $\beta \leq 0.5 - 0.3 = 0.2$. This confirms the results from Figures 3.4-3.5: for uniform offset, the zero WER is achieved if standard deviation $0.2/\sqrt{3} \approx 0.12$; for triangular offset, the zero WER is achieved if standard deviation $0.2/\sqrt{6} \approx 0.08$.

3.3. Maximum Likelihood Decoding for Channels with Gaussian Noise and Offset

This section focus on the situation where the noise and the offset mismatch are Gaussian distributed. In ML decoding, the receiver would like to pick from the codebook a codeword most like the received vector and output it. This gives the codeword that has the highest likelihood of being the one that was actually transmitted. In order to achieve ML decoding, one needs to maximize

$$\psi(\mathbf{r} - \hat{\mathbf{x}}) = \int_{-\infty}^{\infty} \chi(\mathbf{r} - \hat{\mathbf{x}} - b\mathbf{1})\zeta(b)\,db \tag{3.22}$$

over all candidate codewords. It is a convolution of the probability density functions of the noise and the offset. We will investigate the ML criterion for channel model (3.1) considering channel outputs with the length of one word and several words. In both cases, the noise and offset are assumed to be independent of each other. The offset is constant within a codeword length n, and its value may vary for the next block.

An ML decoding method of one-word length n is derived in [62], for which we will present the result but not give the proof. Later, simulations of two linear codes are demonstrated for this case. The noise samples v_i are normally distributed with mean 0 and standard deviation σ. Thus the probability density function $\chi(\mathbf{v})$ of the noise vector $\mathbf{v}$ follows the multi-dimensional Gaussian distribution as shown in Equation (2.15). We also assume that the channel offset b follows a Gaussian distribution with zero mean and variance β^2, that is,

$$\zeta(b) = \frac{1}{\beta\sqrt{2\pi}} e^{-\frac{b^2}{2\beta^2}}.$$

Let λ denote the ratio of the noise and offset variances, i.e., $\lambda = \sigma^2/\beta^2$. Theorem 3 in [62] shows that in case the noise and the offset are assumed to be normally distributed, ML decoding is achieved by minimizing

$$\frac{\lambda}{n+\lambda}\delta_E(\mathbf{r}, \hat{\mathbf{x}}) + \frac{n}{n+\lambda}\delta'_P(\mathbf{r}, \hat{\mathbf{x}}), \tag{3.23}$$

where $\delta_E(\mathbf{r}, \hat{\mathbf{x}})$ is the Euclidean distance defined in Equation (2.16) and $\delta'_P(\mathbf{r}, \hat{\mathbf{x}})$ is the modified Pearson distance defined in Equation (2.39). Next, we will use (3.23) as the decoding criterion in simulations of two codes.

The findings in Table 3.1 is the simulated WER results for the (3,2) parity check code, S^* of length 3 and size 4, in combination with different decoders and various choices for the noise and offset standard deviations.

Table 3.1: Simulated word error rate (WER) results of MED, MMPD, and ML decoding with (3,2) parity check code for channels with Gaussian noise and offset.

σ	β	(2.16) (Euclidean)	(2.39) (Pearson)	(3.23) ML_{Gauss}
0.2	1	0.318	0.031	0.030
0.2	0.2	0.026	0.031	0.009
0.3	0.2	0.064	0.130	0.054
0.3	0.01	0.025	0.130	0.025

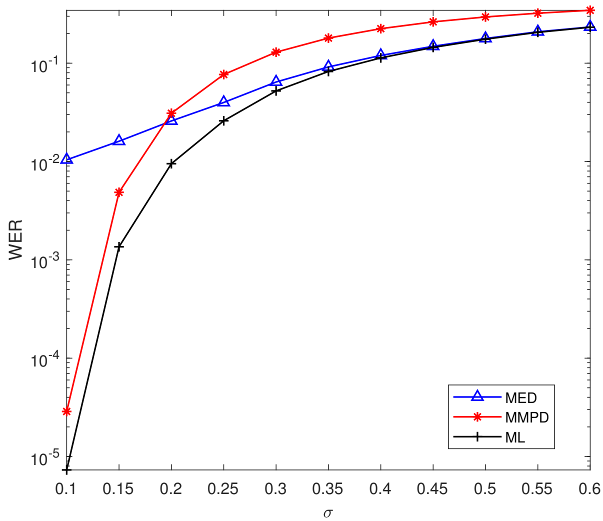


Figure 3.7: Simulated word error rate (WER) versus the standard deviation of the noise of MED, MMPD, and ML detectors for (3,2) parity check code when the standard deviation of the offset $\beta = 0.2$.

Furthermore, performances of these detectors as a function of standard deviations of the noise and offset are shown in the Figures 3.7 and 3.8. In Figure 3.7, the standard deviation of the noise σ is set from 0.1 to 0.6, where the standard deviation of the offset β is fixed at 0.2. In Figure 3.8, the WER results with the offset standard deviation β from 0.1 to 0.6 has been pictured, in which the standard deviation of noise σ is fixed at 0.2.

Note that in the case neither the noise nor the offset is strongly dominating the other, the ML decoder is clearly outperforming both MED and MMPD. The performance of MMPD is unchangeable with β as we expected. What's more, when the offset standard

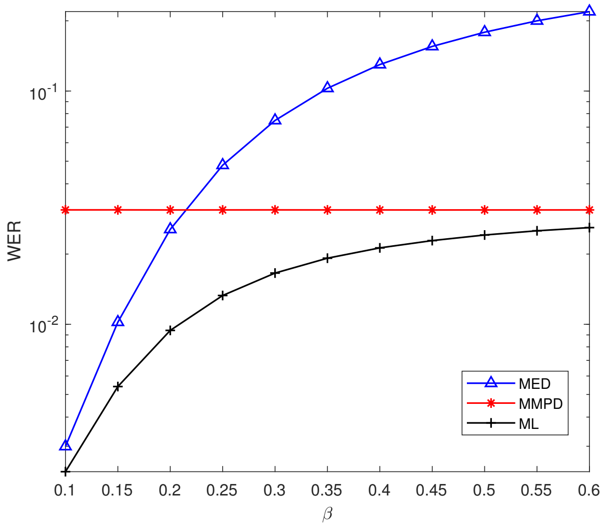


Figure 3.8: Simulated word error rate (WER) versus the standard deviation of the offset of MED, MMPD, and ML detectors for (3,2) parity check code when the standard deviation of the noise $\sigma = 0.2$.

deviation equals the noise standard deviation, i.e., $\sigma = \beta$, the weight of MED and MMPD are $1/(n+1)$ and $n/(n+1)$ respectively in the ML decoder. The criterion tends more and more towards the offset-resistant Pearson distance when n is increasing. The reason for this is that the standard deviation per dimension is $\sigma/\sqrt{n}$ for the noise, while it is constant at β for the offset.

The findings in Table 3.2 is the simulated WER results for the modified (7,4,3) Hamming code of size 14, presented in Section 2.1.2, in combination with different detectors and various choices for the noise and offset standard deviations.

Table 3.2: Simulated word error rate (WER) results of MED, MMPD, and ML decoding with modified (7,4,3) Hamming code for channels with Gaussian noise and offset.

σ	β	(2.16) (Euclidean)	(2.39) (Pearson)	(3.23) ML_{Gauss}
0.3	1	0.187	0.014	0.014
0.3	0.2	0.019	0.014	0.013
0.4	0.2	0.098	0.093	0.090
0.4	0.01	0.085	0.093	0.085

The performances of MED, MMPD, and ML detector with different standard deviations of noise are shown in the Figures 3.9 and 3.10. Figure 3.9 pictures the WER versus the standard deviation of the noise σ from 0.2 to 0.6, where the standard deviation of the

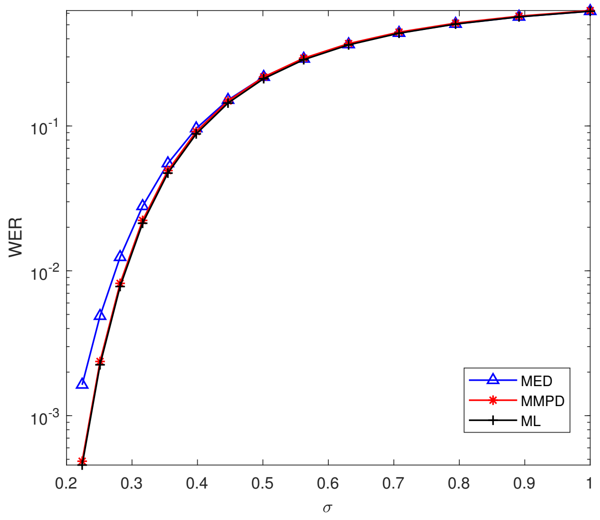


Figure 3.9: Simulated word error rate (WER) versus the standard deviation of the noise of MED, MMPD, and ML detectors for modified (7,4,3) Hamming code when the standard deviation of the offset $\beta = 0.2$.

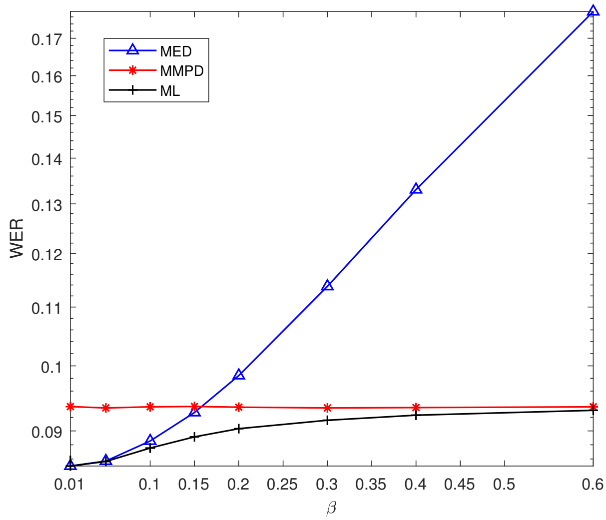


Figure 3.10: Simulated word error rate (WER) versus the standard deviation of the offset of MED, MMPD, and ML detectors for modified (7,4,3) Hamming code when the standard deviation of the noise $\sigma = 0.4$.

offset β is fixed at 0.2. As we can see, performances of MMPD and ML are highly close for a small value of the standard deviation of the noise, while MED is worse than both of them. However, both MED and MMPD show similar performance as ML for large values of the standard deviation of the noise. It is clear that with the increase of standard deviation of the noise, the MED performs only slightly better than MMPD.

In addition, the performances of MED, MMPD, and ML detector with different standard deviations of offset are shown in Figure 3.10. Here, the standard deviation of the noise is set at 0.4 because, in this case, we can get the enlarged view of both MED and MMPD dominant areas. The performance of MMPD is unchangeable with β as we expected, while WER of MED increases with β and ML detector outperforms both MED and MMPD.

Now, we explore a criterion to achieve ML decoding for several successive channel outputs together. This theoretical work serves as the basis for the design of concatenated coding schemes presented in Section 3.4. Specifically, this model assumes that a received sequence, $\mathbf{r}$, of length $N = l \times n$ spans l blocks of length n for a positive integer $l > 1$.

Based on the channel model, we define a total distortion of length N as

$$\mathbf{d} = \mathbf{v} + \mathbf{b},$$

where $\mathbf{v}$ is the noise vector of length N with i.i.d. Gaussian noise samples, and

$$\mathbf{b} = (b_1\mathbf{1}_n, b_2\mathbf{1}_n, \dots, b_l\mathbf{1}_n)$$

is the offset vector with i.i.d. Gaussian offset samples, $b_i \sim \mathcal{N}(0, \beta^2)$, $i = 1, \dots, l$, which are constant within each block of length n. Denote the probability density function of $\mathbf{b}$ as $\gamma(\mathbf{b})$. Then the probability density function of $\mathbf{d}$ is given by

$$\psi(\mathbf{d}) = \int_{-\infty}^{\infty} \phi(\mathbf{d} - \mathbf{b})\gamma(\mathbf{b})\,\mathrm{d}\mathbf{b}. \tag{3.24}$$

An ML decoder will choose a codeword maximizing the probability density function (3.24), that is,

$$\arg\max_{\hat{\mathbf{x}} \in S} \psi(\mathbf{d}).$$

In the situation of zero-mean Gaussian noise and offset, $\mathbf{d}$ has a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and covariance matrix Σ. Since the noise is independent of the offset, Σ is an $N \times N$ matrix partitioned by

$$\Sigma = \begin{bmatrix} \mathbf{A}_n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & & \vdots \\ \vdots & & \mathbf{A}_n & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_n \end{bmatrix}, \tag{3.25}$$

where $\mathbf{A}_n$ is an $n \times n$ matrix with all entries on the main diagonal equal to $\sigma^2 + \beta^2$ and all other entries equal to β^2. Thus, the probability density function of $\mathbf{d}$ is

$$\psi(\mathbf{d}) = \frac{\exp(-\mathbf{d}\Sigma^{-1}\mathbf{d}^{\mathrm{T}}/2)}{\sqrt{(2\pi)^N \det \Sigma}}, \tag{3.26}$$

where Σ^{-1} is the inverse matrix of Σ and $\det\Sigma$ is the determinant of Σ.

The determinant and the inverse matrix of $\mathbf{A}_n$ are

$$\det\mathbf{A}_n = \sigma^{2(n-1)}(\sigma^2 + n\beta^2) \tag{3.27}$$

and

$$\mathbf{A}_n^{-1} = \frac{1}{\sigma^2}\left(\mathbf{I}_n - \frac{\beta^2}{\sigma^2 + n\beta^2}\mathbf{1}_{n\times n}\right), \tag{3.28}$$

where $\mathbf{I}_n$ is an $n \times n$ identity matrix and $\mathbf{1}_{n\times n}$ is an $n \times n$ all-one matrix. $\mathbf{A}_n$ is a non-singular matrix since $\det\mathbf{A}_n \neq 0$. Thus, we have that

$$\det\Sigma = \det\mathbf{A}_n^l \tag{3.29}$$

and

$$\Sigma^{-1} = \begin{bmatrix} \mathbf{A}_n^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & & \vdots \\ \vdots & & \mathbf{A}_n^{-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_n^{-1} \end{bmatrix}. \tag{3.30}$$

Since the logarithm function is strictly increasing on the positive real numbers and ψ is a positive function, ML decoding can also be achieved by maximizing the logarithm of (3.26), i.e.,

$$\ln\psi(\mathbf{d}) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln(\det\Sigma) - \frac{1}{2}\mathbf{d}\Sigma^{-1}\mathbf{d}^{\mathrm{T}},$$

rather than maximizing (3.26) itself. By inverting the sign and ignoring irrelevant terms, we find that maximizing $\ln\psi(\mathbf{d})$ is equivalent to minimizing

$$\ln\det\mathbf{A}_n^l + \sum_{i=1}^{N}\sum_{j=1}^{N} d_i \begin{bmatrix} \mathbf{A}_n^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & & \vdots \\ \vdots & & \mathbf{A}_n^{-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_n^{-1} \end{bmatrix}_{ij} d_j, \tag{3.31}$$

where d_i is the i-th term in the vector $\mathbf{d}$. By applying (3.27), the first term of (3.31) is

$$\ln\det\mathbf{A}_n^l = l\ln(\sigma^{2(n-1)}(\sigma^2 + n\beta^2)). \tag{3.32}$$

The term (3.32) is irrelevant to the optimization process (independent of $\hat{\mathbf{x}}$). For any vector $\mathbf{u}$ of length N, denote its segments of length n as $\mathbf{u}^{(k)}$, $k = 1,\ldots,l$. The second term of (3.31) is

$$\sum_{i=1}^{N}\sum_{j=1}^{N} d_i \begin{bmatrix} \mathbf{A}_n^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & & \vdots \\ \vdots & & \mathbf{A}_n^{-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_n^{-1} \end{bmatrix}_{ij} d_j$$

$$= \frac{1}{\sigma^2}\left[\sum_{i=1}^{n}(d_i)^2 - \frac{\beta^2}{\sigma^2+n\beta^2}n^2(\overline{\mathbf{r}^{(1)}}-\overline{\hat{\mathbf{x}}^{(1)}})^2\right]$$

$$+ \frac{1}{\sigma^2}\left[\sum_{i=n+1}^{2n}(d_i)^2 - \frac{\beta^2}{\sigma^2+n\beta^2}n^2(\overline{\mathbf{r}^{(2)}}-\overline{\hat{\mathbf{x}}^{(2)}})^2\right]$$

$$\cdots$$

$$+ \frac{1}{\sigma^2}\left[\sum_{i=(l-1)n+1}^{N}(d_i)^2 - \frac{\beta^2}{\sigma^2+n\beta^2}n^2(\overline{\mathbf{r}^{(l)}}-\overline{\hat{\mathbf{x}}^{(l)}})^2\right],$$

where the average value of the segment $\mathbf{u}^{(k)}$ is defined by

$$\overline{\mathbf{u}^{(k)}} = \frac{1}{n}\sum_{i=(k-1)n+1}^{kn} u_i, \tag{3.33}$$

for $k = 1,\ldots,l$. Substituting

$$n(\overline{\mathbf{r}^{(k)}}-\overline{\hat{\mathbf{x}}^{(k)}})^2 = \delta_E(\mathbf{r}^{(k)},\hat{\mathbf{x}}^{(k)}) - \delta'_P(\mathbf{r}^{(k)},\hat{\mathbf{x}}^{(k)}),$$

which follows from definitions of Euclidean distance (2.16) and modified Pearson distance (2.39), gives that the ML decoding criterion is to minimize

$$\sum_{k=1}^{l}\left[\frac{\lambda}{n+\lambda}\delta_E(\mathbf{r}^{(k)},\hat{\mathbf{x}}^{(k)}) + \frac{n}{n+\lambda}\delta'_P(\mathbf{r}^{(k)},\hat{\mathbf{x}}^{(k)})\right], \tag{3.34}$$

over all candidate codewords $\hat{\mathbf{x}} \in S$, where $\lambda = \sigma^2/\beta^2$. ML decoding criterion is intuitively separated into weighted combinations of Euclidean distance and modified Pearson distance for each segment. The ML criterion for one word length, i.e., Equation (3.23), can be obtained by letting $l = 1$ in (3.34).

3.4. DECODING OF CONCATENATED CODES FOR CHANNELS WITH GAUSSIAN NOISE AND OFFSET

This section proposes a concatenated coding scheme for channels with Gaussian distributed noise and offset mismatch. A novel decoding algorithm for the concatenated codes is proposed to exploit its error correction potential better. The simulation results show that the proposed scheme achieves considerable coding gain compared with the non-concatenated codes with an even higher code rate over noisy channels with offset mismatch.

3.4.1. RS-COSET CODES

For the noisy channels with unknown offset, we design a concatenated code. A Reed-Solomon (RS) code [69] and a certain coset of a binary block code proposed in [66] are used as outer and inner codes, respectively. The two codes are chosen according to a rule that the inner code is of a short length. In such a way, an exhaustive search of MMPD is for relatively small codes, and the outer code's decoding algorithm can make up for errors decoding the inner code. We call coset codes and RS codes together as RS-Coset codes.

The encoder block diagram of RS-Coset codes is shown by Figure 3.11. RS-Coset code is composed of $S_{\mathbf{a}}$ inner coset codes defined over $GF(2)$, or alphabet $[2] = \{0,1\}$, and RS outer code defined over $GF(2^m)$, where m is a positive integer. RS codes are non-binary codes, which may be defined over any finite field $GF(q)$. However, since most modern applications use binary data, RS codes over $GF(2^m)$ are of great interest. A block interleaver between the inner and outer encoders is not considered, which will be investigated with iterative decoding algorithms in further research.

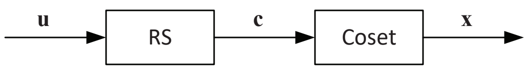


Figure 3.11: Block diagram of the RS-Coset code.

For (n_2, k_2, d_2) RS code n_2, k_2 and d_2 represent the length of a codeword, number of information symbols, and the minimum Hamming distance, respectively. The message vector of an (n_2, k_2, d_2) RS code can be written as

$$\mathbf{u} = (u_1, u_2, \ldots, u_{k_2}) \in \mathbb{F}_q^{k_2}, \tag{3.35}$$

where $n_2 = 2^m - 1$, or a polynomial

$$u(x) = u_1 + u_2 x + \ldots + u_{k_2} x^{k_2 - 1}.$$

RS codes of length n_2 have generator polynomials $g(x) = (x-\alpha)(x-\alpha^2)\ldots(x-\alpha^{n_2-k_2})$ or a generator matrix

$$\mathbf{G}_r = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{(n_2-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k_2-1} & \cdots & \alpha^{(k_2-1)(n_2-1)} \end{pmatrix},$$

where α is a primitive element of $\mathbb{F}_q$. The RS codeword is generated by

$$\mathbf{c} = \mathbf{u} \cdot \mathbf{G}_r = (c_1, c_2, \ldots, c_{n_2}) \in \mathbb{F}_q^{n_2}. \tag{3.36}$$

Because all codeword polynomials $c(x)$ have $g(x)$ as a factor, every $c(x)$ has all the roots of $g(x)$ as its roots and has a parity check matrix

$$\mathbf{H}_r = \begin{pmatrix} 1 & \alpha & \cdots & \alpha^{n_2-1} \\ 1 & \alpha^2 & \cdots & \alpha^{2(n_2-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n_2-k_2} & \cdots & \alpha^{(n_2-k_2)(n_2-1)} \end{pmatrix}$$

such that $\mathbf{c}\mathbf{H}_r^T = 0$. After the RS codeword has been generated, it will be converted into a binary coded bit sequence to form the input to the inner encoder, i.e.,

$$\mathbf{c}' = c'_1, c'_2, \ldots, c'_{n_2 m}. \tag{3.37}$$

Binary block codes proposed in [66] work well with the modified Pearson distance based decoding criterion (2.39), which guarantees immunity to channel offset mismatch. Note that for any binary linear block code S containing the all-one vector, the minimum δ'_P distance is zero since $\delta'_P(\mathbf{0}, \mathbf{1}) = 0$.

It is shown that the minimum δ'_P distance significantly increases by using a well-chosen coset of S rather than S itself [66]. Assume that the generator matrix of a block (n_1, k_1, d_1) code S is $\mathbf{G}_s$. We first cluster the binary coded bit sequence $\mathbf{c}'$ into groups each of k_1 bits, and then encode each group of k_1 bits with the generator matrix $\mathbf{G}_s$. This results in codewords each of n_1 bits. The inner binary block codes proposed in [66] – the coset of codes $S_\mathbf{a}$ – is obtained by

$$S_\mathbf{a} = \{\mathbf{v} + \mathbf{a} | \mathbf{v} \in S\}, \tag{3.38}$$

where $\mathbf{a}$ is any binary vector of length n_1 with weight $\lfloor \frac{d_1}{2} \rfloor$, $\lceil \frac{d_1}{2} \rceil$, $n_1 - \lfloor \frac{d_1}{2} \rfloor$, or $n_1 - \lceil \frac{d_1}{2} \rceil$. Codeword length and dimension of $S_\mathbf{a}$ is still n_1 and k_1, respectively. Information can also be uniquely mapped to its codewords by using the generator matrix $\mathbf{G}_s$ followed by a simple shift operation.

With the above mentioned input, the inner codeword is

$$\mathbf{x} = (x_1, x_2, \ldots, x_{n_1}) \in \mathbb{F}_2^{n_1}. \tag{3.39}$$

Length, dimension, and code rate of RS-Coset code are the products of the corresponding parameters of the inner and outer code. Note that variable rates of the concatenated codes can be realized by puncturing the output of the inner code.

3.4.2. Decoding Algorithm for Concatenated Coding Scheme

The block diagram of the proposed decoding algorithm is shown by Figure 3.12. The MMPD detection is used to decode the inner coset code $S_\mathbf{a}$, which guarantees immunity to channel offset mismatch. Its output will be given to a two-stage hybrid decoding algorithm for the outer RS code. The first stage of decoding carries out an algebraic HDD algorithm, such as the Berlekamp-Massey algorithm (BMA), to the outer RS code. If the HDD declares a successful decoding, the algorithm outputs the decoded codeword and terminates the decoding process. Otherwise, the decoding is continued with the second stage of decoding, using the reduced test-pattern Chase algorithm. The reliability information used in the Chase algorithm is obtained from a subtraction between the received vector and an estimated offset, where a dynamic threshold estimation [57] is explored.

Decoding Coset Code Using Modified Pearson Distance Decoder

Consider transmitting a codeword $\mathbf{x}$ over a Gaussian noisy channel with offset mismatch (3.1), i.e., $\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}$. When the receiver uses a fixed threshold detection, based on a binary symmetric channel model, the elements of $\mathbf{r}$ are quantized to 0's and 1's. Its reliability faces severe degradation as the offset mismatch introduces asymmetric or

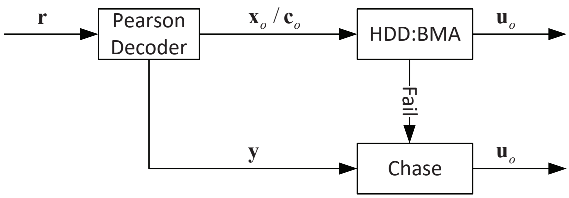


Figure 3.12: Block diagram of the decoding algorithm.

3

unidirectional errors. Better results are expected when the real received signal is passed directly to a modified Pearson distance based decoder (2.39).

A Pearson decoder chooses a codeword minimizing the modified Pearson distance between the received vector and candidate codewords from the coset codes, i.e.,

$$\mathbf{x}_o = \arg\min_{\hat{\mathbf{x}} \in S_{\mathbf{a}}} \delta'_P(\mathbf{r}, \hat{\mathbf{x}}).$$

Subsequently we find an estimate of the offset [57], denoted by $\hat{b}$, by averaging (3.1), or

$$\hat{b} = \frac{1}{n_1} \sum_{i=1}^{n_1} (r_i - x_{o,i}) = \bar{\mathbf{r}} - \frac{\omega_o}{n_1}, \tag{3.40}$$

where ω_o is the weight of $\mathbf{x}_o$.

In the case of passing the analog received signal, the inner decoder will generate the information reliability $\mathbf{y}$ from $\mathbf{r}$ based on a posteriori log-likelihood-ratio (LLR) of each bit input. The received symbol r_i has a Gaussian probability density function which is offset by a parameter, b. Denote the LLR of the bit x_i by $L(x_i)$. Assuming independent and uniformly distributed channel inputs, we thus have

$$\begin{aligned} L(x_i) &= \ln \frac{\Pr\left(y_i | x_i = 1\right)}{\Pr\left(y_i | x_i = 0\right)} + \ln \frac{\Pr(x_i = 1)}{\Pr(x_i = 0)} \\ &= \ln \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r_i - 1 - b)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r_i - 0 - b)^2}{2\sigma^2}}} \\ &= \frac{(r_i - b)^2}{2\sigma^2} - \frac{(r_i - 1 - b)^2}{2\sigma^2} \\ &= \frac{1}{\sigma^2}\left(r_i - b - \frac{1}{2}\right). \end{aligned}$$

Considering a stationary channel, we can normalize the LLR with respect to a constant $1/\sigma^2$.

Thus, the reliability of the component r_i is defined using the absolute value of re-scaled retrieved vector by subtracting the estimated offset, $\hat{b}$, so that

$$y_i = \left| r_i - \hat{b} - \frac{1}{2} \right| = \left| r_i - \left(\bar{\mathbf{r}} - \frac{\omega_o}{n_1} \right) - \frac{1}{2} \right|, \tag{3.41}$$

for $i = 1, \ldots, n_1$.

HYBRID DECODING OF RS CODES

There are several existing algorithms for decoding RS codes that provide better performances than the HDD. However, the large alphabet size of the RS code precludes the use of ML decoders due to the very high computational complexity. The Chase algorithm is a soft input decoding algorithm for linear block code with lower computational complexity than the ML algorithm. Furthermore, Chase decoding can be implemented efficiently in a parallel manner, and the decoder has no iterations. In order to further reduce the decoding latency while still retaining a similar performance gain, a simple but effective two-stage hybrid decoding algorithm [70] is adopted here for decoding RS codes.

3

Stage 1 : Berlekamp-Massey Algorithm (BMA) Decoding of the outer code is realized first by BMA [71, 72]. By reading out each output information bit of the inner decoder and converting binary sequences into the 2^m-ary expression

$$\mathbf{x}_o \mapsto \mathbf{c}_o,$$

we can obtain a hard decision for each RS coded symbol. First-stage of decoding carries out the BMA to $\mathbf{c}_o$. For a (n_2, k_2, d_2) RS code, BMA can correct errors of length up to

$$t_0 = \lfloor \frac{n_2 - k_2}{2} \rfloor.$$

If more than t_0 errors occur, one of two cases can happen. Either the BMA decoder declares decoding failure or finds an incorrect codeword different from the transmitted one. We say the BMA decoding successful whenever BMA finds a codeword within the Hamming sphere of radius t_0 around the received vector. If the BMA declares a successful decoding, it outputs the decoded message $\mathbf{u}_o$, and terminates the decoding process. Otherwise, the decoding is continued with the second stage of decoding, using the Chase decoder optimized for the noisy channel with unknown offset.

Stage 2 : Chase Algorithm We implement Chase algorithm [58] to decode RS codes in the second stage. Note that it is unnecessary to transfer LLR at bit level of the coset code to LLR at byte level of RS since the Chase algorithm is based on binary data sequence. During Chase decoding, the hard-decision decoding operation that is applied to each test pattern is computationally expensive. A test pattern reduced Chase algorithm [73], which eliminates the unnecessary hard-decision decoding operations, is adopted. The Chase algorithm in the second stage can be described as follows:

1. Convert $\mathbf{c}_o$ into a binary sequence $\mathbf{c}'_o$ and determine the positions of p least reliable binary bits of $\mathbf{c}'_o$.
2. Form test patterns $\mathbf{T}^P$ for $P = 1, \dots, 2^p$ by forming all possible binary combinations over these p least reliable positions.
3. Take a new test pattern from the set of test patterns $\mathbf{T}^P$ obtained from Step 2. Check whether a stored set of candidate patterns $\mathbf{E}$ is empty. If yes, go to Step 4. Otherwise, compute Hamming distances between the current test pattern and each

stored candidate pattern in **E**. If all the Hamming distances are greater than t, where t is the number of errors that can be corrected by the code, go to Step 4. Otherwise, go to Step 3.

4. Decode the 2^m-ary expression of the current test pattern using an algebraic decoder and store the successfully decoded codeword into **E**. Go back to Step 3 until all test patterns in $\mathbf{T}^P$ have been processed.

5. Estimate the most likely transmitted sequence from the reduced set of candidate patterns **E**.

We now explain more details of the above algorithm. For the first step of Chase decoding, the p least reliable bits in the binary outputs of the inner decoder, $\mathbf{c}'_o$, are identified. The reliability information of the channel detected bits needs to be obtained in order to determine p least reliable positions. We assume that the data bits are transmitted with an equal probability, and the p least reliable bits are obtained based on the information reliability $\mathbf{y}$ given by (3.41). The smaller the magnitude of the LLR, the less reliable the inner decoder output bit, and vice versa.

Furthermore, all 2^p binary permutations that have '1's in these p positions and '0's elsewhere, are modulo-2 added to $\mathbf{c}'_o$ in order to form the test patterns $\mathbf{T}^P$. Note that if the Hamming distance between the current test pattern and a candidate pattern is not greater than t, decoding the current test pattern will lead to the candidate pattern. Hence, the corresponding HDD operation is redundant and can be skipped.

After decoding the test patterns, in order to select the most likely candidate pattern in **E**, we adopt the ML decision rule of (3.34) derived for the noisy channel with unknown offset. From the previous subsection, we have that the offset varies each block of length k_1 within the binary version of RS codewords of length $n_2 m$. Denote $l' = n_2 m / k_1$. Therefore, the ML metric for the Chase decoder for the noisy channels with unknown offset is given by

$$\sum_{k=1}^{l'} \left[\frac{\lambda}{k_1+\lambda} \delta_E(\mathbf{r}^{(k)}, \hat{\mathbf{x}}^{(k)}) + \frac{k_1}{k_1+\lambda} \delta'_P(\mathbf{r}^{(k)}, \hat{\mathbf{x}}^{(k)}) \right], \tag{3.42}$$

and the codeword with the minimum (3.42) is chosen as the decoded codeword.

3.4.3. PERFORMANCE EVALUATION

We evaluate an example concatenated code, which implements (7,3,5) RS code as the outer code and the coset of binary (6,3,3) code as the inner code. Binary (6,3,3) code is a shortened version of (7,4,3) Hamming code and a coset vector $\mathbf{a} = (1,0,0,0,0,0)$ is considered. These 2^3 codewords are

100000	111011	010110
101110	000011	011000
110101	001101	

The outer code is the (7,3,5) RS code, which is over $GF(2^3) = GF(8)$. The resulting concatenated code is thus a binary (42,9,15) code. Hence, the code rate is $9/42 = 0.21$.

In the encoding process, 9 information bits are transformed into 3 symbols from $GF(8)$, which are encoded using the RS code. These 9 information bits are encoded to 21 bits grouped in a 7-symbols RS codeword. The 7 resulting symbols are each considered as binary 3-tuples. Each 3-bits of the coded binary RS sequences is encoded using the shortened Hamming code and shifted with the coset vector $\mathbf{a}$. The final result is a $(3+4)\times(3+3) = 42$ bit string.

The received string of 42 bits is first parsed into 7 strings of 6 bits, which are decoded according to a criterion of minimizing modified Pearson distance. The resulting 7 binary 3-tuples are considered as symbols from $GF(8)$, which are collected into a vector to be decoded by the hybrid decoding of RS codes. This leads to three symbols from $GF(8)$, which can be transformed into 3 bits each, so 9 bits in total.

3

In Figure 3.13, we show bit error rate (BER) performances of RS-Coset codes versus SNR (dB) over channels with Gaussian noise and offset mismatch. We also assume that standard deviations of the offset are $\beta = 0.3$ (red curves) and $\beta = 0.5$ (blue curves). Let us first compare the performance of the inner soft-decision decoders based on two criteria, specifically MED and MMPD. The MED detection has worse performance when the offset is larger, while the scheme using MMPD remains the same for any offset as we expected. We conclude that the MMPD detection is immune to the offset mismatch and achieves considerable performance improvements, particularly when the offset is large compared to the noise.

We also observe that with the Hamming coset code as inner code instead of Hamming code itself, the simulated results of the concatenated scheme have been improved. The MMPD of the shortened Hamming code $(6,3,3)$ and its coset code is the same. However, the average number of neighbors with the minimum distance has decreased after introducing the coset of code. In addition, the introduction of the coset increases the MED detection's resistance to the offset mismatch. Thus, the performances of both MED and MMPD detection have improved by using the coset of block codes as proved in [66].

BER performance of the variable number of least reliable bits in Chase decoding is depicted in Figure 3.14. For the Chase algorithm, the number of test positions is set to be from 2 to 6. The results show that more performance enhancement can be achieved if we select more test positions in the Chase decoding algorithm. For example, the performance with 2 test positions has been increased by 0.5 dB compared to when 6 test positions are used. However, better performance comes at the price of higher complexity.

Figure 3.15 compares the proposed concatenated scheme with the Hamming coset code and uncoded scheme. Simulation is carried out over channels with $\beta = 0.5$. The number of the test positions is 2 for all the concatenated schemes in the simulations. The Hamming coset code is decoded using the MMPD scheme. By referring to the channel raw BER illustrated by Curve Uncoded, the proposed concatenated scheme achieves a significant gain in BER over a wide range of SNR. Furthermore, compared with the Hamming coset code, it can be observed that at BER = 10^{-4}, the gain of the concatenated scheme corresponds to the decrease of the system required SNR from around 9 dB (corresponding to the HM(6,3)) to 5 dB (corresponding to the HM(6,3)+RS(7,3)). Thus, we achieve more than 4 dB SNR improvement of achieving a BER = 10^{-4} with the proposed RS-Coset codes.

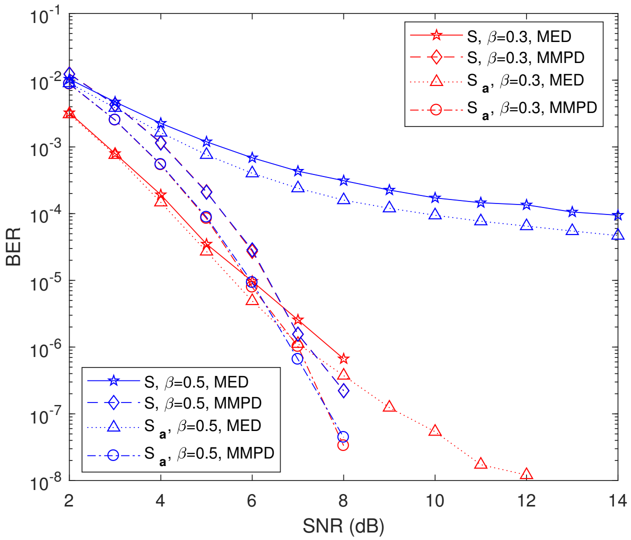


Figure 3.13: BER performances of (7,3,5) RS code concatenated with shortened (6,3,3) Hamming code or its coset, with the inner soft decision decoder based on two criteria – MED and MMPD – over channels with Gaussian noise and offset, where standard deviations of the offset are $\beta = 0.5$ (blue) and 0.3 (red).

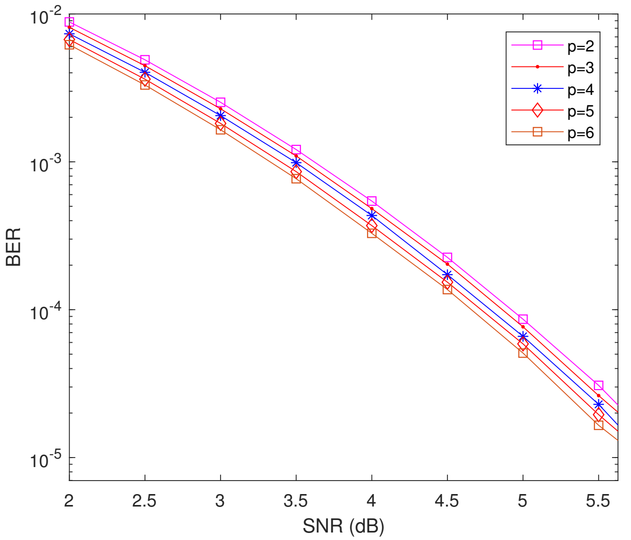


Figure 3.14: BER performances of RS-Coset codes with the different number of test positions under noisy channels with unknown offset mismatch.

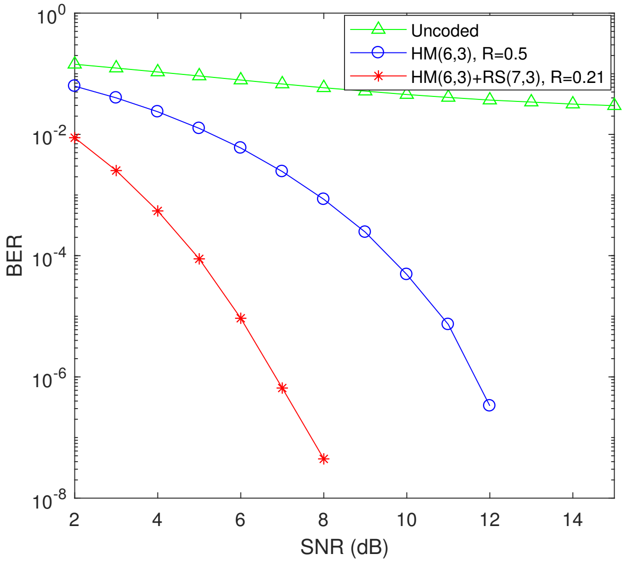


Figure 3.15: BER performances of (7,3,5) RS code concatenated with coset shortened Hamming (6,3,3), Hamming (6,3,3) coset code, and uncoded case over channels with Gaussian noise and offset, where the standard deviation of the offset is $\beta = 0.5$.

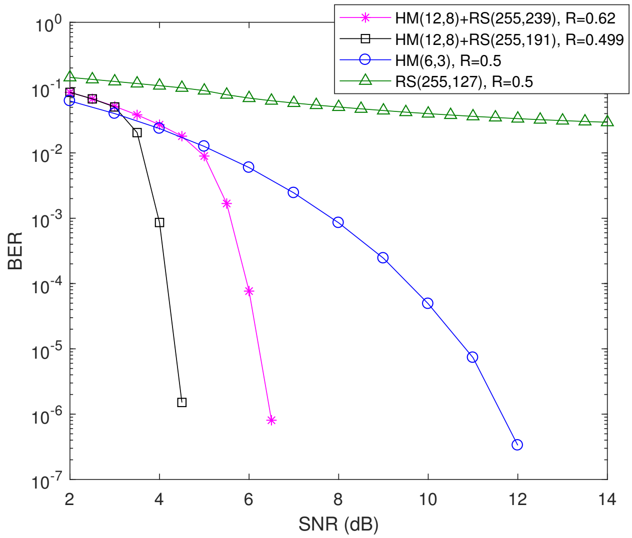


Figure 3.16: BER performances of different coding schemes over channels with Gaussian noise and offset, where the standard deviation of the offset is $\beta = 0.5$.

Our example code has a code rate of 0.21, which is lower than the non-concatenated scheme. We have two other concatenation schemes with higher code rates: first, coset shortened Hamming (12,8) concatenated with (255,191) RS with a code rate of 0.499; and second, coset shortened Hamming (12,8) concatenated with (255,239) RS with a code rate of 0.62. Observe from Figure 3.16, the concatenation codes with a higher rate are shown to have worse BER performance. However, the concatenated scheme with a code rate of 0.62 still performs better than the Hamming coset code, which has a code rate of 0.5. Moreover, (255,127) RS code is simulated under the same channel condition, whose code rate is 0.5. We can see that the offset mismatch seriously distorts its performance without the help of the inner MMPD decoder. The simulation results show the considerable coding gain compared with the non-concatenated codes with an even higher code rate over noisy channels with offset mismatch. Thus, we conclude that the RS-Coset codes with the proposed decoding scheme can achieve significant coding gain and maintain immunity to offset mismatch.

3.5. CONCLUSION

Our attention is on the noisy channels with the offset mismatch in the all-one direction in this chapter. We have derived an ML criterion for channels with bounded noise and offset mismatch. In particular, it has been shown that the WER for MED, MMPD, and ML decoders is equal to zero if the noise and offset ranges satisfy certain conditions. Further, a combination of weighted MED and MMPD is shown to be the ML decoding for channels with Gaussian noise and offset. We have proposed a decoding algorithm for concatenated codes in channels with Gaussian noise and offset mismatch. Simulation results demonstrate that the proposed decoding scheme can achieve significant coding gain and maintain immunity to offset mismatch.

4

Noisy Channels with Signal Dependent Offset Mismatch

In the previous chapter, maximum likelihood (ML) decision criteria have already been developed for noisy channels suffering from signal independent offset. In this chapter, such ML criteria are considered for the case of signals suffering from noise and signal dependent offset. The signal dependency of the offset signifies that it may differ for distinct signal levels. For instance, the offset experienced by the zeroes in a transmitted codeword is not necessarily the same as the offset for the ones.

An ML decision criterion is derived, assuming uniform distributions for both the noise and the signal dependent offset. In particular, for the proposed ML decoder, bounds are determined on the standard deviations of the noise and the offset, leading to a word error rate equal to zero. Simulation results are presented, confirming the findings.

Secondly, we consider Gaussian distributions for both the noise and the signal dependent offset. Besides the ML criterion itself, an option to reduce the complexity is also considered. Further, a brief performance analysis is provided, validating the superiority of the newly developed ML decoder over classical decoders based on the Euclidean or Pearson distances.

The material in this chapter has appeared in

- Section 4.2

  R. Bu, J. H. Weber and K. A. S. Immink, "Maximum Likelihood Decoding for Channels with Uniform Noise and Signal Dependent Offset", *Proc. of IEEE International Symposium on Information Theory*, August, 2020.

- Section 4.3

  R. Bu, J. H. Weber and K. A. S. Immink, "Maximum Likelihood Decoding for Channels With Gaussian Noise and Signal Dependent Offset", *IEEE Transactions on Communications*, vol. 69, pp. 85-93, Sep., 2020.

4.1. INTRODUCTION

In the last chapter, we assume that the offset is independent of the signal levels. Here, we look at a different model of the offset mismatch, where it is assumed to be a signal dependent parameter. For instance, b_0 is the offset for the '0' signal level, and b_1 is the offset for the '1' signal level in the binary case. The signal dependent offset model is appropriate in many scenarios. For example, the binary input user data is stored as the two resistance states of a spin-torque transfer magnetic random access memory (STT-MRAM) cell [70]. A signal dependent offset model is reasonable when process variation causes an asymmetric distribution of both the low and high resistance states. The model is also appropriate for modeling the retention of multilevel-cell phase-change memory, which is adversely affected by resistance states dependent drift and noise [74]. Moreover, degradation of the data reliability can be modeled as a signal dependent offset model, for the situation that with the increase of temperature, the low signal level hardly changes, while the high signal level decreases, leading to a drift of the high signal level to the low signal level [75]. Cai et al. [76] have proposed and analyzed a k-means clustering technique as a detection method for channels where the signal dependent offsets are assumed to be uncorrelated stochastic variables with a uniform probability distribution.

We consider transmitting a codeword $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ from a codebook $S \subseteq [q]^n$, where $[q]$ is a q-ary alphabet. With every signal level $j \in [q]$ we associate an offset value b_j. The transmitted symbols x_i are distorted by additive noise v_i and by signal dependent offsets b_{x_i}. In other words, the received symbols read

$$r_i = x_i + v_i + b_{x_i}, \tag{4.1}$$

for $i = 1, \ldots, n$. The offset b_{x_i} equals b_j, where $j \in [q]$, depending on the value of x_i. Let

$$\mathbf{b_x} = (b_{x_1}, b_{x_2}, \ldots, b_{x_n})$$

denote the offset vector when $\mathbf{x}$ is transmitted. For example, $\mathbf{b_x} = (b_0, b_1, b_0, b_1)$ if $\mathbf{x} = (0, 1, 0, 1)$. All values of offset, b_j, $j \in [q]$, which neither the transmitter nor the receiver knows, may vary from one transmitted codeword to the next one, but they do not vary within a codeword length of n. The received vector when a codeword $\mathbf{x}$ is transmitted is

$$\mathbf{r} = \mathbf{x} + \mathbf{v} + \mathbf{b_x}, \tag{4.2}$$

where $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is the noise vector.

The uniform distribution and the Gaussian distribution are two classical probability distributions of stochastic processes used to model real-world noise and offset phenomena. Which of the two is the most appropriate depends on the situation under consideration.

This chapter first studies an ML decoding criterion for channels with uniform noise and signal dependent offset. In particular, for the proposed ML decoder, bounds are determined on the standard deviations of the noise and the offset, which lead to a word error rate equal to zero.

Secondly, the situation of Gaussian noise and signal dependent offset is discussed, where the correlation between different offset random variables is considered. In this case, we show that an ML criterion in the prior art can be derived as a special case of our decoding criterion obtained by letting offsets be identically fully correlated distributed. Further, an option to effectively reduce the complexity is considered for uncorrelated offsets.

The remainder of the chapter is organized as follows. In Section 4.2, we present an ML decoding criterion for channels with uniform noise and signal dependent offset. A zero WER is shown to be achievable if the standard deviations of the noise and offset satisfy certain conditions in Subsection 4.2.2. We present, in Section 4.3, the ML decoding criterion for such channels, where the noise and signal dependent offset are Gaussian distributed. Furthermore, in Subsection 4.3.1, we focus on a special case when the offsets are identical, followed by a complexity discussion in Subsection 4.3.2. Conclusion in Section 4.4 terminates the chapter.

4.2. Maximum Likelihood Decoding For Channels with Uniform Noise and Signal Dependent Offset

This section presents an ML decoding criterion for channels where noise and signal dependent offsets are uniformly distributed. A consequence of having bounded noise and offset is that a zero word error rate (WER) is achievable under certain constraints. A major result of this work is that we provide sufficient conditions on the standard deviations of the noise and offset in combination with the code properties to guarantee the zero WER.

For the noise vector $\mathbf{v} = (v_1, \ldots, v_n)$, we assume that the v_i are independently uniformly distributed with mean 0 and variance σ^2. Hence, the probability density function of each v_i, $i = 1, 2, \ldots, n$, is

$$\upsilon(v_i) = \begin{cases} \dfrac{1}{2\sqrt{3}\sigma}, & -\sqrt{3}\sigma < v_i < \sqrt{3}\sigma, \\ 0, & \text{otherwise,} \end{cases} \tag{4.3}$$

leading to a probability density function $\chi(\mathbf{v}) = \prod_{i=1}^{n} \upsilon(v_i)$ for $\mathbf{v}$.

We assume that the b_j are independently uniformly distributed with mean 0 and standard deviations β_j. The probability density function of each b_j, $j \in [q]$. is

$$\zeta(b_j) = \begin{cases} \dfrac{1}{2\sqrt{3}\beta_j}, & -\sqrt{3}\beta_j < b_j < \sqrt{3}\beta_j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

Note that the offset values, b_j, $j \in [q]$, may vary from codeword to codeword, but that for each j, it is fixed within a codeword of length n. Like the noise values, the offset values are unknown to both the receiver and the transmitter.

4.2.1. Decoding Criterion

For $j \in [q]$, let $\hat{X}^{(j)}$ denote the index set indicating the positions in $\mathbf{x}$ for which the symbol value equals j. For example, in case $q = 2$,

$$\hat{X}^{(0)} = \{1, 3\} \text{ and } \hat{X}^{(1)} = \{2, 4\}$$

when $\mathbf{x} = (0,1,0,1)$.

For the received vector $\mathbf{r}$ and a candidate codeword $\hat{\mathbf{x}} \in S$, we define

$$\begin{aligned} u_j(\mathbf{r},\hat{\mathbf{x}}) &= \min\left(\{r_i - \hat{x}_i + \sqrt{3}\sigma \,\middle|\, i \in \hat{X}^{(j)}\} \cup \{\sqrt{3}\beta_j\}\right), \\ l_j(\mathbf{r},\hat{\mathbf{x}}) &= \max\left(\{r_i - \hat{x}_i - \sqrt{3}\sigma \,\middle|\, i \in \hat{X}^{(j)}\} \cup \{-\sqrt{3}\beta_j\}\right), \end{aligned}$$

for $j \in [q]$. These parameters correspond to upper and lower bounds on the possible values of b_j for $\hat{\mathbf{x}}$ when $\mathbf{r}$ is received. Further, let

$$m_j(\mathbf{r},\hat{\mathbf{x}}) = \max\{u_j(\mathbf{r},\hat{\mathbf{x}}), l_j(\mathbf{r},\hat{\mathbf{x}})\},$$

and

$$I_j(\mathbf{r},\hat{\mathbf{x}}) = \max\{u_j(\mathbf{r},\hat{\mathbf{x}}) - l_j(\mathbf{r},\hat{\mathbf{x}}), 0\}$$

for $j \in [q]$. Next, we present a decoding criterion and show that it is ML for the channel under consideration.

Theorem 4. *If the noise and the offsets in (4.2) have probability density functions as* (4.3) *and* (4.4)*, respectively, then ML decoding is achieved by maximizing*

$$\prod_{j=0}^{q-1} I_j(\mathbf{r},\hat{\mathbf{x}}) \tag{4.5}$$

over all codewords $\hat{\mathbf{x}} \in S$.

Proof. If a vector $\mathbf{r}$ is received, ML decoding must determine a codeword $\hat{\mathbf{x}} \in S$ maximizing $\mathbb{P}(\mathbf{r}|\hat{\mathbf{x}})$, that is, the probability that $\mathbf{r}$ is received, given $\hat{\mathbf{x}}$ is sent. This satisfies

$$\begin{aligned} \mathbb{P}(\mathbf{r}|\hat{\mathbf{x}}) &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \chi(\mathbf{r} - \hat{\mathbf{x}} - b_{\hat{\mathbf{x}}}) \prod_{j=0}^{q-1} \left(\zeta(b_j)\,db_j\right) \\ &= \frac{1}{(2\sqrt{3}\sigma)^n} \prod_{j=0}^{q-1} \left(\int_{l_j(\mathbf{r},\hat{\mathbf{x}})}^{m_j(\mathbf{r},\hat{\mathbf{x}})} \frac{1}{2\sqrt{3}\beta_j}\,db_j \right) \\ &= \frac{\prod_{j=0}^{q-1} I_j(\mathbf{r},\hat{\mathbf{x}})}{(2\sqrt{3}\sigma)^n (2\sqrt{3})^q \prod_{j=0}^{q-1} \beta_j}. \end{aligned} \tag{4.6}$$

The first equality is due to the channel model (4.2). The second equality follows from the probability density functions (4.3) and (4.4). The third equality follows from the observation that

$$\int_{l_j(\mathbf{r},\hat{\mathbf{x}})}^{m_j(\mathbf{r},\hat{\mathbf{x}})} db_j = m_j(\mathbf{r},\hat{\mathbf{x}}) - l_j(\mathbf{r},\hat{\mathbf{x}}) = I_j(\mathbf{r},\hat{\mathbf{x}})$$

for all j. Since the denominator in (4.6) is a constant term for all candidate codewords, we can ignore it during the maximization process, which gives (4.5). □

The bounded nature of the noise and the offset has interesting consequences with respect to the WER of the ML decoder, as will be further explored in the following subsection.

4.2.2. ZERO ERROR ANALYSIS

Most interestingly, for the ML decoder based on (4.5), a word error rate (WER) of zero is achieved if the standard deviations of the noise and the offset satisfy certain conditions. This is shown in the following theorem.

Theorem 5. *If the noise and the offsets in (4.2) have probability density functions as* (4.3) *and* (4.4), *respectively, with*

$$\sigma \leq \min_{\substack{\mathbf{s},\mathbf{c}\in S\\ \mathbf{s}\neq\mathbf{c}}} \left(\frac{\max_{i\in\{1,\dots,n\}} \{|c_i - s_i| - \sqrt{3}(\beta_{c_i} + \beta_{s_i})\}}{2\sqrt{3}} \right) \tag{4.7}$$

or

$$\sigma \leq \min_{\substack{\mathbf{s},\mathbf{c}\in S\\ \mathbf{s}\neq\mathbf{c}}} \left(\frac{\max_{j\in[q]} \left\{ \max_{i,k\in\mathbf{c}^{(j)}} \{s_k - s_i - \sqrt{3}(\beta_{s_k} + \beta_{s_i})\} \right\}}{4\sqrt{3}} \right), \tag{4.8}$$

then the ML decoder achieves a WER equal to zero.

Proof. Assume that $\mathbf{x} \in S$ is sent and $\mathbf{r} = \mathbf{x} + \mathbf{v} + \mathbf{b_x}$ is received. We will show that if (4.7) or (4.8) holds, then $\mathbb{P}(\mathbf{r}|\hat{\mathbf{x}}) = 0$ for all codewords $\hat{\mathbf{x}} \neq \mathbf{x}$. First of all, note that

$$\begin{aligned}
& u_j(\mathbf{r},\hat{\mathbf{x}}) - l_j(\mathbf{r},\hat{\mathbf{x}}) \\
&= \min\left(\{r_i - \hat{x}_i + \sqrt{3}\sigma \,\middle|\, i \in \hat{X}^{(j)}\} \cup \{\sqrt{3}\beta_j\}\right) \\
&\quad - \max\left(\{r_i - \hat{x}_i - \sqrt{3}\sigma \,\middle|\, i \in \hat{X}^{(j)}\} \cup \{-\sqrt{3}\beta_j\}\right) \\
&= \min\left(\{r_i - \hat{x}_i + \sqrt{3}\sigma \,\middle|\, i \in \hat{X}^{(j)}\} \cup \{\sqrt{3}\beta_j\}\right) \\
&\quad + \min\left(\{-(r_i - \hat{x}_i) + \sqrt{3}\sigma \,\middle|\, i \in \hat{X}^{(j)}\} \cup \{\sqrt{3}\beta_j\}\right) \\
&= \min\Bigg(\{2\sqrt{3}\beta_j\} \cup \{\min_{i\in\hat{X}^{(j)}} \{-|r_i - \hat{x}_i| + \sqrt{3}\sigma + \sqrt{3}\beta_j\}\} \\
&\quad \cup \{\min_{i,k\in\hat{X}^{(j)}} \{(r_i - \hat{x}_i) - (r_k - \hat{x}_k) + 2\sqrt{3}\sigma\}\}\Bigg), \\
&= \min\Bigg(\{2\sqrt{3}\beta_j\} \cup \{\min_{i\in\hat{X}^{(j)}} \{-|r_i - \hat{x}_i| + \sqrt{3}\sigma + \sqrt{3}\beta_j\}\} \\
&\quad \cup \{\min_{i,k\in\hat{X}^{(j)}} \{r_i - r_k + 2\sqrt{3}\sigma\}\}\Bigg),
\end{aligned} \tag{4.9}$$

for $j \in [q]$.

Next, we show that if (4.7) or (4.8) holds, (4.9) will be negative for some j whenever $\hat{\mathbf{x}} \neq \mathbf{x}$. Note that the final expression in (4.9) contains a union of three terms, where the first term is always positive since β_j is positive. We show that the second term is negative for some j if (4.7) holds and that the third term is negative for some j if (4.8) holds.

For each $\hat{\mathbf{x}} \in S$, $\hat{\mathbf{x}} \neq \mathbf{x}$, let j_0 be a symbol from $[q]$ such that a position i_0 in $\hat{X}^{(j_0)} \subseteq \{1,\ldots,n\}$ maximizes the expression $|x_i - \hat{x}_i| - \sqrt{3}(\beta_{x_i} + \beta_{\hat{x}_i})$. That is,

$$\begin{aligned} i_0 &= \arg \max_{i\in\{1,\ldots,n\}} \left\{|x_i - \hat{x}_i| - \sqrt{3}(\beta_{x_i} + \beta_{\hat{x}_i})\right\}, \\ j_0 &= \hat{x}_{i_0}. \end{aligned}$$

Note that this j_0 is not necessarily the same for each $\hat{\mathbf{x}}$. If (4.7) holds, then we have

$$\begin{aligned} &\min_{i\in\hat{X}^{(j_0)}} \{-|r_i - \hat{x}_i| + \sqrt{3}\sigma + \sqrt{3}\beta_j\} \\ &= \min_{i\in\hat{X}^{(j_0)}} \{-|r_i - \hat{x}_i| - \sqrt{3}\sigma - \sqrt{3}\beta_{x_i} + \sqrt{3}(2\sigma + \beta_{x_i} + \beta_j)\} \\ &< \min_{i\in\hat{X}^{(j_0)}} \{-|r_i - \hat{x}_i| - \left|v_i + b_{x_i}\right| + \sqrt{3}(2\sigma + \beta_{x_i} + \beta_j)\} \\ &= \min_{i\in\hat{X}^{(j_0)}} \{-|r_i - \hat{x}_i| - |r_i - x_i| + \sqrt{3}(2\sigma + \beta_{x_i} + \beta_j)\} \\ &\le \min_{i\in\hat{X}^{(j_0)}} \{-|x_i - \hat{x}_i| + \sqrt{3}(2\sigma + \beta_{x_i} + \beta_j)\} \\ &= -\max_{i\in\hat{X}^{(j_0)}} \{|x_i - \hat{x}_i| - \sqrt{3}(\beta_{x_i} + \beta_{\hat{x}_i})\} + 2\sqrt{3}\sigma \\ &= -\max_{i\in\{1,\ldots,n\}} \{|x_i - \hat{x}_i| - \sqrt{3}(\beta_{x_i} + \beta_{\hat{x}_i})\} + 2\sqrt{3}\sigma \\ &\le 0. \end{aligned}$$

The first inequality follows from the fact that $|v_i + b_{x_i}| \le |v_i| + |b_{x_i}| < \sqrt{3}\sigma + \sqrt{3}\beta_{x_i}$, the second inequality from the triangular inequality, and the last inequality from (4.7). Thus the second term of (4.9) is negative for some j whenever $\hat{\mathbf{x}} \neq \mathbf{x}$ if (4.7) holds.

Similarly, for each $\hat{\mathbf{x}} \in S, \hat{\mathbf{x}} \neq \mathbf{x}$, let j_1 be a symbol from $[q]$ such that

$$j_1 = \arg\max_{j\in[q]} \left\{ \max_{i,k\in\hat{X}^{(j)}} \{x_k - x_i - \sqrt{3}(\beta_{x_k} + \beta_{x_i})\} \right\}.$$

If (4.8) holds, then we have

$$\begin{aligned} &\min_{i,k\in\hat{X}^{(j_1)}} \{r_i - r_k + 2\sqrt{3}\sigma\} \\ &< \min_{i,k\in\hat{X}^{(j_1)}} \{r_i - r_k - (v_i - v_k)\} + 4\sqrt{3}\sigma \\ &= \min_{i,k\in\hat{X}^{(j_1)}} \{x_i - x_k + b_{x_i} - b_{x_k}\} + 4\sqrt{3}\sigma \\ &< \min_{i,k\in\hat{X}^{(j_1)}} \{x_i - x_k + \sqrt{3}(\beta_{x_i} + \beta_{x_k})\} + 4\sqrt{3}\sigma \\ &= -\max_{i,k\in\hat{X}^{(j_1)}} \{x_k - x_i - \sqrt{3}(\beta_{x_i} + \beta_{x_k})\} + 4\sqrt{3}\sigma \\ &= -\max_{j\in[q]} \left\{ \max_{i,k\in\hat{X}^{(j)}} \{x_k - x_i - \sqrt{3}(\beta_{x_i} + \beta_{x_k})\} \right\} + 4\sqrt{3}\sigma \\ &\le 0. \end{aligned}$$

The first inequality follows because $v_i - v_k \leq |v_i| + |v_k| < 2\sqrt{3}\sigma$, the second inequality follows because $b_{x_i} - b_{x_k} \leq \left|b_{x_i}\right| + \left|b_{x_k}\right| < \sqrt{3}(\beta_{x_i} + \beta_{x_k})$, and the last one from (4.8).

In conclusion, we have for any codeword $\hat{\mathbf{x}} \neq \mathbf{x}$ that $I_j(\mathbf{r}, \hat{\mathbf{x}}) = 0$ for some j if (4.7) or (4.8) holds. Hence,

$$\prod_{j=0}^{q-1} I_j(\mathbf{r}, \hat{\mathbf{x}}) = \mathbb{P}(\mathbf{r}|\hat{\mathbf{x}}) = 0$$

for all codewords $\hat{\mathbf{x}} \neq \mathbf{x}$, while

$$\prod_{j=0}^{q-1} I_j(\mathbf{r}, \mathbf{x}) > 0.$$

Hence, the transmitted codeword is always the outcome of the decoding procedure maximizing (4.5), and thus ML decoding achieves a WER equal to zero. □

4

Next, we give a sufficient condition to achieve zero WER for the ML decoder in the binary case, i.e., $q = 2$.

Corollary 6. *If the noise and the offsets have probability density functions as* (4.3) *and* (4.4)*, respectively, with*

$$2\sigma + \beta_0 + \beta_1 \leq \frac{1}{\sqrt{3}}, \tag{4.10}$$

then the ML decoder achieves a WER equal to zero for a binary codebook.

Proof. In the binary case, the expression $|c_i - s_i| - \sqrt{3}(\beta_{c_i} + \beta_{s_i})$ in (4.7) has one of three values depending on c_i and s_i, i.e.,

$$|c_i - s_i| - \sqrt{3}(\beta_{c_i} + \beta_{s_i}) = \begin{cases} -2\sqrt{3}\beta_0, & \text{if } (c_i, s_i) = (0,0); \\ -2\sqrt{3}\beta_1, & \text{if } (c_i, s_i) = (1,1); \\ 1 - \sqrt{3}(\beta_0 + \beta_1), & \text{otherwise.} \end{cases} \tag{4.11}$$

Since β_0 and β_1 are both positive, $-2\sqrt{3}\beta_0$ and $-2\sqrt{3}\beta_1$ are both negative, and thus if (4.10) holds then it immediately follows from the fact that σ is positive as well that

$$1 - \sqrt{3}(\beta_0 + \beta_1) > 0.$$

For any codewords $\mathbf{s}$ and $\mathbf{c} \neq \mathbf{s}$, there exists at least one position, k, such that $c_k \neq s_k$, and then we have $|c_k - s_k| = 1$ and $\beta_{c_k} + \beta_{s_k} = \beta_0 + \beta_1$. In conclusion, if (4.10) holds, maximizing (4.11) over $i \in \{1, \ldots, n\}$ outputs $1 - \sqrt{3}(\beta_0 + \beta_1)$ as its maximum value for any codewords $\mathbf{s}$ and $\mathbf{c} \neq \mathbf{s}$. Hence, according to Theorem 5, the ML decoder achieves a WER equal to zero when (4.10) holds. □

Theorem 5 has important implications for developing zero WER codes for channels suffering from uniform noise and signal dependent offset. Code design for these channels is beyond the scope of this work, but in the next subsection, we provide a performance analysis for a simple example code to show the advantage of the ML decoding technique compared to the MED and MMPD decoders. Also, it will be illustrated that a zero WER indeed appears in cases that the standard deviations of the noise and the offsets are sufficiently small.

4

4.2.3. PERFORMANCE EVALUATION

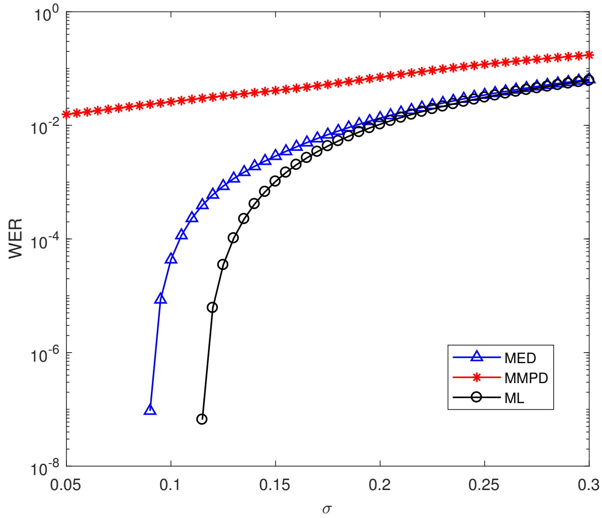


Figure 4.1: WER versus the standard deviation σ of the uniform noise for S^* in combination with MED, MMPD, and ML decoders, when the standard deviations of the uniform offsets are $\beta_0 = 0.2$, $\beta_1 = 0.15$.

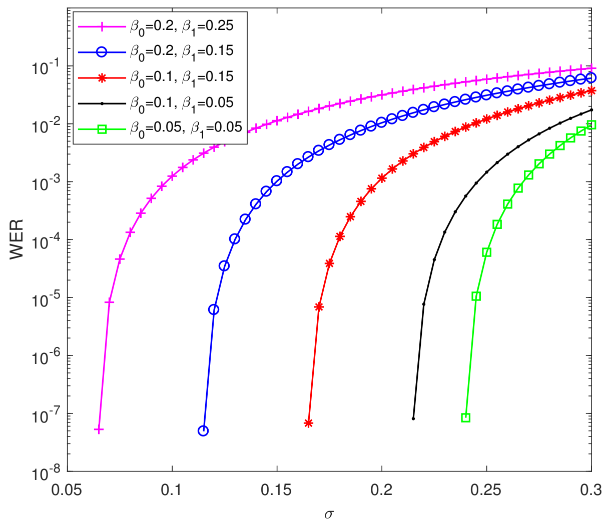


Figure 4.2: WER versus the standard deviation σ of the uniform noise for S^* in combination with ML decoding, with different values of the uniform offset standard deviations β_0, β_1.

Simulated WER results as a function of σ are shown in Fig. 4.1 for a binary code – (3,2) parity check code S^* of length 3 and size 4, in combination with MED, MMPD, and ML decoders. The standard deviations of the signal dependent offsets are set to $\beta_0 = 0.2$ and $\beta_1 = 0.15$. We observe that the performance of each of the three decoders declines with increasing values of σ. The MMPD decoder has the worst performance among these three. The performance of the proposed ML decoder is better than for the MED and MMPD decoders.

Simulation results for S^* with various values of β_0 and β_1 are shown in Fig. 4.2. The results from Fig. 4.2 confirm Corollary 6. Zero WER of the ML decoder is indeed achieved if $2\sigma + \beta_0 + \beta_1 \leq 1/\sqrt{3} \approx 0.58$. We can also observe this in Fig. 4.1, where for the ML decoder, zero WER is achieved when the value of σ is less than $(1/\sqrt{3} - 0.20 - 0.15)/2 \approx 0.11$.

So far, we have studied an ML decoding criterion for channels with uniform noise and signal dependent offset. The situation of Gaussian noise and signal dependent offset is discussed in the next section.

4.3. Maximum Likelihood Decoding For Channels with Gaussian Noise and Signal Dependent Offset

This section is still concerned with the situation that the offset is a signal dependent parameter. We assume that both the noise and the offset are Gaussian distributed. Consider transmitting a codeword $\mathbf{x}$ from a binary code $S \subseteq [2]^n$, where $[2]$ is the binary alphabet. We pursue the binary alphabet here as it is enough to demonstrate some important results. The extension of this channel model to a q-ary alphabet can be explored similarly as manifested for the binary case in this section.

The noise samples v_i are i.i.d. Gaussian random variables with zero mean and variance σ^2, i.e., $v_i \sim \mathcal{N}(0, \sigma^2)$. Throughout the transmitted codeword, the noise is independent of the offsets. Note that the noise value varies from symbol to symbol, while the offsets b_0 and b_1 are assumed to be constant for all '0' and '1' symbols within a codeword, respectively. We assume that the offsets, b_0 and b_1, are Gaussian distributed with mean 0 and variances β_0^2 and β_1^2, respectively. The probability density functions of b_0 and b_1 are

$$\zeta(b_i) = \frac{1}{\beta_i\sqrt{2\pi}} e^{-b_i^2/(2\beta_i^2)}, \tag{4.12}$$

where $i = 0, 1$. The correlation between b_0 and b_1 is

$$\rho = \frac{\mathrm{cov}(b_0, b_1)}{\beta_0\beta_1},$$

where $\mathrm{cov}(b_0, b_1)$ is the covariance of b_0 and b_1. The correlation is bounded by $-1 \leq \rho \leq 1$. We have $\rho = 0$ when b_0 and b_1 are uncorrelated, $\rho = 1$ when they are fully correlated, and $\rho = -1$ when they are completely anti-correlated.

Based on the channel model (4.2), we define the total distortion as

$$\mathbf{d_x} = \mathbf{v} + \mathbf{b_x} = \mathbf{r} - \mathbf{x}.$$

Then the probability density function of $\mathbf{d_x}$ is given by

$$\psi(\mathbf{d_x}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(\mathbf{d_x} - \mathbf{b_x})\zeta(b_0)\zeta(b_1)\,\mathrm{d}b_0\,\mathrm{d}b_1. \tag{4.13}$$

An ML decoder will choose a codeword maximizing the probability density function (4.13), that is,

$$\arg\max_{\hat{\mathbf{x}}\in S} \psi(\mathbf{d}_{\hat{\mathbf{x}}}).$$

In the situation of zero-mean Gaussian noise samples and signal dependent offset, $\mathbf{d}_{\hat{\mathbf{x}}}$ has a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and covariance matrix $\Sigma_{\hat{\mathbf{x}}}$. Since the noise is independent of the offsets and the correlation coefficient between the offset values is ρ, $\Sigma_{\hat{\mathbf{x}}}$ is an $n \times n$ matrix with the (i, j)th entry specified by

$$\Sigma_{\hat{\mathbf{x}}}(i,j) = \begin{cases} \beta_0^2 & \text{if } i \neq j, i, j \in \hat{X}^{(0)}, \\ \beta_1^2 & \text{if } i \neq j, i, j \in \hat{X}^{(1)}, \\ \sigma^2 + \beta_0^2 & \text{if } i = j \in \hat{X}^{(0)}, \\ \sigma^2 + \beta_1^2 & \text{if } i = j \in \hat{X}^{(1)}, \\ \rho\beta_0\beta_1 & \text{otherwise,} \end{cases} \tag{4.14}$$

4

where $\hat{X}^{(1)}$ and $\hat{X}^{(0)}$ are index sets, indicating the positions of the ones and zeroes in $\hat{\mathbf{x}}$, respectively. Thus, the probability density function of $\mathbf{d}_{\hat{\mathbf{x}}}$ is

$$\psi(\mathbf{d}_{\hat{\mathbf{x}}}) = \frac{\exp(-\mathbf{d}_{\hat{\mathbf{x}}}\Sigma_{\hat{\mathbf{x}}}^{-1}\mathbf{d}_{\hat{\mathbf{x}}}^{\mathrm{T}}/2)}{\sqrt{(2\pi)^n(\det\Sigma_{\hat{\mathbf{x}}})}}, \tag{4.15}$$

where $\Sigma_{\hat{\mathbf{x}}}^{-1}$ is the inverse matrix of $\Sigma_{\hat{\mathbf{x}}}$ and $\det\Sigma_{\hat{\mathbf{x}}}$ is the determinant of $\Sigma_{\hat{\mathbf{x}}}$.

Before working out this expression, we first introduce some further notations. Let ω denote the weight of $\hat{\mathbf{x}}$, i.e., the size of set $\hat{X}^{(1)}$. Clearly, the size of set $\hat{X}^{(0)}$ is $n-\omega$. According to $\hat{X}^{(1)}$ and $\hat{X}^{(0)}$, we can cluster symbols of the received vector. Specifically, symbols of the received vector at positions where the values of $\hat{x}_i$ are 0 are grouped in one category, and the rest of the symbols form another category. We focus on the average values of these two categories and define two quantities, namely, the average value of received symbols in the '1' positions of $\hat{\mathbf{x}}$, i.e.,

$$\overline{\mathbf{r}^{(1)}} = \frac{1}{\omega}\sum_{i\in\hat{X}^{(1)}} r_i, \tag{4.16}$$

and the average value of received symbols in the '0' positions of $\hat{\mathbf{x}}$, i.e.,

$$\overline{\mathbf{r}^{(0)}} = \frac{1}{n-\omega}\sum_{i\in\hat{X}^{(0)}} r_i. \tag{4.17}$$

Finally, λ_0 and λ_1 are the ratios of noise variance to offset variances, i.e.,

$$\lambda_0 = \sigma^2/\beta_0^2, \tag{4.18}$$

$$\lambda_1 = \sigma^2/\beta_1^2. \tag{4.19}$$

When the channels suffer from the signal dependent offsets, the positions of the zeroes and the ones in a candidate codeword $\hat{\mathbf{x}}$ are critical factors for the ML decoding. The weight of the codeword is a critical issue as well. Attention is drawn to these factors in the following theorem, where we present the ML decoding criterion for channels with Gaussian noise and signal dependent offsets.

Theorem 7. *In the case that the i.i.d. noise* $v_i \sim \mathcal{N}(0,\sigma^2)$*, the offsets* $b_0 \sim \mathcal{N}(0,\beta_0^2)$*,* $b_1 \sim \mathcal{N}(0,\beta_1^2)$*, and the correlation between* b_0 *and* b_1 *is* ρ*, ML decoding is achieved by minimizing*

$$\begin{aligned}\ln\eta + \frac{1}{\sigma^2}\Bigg[\delta_E(\mathbf{r},\hat{\mathbf{x}}) &- \frac{\lambda_0+(1-\rho^2)(n-\omega)}{\eta}\omega^2(\overline{\mathbf{r}}^{(1)}-1)^2 \\ &- \frac{\lambda_1+(1-\rho^2)\omega}{\eta}(n-\omega)^2\overline{\mathbf{r}}^{(0)2} \\ &- \frac{2\rho\sqrt{\lambda_0\lambda_1}}{\eta}\omega(n-\omega)(\overline{\mathbf{r}}^{(1)}-1)\overline{\mathbf{r}}^{(0)}\Bigg],\end{aligned} \tag{4.20}$$

over all candidate codewords $\hat{\mathbf{x}} \in S$*, where* $\eta = \lambda_1\lambda_0 + \omega\lambda_0 + (n-\omega)\lambda_1 + \omega(n-\omega)(1-\rho^2)$.

Before proving this theorem, an example of the covariance matrix (4.14) is given here.

Example 8. *Consider a candidate codeword* $\hat{\mathbf{x}} = (1,0,0,1)$ *and a received vector* $\mathbf{r}$*. Then*

$$\mathbf{d}_{\hat{\mathbf{x}}} = \mathbf{r} - \hat{\mathbf{x}} = (r_1 - 1, r_2, r_3, r_4 - 1).$$

For this example codeword, we have $\hat{X}^{(1)} = \{1,4\}$ *and* $\hat{X}^{(0)} = \{2,3\}$*. Thus, based on (4.14), the covariance matrix of* $\mathbf{d}_{\hat{\mathbf{x}}}$ *is*

$$\Sigma_{\hat{\mathbf{x}}} = \begin{bmatrix} \sigma^2+\beta_1^2 & \rho\beta_0\beta_1 & \rho\beta_0\beta_1 & \beta_1^2 \\ \rho\beta_0\beta_1 & \sigma^2+\beta_0^2 & \beta_0^2 & \rho\beta_0\beta_1 \\ \rho\beta_0\beta_1 & \beta_0^2 & \sigma^2+\beta_0^2 & \rho\beta_0\beta_1 \\ \beta_1^2 & \rho\beta_0\beta_1 & \rho\beta_0\beta_1 & \sigma^2+\beta_1^2 \end{bmatrix}. \tag{4.21}$$

Further, the average value of received symbols in the '1' positions is

$$\overline{\mathbf{r}^{(1)}} = (r_1 + r_4)/2$$

and the average value of received symbols in the '0' positions is

$$\overline{\mathbf{r}^{(0)}} = (r_2 + r_3)/2.$$

Thus the ML measurement for $\hat{\mathbf{x}} = (1,0,0,1)$ *according to Theorem 7 is*

$$\begin{aligned}\ln\eta + \frac{1}{\sigma^2}\Bigg[\delta_E(\mathbf{r},\hat{\mathbf{x}}) &- \frac{\lambda_0+2(1-\rho^2)}{\eta}(r_1+r_4-2)^2 \\ &- \frac{\lambda_1+2(1-\rho^2)}{\eta}(r_2+r_3)^2 \\ &- \frac{2\rho\sqrt{\lambda_0\lambda_1}}{\eta}(r_1+r_4-2)(r_2+r_3)\Bigg],\end{aligned}$$

where $\eta = \lambda_1\lambda_0 + 2\lambda_0 + 2\lambda_1 + 4(1-\rho^2)$.

We may assume without loss of generality that $\hat{\mathbf{x}}$ is rearranged such that $\hat{x}_1 \geq \hat{x}_2 \geq \ldots \geq \hat{x}_n$, as long as $\mathbf{r}$ and thus $\mathbf{d}_{\hat{\mathbf{x}}} = \mathbf{r} - \hat{\mathbf{x}}$ are rearranged according to the same permutation as $\hat{\mathbf{x}}$. Throughout the rest of this section, we will do so since it allows a more convenient representation of the covariance matrix. In the example as just presented, one possible permutation of $\hat{\mathbf{x}}$ is

$$(\hat{x}_1, \hat{x}_4, \hat{x}_2, \hat{x}_3) = (1,1,0,0).$$

Then the corresponding representation of $\mathbf{d}_{\hat{\mathbf{x}}}$ is

$$(r_1 - 1, r_4 - 1, r_2, r_3),$$

and its covariance matrix yields

$$\Sigma_{\hat{\mathbf{x}}} = \left[\begin{array}{cc|cc} \sigma^2 + \beta_1^2 & \beta_1^2 & \rho\beta_0\beta_1 & \rho\beta_0\beta_1 \\ \beta_1^2 & \sigma^2 + \beta_1^2 & \rho\beta_0\beta_1 & \rho\beta_0\beta_1 \\ \hline \rho\beta_0\beta_1 & \rho\beta_0\beta_1 & \sigma^2 + \beta_0^2 & \beta_0^2 \\ \rho\beta_0\beta_1 & \rho\beta_0\beta_1 & \beta_0^2 & \sigma^2 + \beta_0^2 \end{array}\right]. \tag{4.22}$$

Proof of Theorem 7. We start the evaluation of (4.15) by considering the covariance matrix $\Sigma_{\hat{\mathbf{x}}}$ given in (4.14). Since the entries of this matrix are assigned values according to two index sets $\hat{X}^{(1)}$ and $\hat{X}^{(0)}$, each $\Sigma_{\hat{\mathbf{x}}}$ is interpreted being subdivided into four blocks, that is,

$$\Sigma_{\hat{\mathbf{x}}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix},$$

where $\mathbf{A}$ is an $\omega \times \omega$ matrix with all entries on the main diagonal equal to $\sigma^2 + \beta_1^2$ and all other entries equal to β_1^2, $\mathbf{D}$ is an $(n-\omega) \times (n-\omega)$ matrix with all entries on the main diagonal equal to $\sigma^2 + \beta_0^2$ and all other entries equal to β_0^2, and $\mathbf{B}$, $\mathbf{C}$ are matrices of sizes $\omega \times (n-\omega)$ and $(n-\omega) \times \omega$, respectively, with all entries equal to $\rho\beta_0\beta_1$. An example of such a block structure can be found in (4.22).

If $\mathbf{A}$ and $\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$ are non-singular, then the inverse and determinant of $\Sigma_{\hat{\mathbf{x}}}$ are [77, pp. 107-108]

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\mathbf{X}^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{X}^{-1} \\ -\mathbf{X}^{-1}\mathbf{C}\mathbf{A}^{-1} & \mathbf{X}^{-1} \end{bmatrix}, \tag{4.23}$$

and

$$\det \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = (\det \mathbf{A})(\det \mathbf{X}), \tag{4.24}$$

where $\mathbf{X} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$. We first show that $\mathbf{A}$ and $\mathbf{X}$ are non-singular matrices, and then use the above formulas to calculate the inverse matrix and determinant of $\Sigma_{\hat{\mathbf{x}}}$.

The determinant and the inverse matrix of $\mathbf{A}$ are

$$\det \mathbf{A} = \sigma^{2(\omega-1)}(\sigma^2 + \omega\beta_1^2)$$

and

$$\mathbf{A}^{-1} = \begin{bmatrix} s & t & \cdots & t \\ t & s & & \vdots \\ \vdots & & \ddots & t \\ t & \cdots & t & s \end{bmatrix},$$

where $s = \frac{\sigma^2+(\omega-1)\beta_1^2}{\sigma^2(\sigma^2+\omega\beta_1^2)}$ and $t = \frac{-\beta_1^2}{\sigma^2(\sigma^2+\omega\beta_1^2)}$. $\mathbf{A}$ is a non-singular matrix since $\det\mathbf{A} \neq 0$.

Next, we investigate the matrix $\mathbf{X}$. After simple calculations, $\mathbf{X}$ can be written down as

$$\mathbf{X} = \begin{bmatrix} \sigma^2+\gamma & \gamma & \cdots & \gamma \\ \gamma & \sigma^2+\gamma & & \vdots \\ \vdots & & \ddots & \gamma \\ \gamma & \cdots & \gamma & \sigma^2+\gamma \end{bmatrix},$$

where $\gamma = \frac{\lambda_1+(1-\rho^2)\omega}{\lambda_1+\omega}\beta_0^2$. The determinant of $\mathbf{X}$ is

$$\det\mathbf{X} = \frac{\sigma^{2(n-\omega-1)}\beta_0^2\eta}{\lambda_1+\omega},$$

where $\eta = \lambda_0\lambda_1 + \omega\lambda_0 + (n-\omega)\lambda_1 + \omega(n-\omega)(1-\rho^2)$. The matrix $\mathbf{X}$ is singular only when $\eta = 0$. However, $\lambda_0\lambda_1 > 0$ implies that $\eta > 0$ which means that $\mathbf{X}$ is a non-singular matrix. Then the inverse matrix of $\mathbf{X}$ is

$$\mathbf{X}^{-1} = \begin{bmatrix} g & h & \cdots & h \\ h & g & & \vdots \\ \vdots & & \ddots & h \\ h & \cdots & h & g \end{bmatrix},$$

where $g = \frac{\sigma^2+(n-\omega-1)\gamma}{\sigma^2(\sigma^2+(n-\omega)\gamma)}$ and $h = \frac{-\gamma}{\sigma^2(\sigma^2+(n-\omega)\gamma)}$.

Since $\mathbf{A}$ and $\mathbf{X}$ are non-singular, we use (4.23) and (4.24) to calculate the determinant and the inverse matrix of $\Sigma_{\hat{\mathbf{x}}}$. Let γ' be $\frac{\lambda_0+(n-\omega)(1-\rho^2)}{\lambda_0+(n-\omega)}\beta_1^2$. Then we have

$$\mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\mathbf{X}^{-1}\mathbf{C}\mathbf{A}^{-1} = \begin{bmatrix} g' & h' & \cdots & h' \\ h' & g' & & \vdots \\ \vdots & & \ddots & h' \\ h' & \cdots & h' & g' \end{bmatrix},$$

where $g' = \frac{\sigma^2+(\omega-1)\gamma'}{\sigma^2(\sigma^2+\omega\gamma')}$ and $h' = \frac{-\gamma'}{\sigma^2(\sigma^2+\omega\gamma')}$. We also have that

$$-\mathbf{A}^{-1}\mathbf{B}\mathbf{X}^{-1} \text{ and } -\mathbf{X}^{-1}\mathbf{C}\mathbf{A}^{-1}$$

are two matrices of sizes $\omega \times (n-\omega)$ and $(n-\omega)\times\omega$, respectively, with all entries equal to

$$\frac{-\rho}{\beta_0\beta_1\eta}.$$

Since the logarithm function is strictly increasing on the positive real numbers and ψ is a positive function, ML decoding can also be achieved by maximizing the logarithm of (4.15), i.e.,

$$\ln\psi(\mathbf{d}_{\hat{\mathbf{x}}}) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln(\det\Sigma_{\hat{\mathbf{x}}}) - \frac{1}{2}\mathbf{d}_{\hat{\mathbf{x}}}\Sigma_{\hat{\mathbf{x}}}^{-1}\mathbf{d}_{\hat{\mathbf{x}}}^{\mathrm{T}},$$

4

rather than maximizing (4.15) itself. By inverting the sign and ignoring irrelevant terms, we find that maximizing $\ln\psi(\mathbf{d}_{\hat{\mathbf{x}}})$ is equivalent to minimizing

$$\ln\left(\det\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\right)+\sum_{i=1}^{n}\sum_{j=1}^{n}d_i\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}^{-1}_{ij}d_j, \tag{4.25}$$

where d_i is the i-th term in the vector $\mathbf{d}_{\hat{\mathbf{x}}}$. By applying (4.24), the first part of (4.25) is

$$\begin{aligned}\ln\left(\det\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\right) &= \ln((\det\mathbf{A})(\det\mathbf{X}))\\ &= \ln\eta+\ln(\sigma^{2(n-2)}\beta_1^2\beta_0^2).\end{aligned} \tag{4.26}$$

By ignoring the last term, that is irrelevant to the optimization process (independent of $\hat{\mathbf{x}}$), we have that minimizing (4.26) is equivalent to minimizing $\ln\eta$. The second part of (4.25) is more complicated, but we can use $g-h=g'-h'=1/\sigma^2$ to simplify several terms. Since the average value of the first ω symbols of $\mathbf{d}_{\hat{\mathbf{x}}}$ is $\overline{\mathbf{r}^{(1)}}-1$ and the average value of the other symbols is $\overline{\mathbf{r}^{(0)}}$, we have

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j=1}^{n}d_i\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}^{-1}_{ij}d_j\\ &=\left[(g'-h')\sum_{i=1}^{\omega}(d_i)^2+h'\omega^2(\overline{\mathbf{r}^{(1)}}-1)^2\right]\\ &\qquad+\left[(g-h)\sum_{i=\omega+1}^{n}(d_i)^2+h(n-\omega)^2(\overline{\mathbf{r}^{(0)}})^2\right]\\ &\qquad-\frac{2\rho}{\beta_0\beta_1\eta}\sum_{i=1}^{\omega}\sum_{j=\omega+1}^{n}d_id_j\\ &=\frac{1}{\sigma^2}\left[\delta_E(\mathbf{r},\hat{\mathbf{x}})-\frac{\lambda_0+(1-\rho^2)(n-\omega)}{\eta}\omega^2(\overline{\mathbf{r}^{(1)}}-1)^2\right.\\ &\qquad-\frac{\lambda_1+(1-\rho^2)\omega}{\eta}(n-\omega)^2\overline{\mathbf{r}^{(0)}}^2\\ &\qquad\left.-\frac{2\rho\sqrt{\lambda_0\lambda_1}}{\eta}\omega(n-\omega)(\overline{\mathbf{r}^{(1)}}-1)\overline{\mathbf{r}^{(0)}}\right].\end{aligned} \tag{4.27}$$

Finally, combining (4.26) and (4.27), we can conclude that maximizing $\psi(\mathbf{d}_{\hat{\mathbf{x}}})$ is equivalent to minimizing (4.20), as required. □

Next, we take a look at a special case where the offsets are identical.

4.3.1. Special Case Study

We consider a special situation of (4.2), where b_0 and b_1 are identical, i.e., $b_0=b_1=b$. This signal independent offset b is still assumed to be Gaussian distributed with zero mean and variance β^2. By definitions (4.18) and (4.19), the ratio of the noise and the offset variances is identical, i.e., $\lambda_0 = \lambda_1 = \lambda = \sigma^2/\beta^2$. The received vector is given by

$$\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1}. \tag{4.28}$$

In Section 3.3, an ML decoding criterion for channel model (4.28) is given. Here, we present this criterion in the following corollary and show that it appears as a special case of the result in Theorem 7.

Corollary 9. *In the case of i.i.d. noise $v_i \sim \mathcal{N}(0, \sigma^2)$ and offset $b \sim \mathcal{N}(0, \beta^2)$, an ML decoding criterion for channel model (4.28) is achieved by minimizing a weighted combination of the Euclidean distance (2.16) and the modified Pearson distance (2.39), i.e.,*

$$\frac{\lambda}{n+\lambda}\delta_E(\mathbf{r},\hat{\mathbf{x}}) + \frac{n}{n+\lambda}\delta'_P(\mathbf{r},\hat{\mathbf{x}}), \tag{4.29}$$

over all candidate codewords $\hat{\mathbf{x}}$.

4

Proof. Since the offset values for transmitted zeroes and ones are both equal to b, an ML criterion for (4.28) can be derived from Theorem 7 by substituting $\rho = 1$ and $\lambda_0 = \lambda_1 = \lambda$, which gives that the expression to be minimized is

$$\begin{aligned}
&\ln\eta + \frac{1}{\sigma^2}\left[\delta_E(\mathbf{r},\hat{\mathbf{x}}) - \frac{\lambda\omega^2}{\eta}(\overline{\mathbf{r}^{(1)}}-1)^2 - \frac{\lambda(n-\omega)^2}{\eta}\overline{\mathbf{r}^{(0)}}^2\right. \\
&\qquad\qquad\left. - \frac{2\lambda\omega(n-\omega)}{\eta}(\overline{\mathbf{r}^{(1)}}-1)\overline{\mathbf{r}^{(0)}}\right] \\
&= \ln\eta + \frac{1}{\sigma^2}\left[\delta_E(\mathbf{r},\hat{\mathbf{x}}) - \frac{(\omega(\overline{\mathbf{r}^{(1)}}-1) + (n-\omega)\overline{\mathbf{r}^{(0)}})^2}{\lambda+n}\right] \\
&= \ln\eta + \frac{1}{\sigma^2}\left[\delta_E(\mathbf{r},\hat{\mathbf{x}}) - \frac{n^2}{\lambda+n}(\bar{\mathbf{r}}-\bar{\hat{\mathbf{x}}})^2\right],
\end{aligned}$$

where $\eta = \lambda^2 + n\lambda$. Ignoring the irrelevant term $\ln\eta$ and dividing by $1/\sigma^2$ gives $\delta_E(\mathbf{r},\hat{\mathbf{x}}) - \frac{n^2}{\lambda+n}(\bar{\mathbf{r}}-\bar{\hat{\mathbf{x}}})^2$. Substituting $n(\bar{\mathbf{r}}-\bar{\hat{\mathbf{x}}})^2 = \delta_E(\mathbf{r},\hat{\mathbf{x}}) - \delta'_P(\mathbf{r},\hat{\mathbf{x}}) + n\bar{\mathbf{r}}^2$, which follows from (2.16) and (2.39), and ignoring the irrelevant term $-n^2\bar{\mathbf{r}}^2/(\lambda+n)$, gives (4.29). □

Recall that the Euclidean decoder (2.16) is ML in the situation that there is no offset, while the modified Pearson decoder (2.39) is ML when signals suffer from the identical offset. Here, the ML decoding is shown to be a balanced combination of these two criteria. In the offset dominant regime, i.e., $\beta \gg \sigma$ and thus λ being very small, (4.29) essentially reduces to the modified Pearson criterion from (2.25). On the other hand, in the noise dominant regime, i.e., $\beta \ll \sigma$ and thus λ being very large, (4.29) essentially reduces to the Euclidean criterion (2.16) [62].

4.3.2. Complexity Reduction

Minimization of (4.20) by an exhaustive search over all candidate codewords $\hat{\mathbf{x}} \in S$ may be too complex for large codebooks. In [49], it has been shown that the number of computations in order to minimize (2.25) can be significantly reduced by considering a structured codebook, which is the union of a number of constant composition codes, and applying Slepian's algorithm [37]. Such complexity reductions can be explored for the setting under consideration here as well. Below we will describe an example for a particular case of channel model (4.2). Similar results can also be obtained for the general case, but the corresponding formulas are more complicated and less readable.

We assume that the offsets b_0 and b_1 are identically distributed with zero means and variances β^2, In addition, we assume that they are uncorrelated random variables, i.e., $\rho = 0$. By setting $\lambda_0 = \lambda_1 = \lambda$ and $\rho = 0$ in Theorem 7, the ML decoding criterion for such a situation is thus established by minimizing

$$\begin{aligned}\delta_{\text{ML}}(\mathbf{r}, \hat{\mathbf{x}}) &= \ln(\lambda + \omega) + \ln(\lambda + n - \omega) \\ &+ \frac{1}{\sigma^2}\left[\delta_E(\mathbf{r}, \hat{\mathbf{x}}) - \frac{\omega^2}{\lambda + \omega}(\overline{\mathbf{r}^{(1)}} - 1)^2 - \frac{(n-\omega)^2}{\lambda + n - \omega}\overline{\mathbf{r}^{(0)}}^2\right],\end{aligned} \tag{4.30}$$

4

where ω is the weight of the candidate codeword $\hat{\mathbf{x}}$.

Let S_ω denote the set as

$$S_\omega = \{\mathbf{x} \in [2]^n : \sum_{i=1}^{n} x_i = \omega\}, \omega = 0, \dots, n.$$

Note that each of these S_ω contains *all* the vectors of length n and a particular weight ω. The codebook S under consideration is the union of $|V|$ of such sets, i.e.,

$$S = \bigcup_{\omega \in V} S_\omega,$$

where the index set $V \subseteq \{0, \dots, n\}$. Note that the index set V is of size at most $n+1$.

By working out (2.16), (4.16), and (4.17), we obtain

$$\delta_E(\mathbf{r}, \hat{\mathbf{x}}) = \sum_{i=1}^{n} (r_i - \hat{x}_i)^2 = \sum_{i=1}^{n} r_i^2 + n\bar{\hat{\mathbf{x}}} - 2\sum_{i=1}^{n} r_i \hat{x}_i,$$

$$\overline{\mathbf{r}^{(1)}} = \frac{\sum_{i=1}^{n} r_i \hat{x}_i}{n\bar{\hat{\mathbf{x}}}},$$

and

$$\overline{\mathbf{r}^{(0)}} = \frac{n\bar{\mathbf{r}} - \sum_{i=1}^{n} r_i \hat{x}_i}{n - n\bar{\hat{\mathbf{x}}}},$$

where the first equation is the squared Euclidean distance between $\mathbf{r}$ and $\hat{\mathbf{x}}$, and the second and the third equations are the average value of received symbols in the positions where $\hat{\mathbf{x}}$ has ones and zeroes, respectively.

Now, we first investigate the minimization of (4.30) over all candidate codewords $\hat{\mathbf{x}} \in$

S_ω for a fixed value of $\omega \in V$. For any $\hat{\mathbf{x}}$ in S_ω we have

$$\begin{aligned}
&\delta_{\mathrm{ML}}(\mathbf{r},\hat{\mathbf{x}}) \\
&= \ln(\lambda+\omega)+\ln(\lambda+n-\omega)+\frac{1}{\sigma^2}\left[\sum_{i=1}^{n} r_i^2+\omega-2\sum_{i=1}^{n} r_i\hat{x}_i\right. \\
&\left. -\frac{\omega^2}{\lambda+\omega}\left(\frac{\sum_{i=1}^{n} r_i\hat{x}_i}{\omega}-1\right)^2-\frac{(n-\omega)^2}{\lambda+n-\omega}\left(\frac{n\bar{\mathbf{r}}-\sum_{i=1}^{n} r_i\hat{x}_i}{n-\omega}\right)^2\right], \\
&= \ln(\lambda+\omega)+\ln(\lambda+n-\omega)+\frac{1}{\sigma^2}\left[\sum_{i=1}^{n} r_i^2+\omega-d(\mathbf{r},\hat{\mathbf{x}})\right], \qquad (4.31)
\end{aligned}$$

where

$$\begin{aligned}
d(\mathbf{r},\hat{\mathbf{x}}) = &\left(\frac{1}{\lambda+\omega}+\frac{1}{\lambda+n-\omega}\right)\left(\sum_{i=1}^{n} r_i\hat{x}_i\right)^2+ \qquad (4.32)\\
&\left(\frac{2\lambda}{\lambda+\omega}-\frac{2n\bar{\mathbf{r}}}{\lambda+n-\omega}\right)\sum_{i=1}^{n} r_i\hat{x}_i+\frac{\omega^2}{\lambda+\omega}+\frac{n^2\bar{\mathbf{r}}^2}{\lambda+n-\omega}.
\end{aligned}$$

Since ML decoding is a minimization process of (4.31), we may ignore irrelevant terms and delete scaling constants. Note that ω equals the number of ones in $\hat{\mathbf{x}}$ and thus, it does not depend on the specific positions of ones in $\hat{\mathbf{x}}$. The only degree of freedom the decoder has for minimizing (4.31) is permuting the symbols in $\hat{\mathbf{x}}$ to maximize $d(\mathbf{r},\hat{\mathbf{x}})$. Hence, for the value of ω under consideration, the ML decoding result is

$$\mathbf{x}_{o,\omega} = \arg\max_{\hat{\mathbf{x}}\in S_\omega} d(\mathbf{r},\hat{\mathbf{x}}).$$

Note from (4.32) that $d(\mathbf{r},\hat{\mathbf{x}})$ can be regarded as a quadratic function of $\sum_{i=1}^{n} r_i\hat{x}_i$. Since the graph of the quadratic function is a convex parabola, the maximum value of (4.32) occurs when $\sum_{i=1}^{n} r_i\hat{x}_i$ is minimal or maximal among all $\hat{\mathbf{x}}\in S_\omega$. However, which of these two options leads to the maximum value is not apparent from the expression. Therefore, both the maximum and the minimum values of $\sum_{i=1}^{n} r_i\hat{x}_i$ are considered. Slepian [37] showed that the value of $\sum_{i=1}^{n} r_i\hat{x}_i$ can be maximized by matching the largest symbol of $\mathbf{r}$ with the largest symbol of $\hat{\mathbf{x}}$, the second largest symbol of $\mathbf{r}$ with the second largest symbol of $\hat{\mathbf{x}}$, etc. On the other hand, the value of $\sum_{i=1}^{n} r_i\hat{x}_i$ can be minimized by matching the largest symbol of $\mathbf{r}$ with the smallest symbol of $\hat{\mathbf{x}}$, the second largest symbol of $\mathbf{r}$ with the second smallest symbol of $\hat{\mathbf{x}}$, etc. Hence, only two codewords from S_ω under consideration need to be evaluated. The n symbols of the received word, $\mathbf{r}$, are sorted, largest to smallest, in the same way as taught in Slepian's prior art. For every $\omega\in V$, decide $\mathbf{x}_{o,\omega}$ by maximizing the values of $d(\mathbf{r},\hat{\mathbf{x}}\in S_\omega)$ over two candidate codewords, $(\underbrace{1,\ldots,1}_{\omega},0,\ldots,0)$ and $(0,\ldots,0,\underbrace{1,\ldots,1}_{\omega})$ for a fixed ω.

For the complete codebook S, ML decoding of the received vector, $\mathbf{r}$, can thus be accomplished by finding $\mathbf{x}_{o,\omega}$ as described above for all $\omega\in V$, and then minimizing $\delta_{\mathrm{ML}}(\mathbf{r},\mathbf{x}_{o,\omega})$ over the $|V|$ candidates, i.e.,

$$\mathbf{x}_o = \arg\min_{\omega\in V} \delta_{ML}(\mathbf{r},\mathbf{x}_{o,\omega}).$$

The number of codewords to be evaluated in minimizing (4.30) is $|S|$, which is impractical for larger S, since it tends to grow exponentially with n. By using the method presented in this section, the number of words that need to be evaluated is reduced to only $2|V|$, i.e., twice the number of constant composition codes that constitute the codebook S. The index set V is of size at most $n+1$. Thus, the number of evaluations grows at most linearly with n. For large codebooks, this is a significant reduction compared to an exhaustive search among all the candidate codewords. It should be mentioned that in order to apply this method, the received vector needs to be sorted, and the resulting ML codeword needs to be inversely permuted accordingly. However, sorting is well known to have only moderate computational complexity in terms of the length of the vector n, e.g., $O(n \log n)$ symbol swaps. We can thus conclude that the overall complexity is still considerably lower than for an exhaustive search.

4.3.3. Performance Evaluation

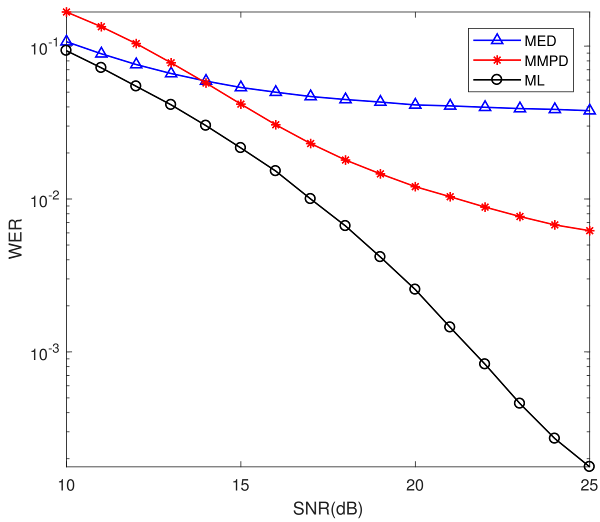


Figure 4.3: Simulated word error rate (WER) versus signal to noise ratio (SNR) of S^* for channels with Gaussian noise and signal dependent offsets, that have standard deviations $\beta_0 = 0.2$, $\beta_1 = 0.3$, and correlation $\rho = 0.75$.

In this section, we investigate the word error rate (WER) performance of three decoders, namely, Euclidean distance based decoding (2.16) (MED), modified Pearson distance based decoding (2.39) (MMPD), and ML decoding (4.20). Simulated WER results are shown for the (3,2) parity check code S^*. This simple codebook suffices to demonstrate some important WER characteristics of the proposed method (4.20) in comparison with the traditional methods (2.16) and (2.39).

WER Versus Signal to Noise Ratio (SNR)

Figures 4.3 and 4.4 show the word error rate results of simulations for a range of noise levels. In both Figs. 4.3 and 4.4, each point was the result of 10,000 trials. The signal de-

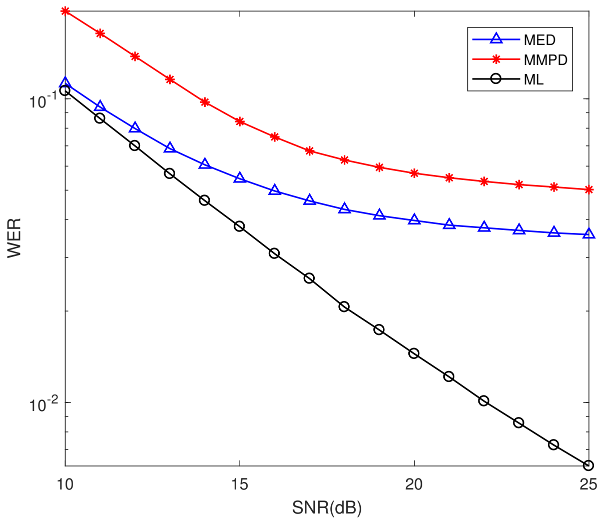


Figure 4.4: Simulated word error rate (WER) versus signal to noise ratio (SNR) of S^* for channels with Gaussian noise and signal dependent offsets, that have standard deviations $\beta_0 = 0.2$, $\beta_1 = 0.3$, and correlation $\rho = 0.15$.

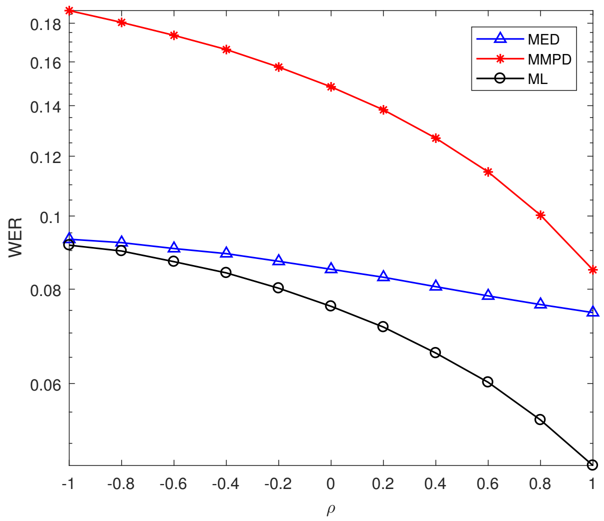


Figure 4.5: Simulated word error rate (WER) versus ρ of S^* for channels with Gaussian noise and signal dependent offsets, that standard deviations $\beta_0 = 0.2$, $\beta_1 = 0.3$, and signal to noise ratio is 12 dB.

pendent offsets b_0 and b_1 have zero means and standard deviations $\beta_0 = 0.2$ and $\beta_1 = 0.3$, respectively. The correlation ρ between b_0 and b_1 is set to be 0.75 in Fig. 4.3 and 0.15 in Fig. 4.4. It can be observed from these figures that the performance of Euclidean distance

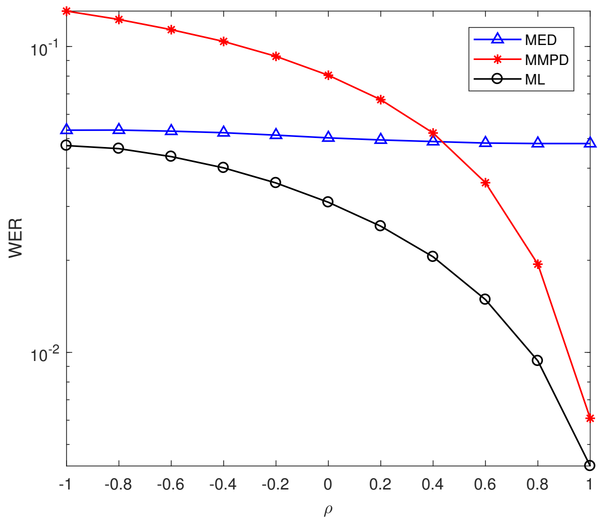


Figure 4.6: Simulated word error rate (WER) versus ρ of S^* for channels with Gaussian noise and signal dependent offsets, that have standard deviations $\beta_0 = 0.2$, $\beta_1 = 0.3$, and signal to noise ratio is 17 dB.

based decoding is close to ML decoding when the value of SNR is small, for both $\rho = 0.75$ and $\rho = 0.15$. For high correlation, Pearson distance based decoding outperforms Euclidean distance based decoding at large SNR values, see Fig. 4.3 for $\rho = 0.75$. However, for low correlation, the performance of Pearson distance based decoding is worse than that of Euclidean distance based decoding, as illustrated in Fig. 4.4 for $\rho = 0.15$. ML decoding is always better than both of them, as expected. By comparing the two figures, it can be seen that the offset correlation, ρ, plays a crucial role in the WER performances. The following subsection is concerned with WER versus ρ.

WER VERSUS CORRELATION BETWEEN OFFSETS

The simulation results for a range of ρ values are shown in Figures 4.5 and 4.6. Each point was the result of 10,000 trials, in the situation that the signal dependent offsets b_0 and b_1 are still assumed to have zero means and standard deviations $\beta_0 = 0.2$ and $\beta_1 = 0.3$, respectively. Here, SNR is set to be 12 dB in Fig. 4.5 and 17 dB in Fig. 4.6.

WERs of all three decoding criteria decrease when ρ increases from -1 to 1. Note that the value of ρ has only little effect on the performance of Euclidean distance based decoding but that it has a high impact on the results of Pearson distance based decoding and ML decoding, especially when SNR is equal to 17 dB. The WER of ML decoding is always better than that of the other two decoders, as expected. These results are in accordance with our earlier observations, which showed that Pearson distance based decoding outperforms Euclidean distance based decoding when ρ is close to 1 and SNR is large, and that the ML decoding surpasses both of them.

4.4. CONCLUSION

In this chapter, we have studied channels that are not only distorted by Gaussian noise but also by another substantial channel impairment, *signal dependent offset*. We have investigated a maximum likelihood (ML) decoding criterion for the situation that the noise and the offset are uniformly distributed. We have also shown that the ML decoder can achieve a zero WER when the standard deviations of the noise and the offset are small enough. Later, an ML decoding criterion has been derived for such channels to improve and strengthen the resilience to Gaussian noise and signal dependent offset. We have shown that a previous result on ML decoding in the case that the offsets are identical, i.e., signal independent, appears as a special case of our proposed criterion. For codebooks consisting of the union of constant weight sets, it has been shown that significant complexity reductions can be obtained.

5

NOISY CHANNELS WITH GAIN AND OFFSET MISMATCH

We have considered decoding techniques for noisy channels with offset mismatch previously. In this chapter, we put our attention on a more complicated channel model in which the retrieved data is corrupted by Gaussian noise, gain, and offset mismatch. The intervals from which the gain and offset values are taken are known, but there are no further assumptions on the distributions on these intervals. We derive maximum likelihood (ML) decoding methods for such channels based on finding a codeword with the closest Euclidean distance to a specified set defined by the received vector and the gain and offset parameters. We provide geometric interpretations of gain and offset and show that certain known criteria appear as special cases of our general setting.

5.1. INTRODUCTION

We use the same channel model as mentioned in [49]: besides the noise, which varies from symbol to symbol, a multiplicative factor a and an additive term b specify the gain and offset mismatch, respectively, which are assumed to be constant within one codeword length but may be different for the next one. Specifically, consider transmitting a codeword $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ from a codebook S over the q-ary alphabet $[q]$, $q \geq 2$, where n is a positive integer. The transmitted symbols x_i are distorted by additive noise v_i, by a factor $a > 0$, called gain/scaling, and by an additive term b, called offset, i.e., the received symbols r_i read

$$r_i = a(x_i + v_i) + b,$$

The material in this chapter has appeared in

- R. Bu and J. H. Weber, "Maximum Likelihood Decoding for Multi-Level Cell Memories with Scaling and Offset Mismatch", *Proc. of IEEE International Conference on Communications (ICC)*, Shanghai, China, May, 2019.

for $i = 1, \ldots, n$. The parameters $v_i \in \mathbb{R}$ are zero-mean i.i.d. Gaussian noise samples with variance of $\sigma^2 \in \mathbb{R}$, that is, the noise vector $\mathbf{v}$ has multivariant Gaussian distribution $\chi(\mathbf{v})$ as Equation (2.15). The gain and offset (unknown to both the sender and the receiver) may slowly vary over time due to various factors in multi-level cells. So we assume they may differ from codeword to codeword, but do not vary within a codeword. The received vector when a codeword $\mathbf{x}$ is transmitted is

$$\mathbf{r} = a(\mathbf{x} + \mathbf{v}) + b\mathbf{1}, \tag{5.1}$$

where $\mathbf{1} = (1, 1, \ldots, 1)$ is the real all-one vector of length n.

There are many examples of channels with offset and gain mismatch. Reading errors in Flash memories may originate from cell drift in aging devices [11]. In the digital optical recording, fingerprints and scratches on the surface of discs result in offset variations of the retrieved signal [65]. For direct conversion receivers, the local oscillator is the primary source of dc-offset [78]. In wireless communication channels, path loss will result in unknown gain, fading, of the channel [79]. The offset may arise as baseline wander in the low frequencies of baseband transmission channels [80].

5

The well-known MED will decode the received vector $\mathbf{r}$ and output a codeword that minimizing the (squared) Euclidean distance (2.16). Applying (5.1) to the (squared) Euclidean distance gives

$$\begin{aligned}\delta_E(\mathbf{r}, \hat{\mathbf{x}}) &= \sum_{i=1}^{n} [a(x_i + v_i) + b - \hat{x}_i]^2 \\ &= -2\sum_{i=1}^{n} x'_i \hat{x}_i + \sum_{i=1}^{n} (x'_i + b)^2 - 2b \sum_{i=1}^{n} \hat{x}_i + \sum_{i=1}^{n} \hat{x}_i^2,\end{aligned} \tag{5.2}$$

where $x'_i = a(x_i + v_i)$. Gain and offset mismatch have a significant bearing on the error performance of MED as $\hat{\mathbf{x}}$ related terms are dependent on a and b. In the prior art, constrained codes, specifically, $dc/dc^2-balanced$ codes, are considered to counter the effects of gain and offset mismatch [36]. By definition, all codewords $\mathbf{x}$ in a dc/dc^2-balanced code satisfy that the symbol sum

$$\textstyle\sum_{i=1}^{n} x_i = a_1$$

and symbol energy

$$\textstyle\sum_{i=1}^{n} x_i^2 = a_2,$$

are prescribed, where a_1 and a_2 are two positive integers selected by the code designer. It is clear that in the case where all codewords satisfy the symbol sum and energy constraints, the (squared) Euclidean distance (2.16) is equivalent to

$$\delta_E(\mathbf{r}, \hat{\mathbf{x}}) \equiv -\textstyle\sum_{i=1}^{n} (x_i + v_i)\hat{x}_i.$$

Thus the MED of dc/dc^2-balanced codes is immune to channel mismatch. The notion of dynamic thresholds based on balanced codes is introduced in [33] for the reading of binary sequences. The generalization of applying dynamic thresholds for multi-level cell memories is explored in [35]. The generating function is used to enumerating the dc/dc^2-balanced codes in [45]. Encoding/decoding of balanced codes has attracted a considerable amount of research and engineering attention [46, 47].

Immink and Weber [49] advocate the use of Pearson distance based decoding instead of traditional Euclidean distance based decoding in situations that require resistance towards gain and/or offset mismatch. Section 2.4 has introduced the minimum Pearson distance based detection in detail. Since the offset b changes the mean of a vector, it seems reasonable to consider normalized vectors $\hat{\mathbf{x}} - \bar{\hat{\mathbf{x}}}\mathbf{1}$ and $\mathbf{r} - \bar{\mathbf{r}}\mathbf{1}$ rather than $\hat{\mathbf{x}}$ and $\mathbf{r}$. On the other hand, scaling a vector of mean zero by a only changes its standard deviation by a factor of a. So it seems reasonable to scale the normalized vectors so that they have a standard deviation of 1. It is not difficult to see that this is the Pearson correlation coefficient $\rho_{\mathbf{r},\hat{\mathbf{x}}}$.

This chapter aims to generalize maximum likelihood (ML) decoding for channels with Gaussian noise and gain and offset mismatch. The main contribution of this chapter is two-fold. Firstly, we derive an ML decoding criterion for channels with Gaussian noise and also suffering from the gain a and the offset b, which are known to be within specific ranges, specifically $0 < a_1 \le a \le a_2$ and $b_1 \le b \le b_2$. The ML decoding criterion will also be illustrated with geometric interpretations. Secondly, the proposed ML criterion provides a general framework, including the gain-only case and the offset-only case. Some known criteria [62] [61] are shown to be special cases of this framework for particular a_1, a_2, b_1, and b_2 settings.

This chapter is organized as follows. In Section 5.2, we start by providing how to achieve ML decoding for this channel. We continue in Section 5.3 considering several special cases, which relate to known results in this area. We wrap up the paper with some comments and ideas for future work in Section 5.5.

5

5.2. MAXIMUM LIKELIHOOD DECODING

A maximum likelihood (ML) decoding will conclude that $\hat{\mathbf{x}}$ is the most likely codeword transmitted if the codeword $\hat{\mathbf{x}}$ maximizes the likelihood $\mathbb{P}(\mathbf{r}|\hat{\mathbf{x}})$, that is, the probability that $\mathbf{r}$ is received, given $\hat{\mathbf{x}}$ is sent. From (5.1), we know $\mathbf{v} = (\mathbf{r} - b\mathbf{1})/a - \hat{\mathbf{x}}$ when a and b are fixed, and since a is nonzero, the likelihood $\mathbb{P}(\mathbf{r}|\hat{\mathbf{x}})$ is

$$\chi((\mathbf{r} - b\mathbf{1})/a - \hat{\mathbf{x}}).$$

Here, we consider the situation that the gain and the offset take their values within specific ranges, specifically $0 < a_1 \le a \le a_2$ and $b_1 \le b \le b_2$, but do not make any further assumptions on the distributions on these intervals. Thus, in order to achieve ML decoding, the criterion to maximize among all candidate codewords $\hat{\mathbf{x}}$ is

$$\max_{0<a_1\le a\le a_2, b_1\le b\le b_2} \chi((\mathbf{r} - b\mathbf{1})/a - \hat{\mathbf{x}}). \tag{5.3}$$

Since the logarithm function is strictly increasing on the positive real numbers and χ is a positive function, an equivalent formulation of the problem is to find $\hat{\mathbf{x}} \in S$ that maximizes

$$\max_{0<a_1\le a\le a_2, b_1\le b\le b_2} \ln \chi((\mathbf{r} - b\mathbf{1})/a - \hat{\mathbf{x}}).$$

Since

$$\ln \chi((\mathbf{r}-b\mathbf{1})/a-\hat{\mathbf{x}}) = -n\ln(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}((r_i-b)/a-\hat{x}_i)^2 \tag{5.4}$$

has a component $-n\ln(\sigma\sqrt{2\pi})$ that is independent of $\hat{\mathbf{x}}$ and $\mathbf{r}$, and since $\frac{1}{2\sigma^2}$ is a positive constant, a maximum likelihood decoder finds a codeword $\hat{\mathbf{x}}$ that minimizes

$$\min_{0<a_1\le a\le a_2, b_1\le b\le b_2} \sum_{i=1}^{n}((r_i-b)/a-\hat{x}_i)^2,$$

i.e., it minimizes the squared Euclidean distance between the candidate codeword $\hat{\mathbf{x}}$ and the points in

$$U = \{(\mathbf{r}-b\mathbf{1})/a | 0 < a_1 \le a \le a_2, b_1 \le b \le b_2\},$$

which is a subset of the subspace

$$U' = \{c\mathbf{r}+d\mathbf{1} | c,d \in \mathbb{R}\}$$

in $\mathbb{R}^n$.

The squared Euclidean distance between a vector $\hat{\mathbf{x}}$ and the set U is defined as

$$\delta_E(U,\hat{\mathbf{x}}) = \sum_{i=1}^{n}(p_i-\hat{x}_i)^2,$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ is the point in U that is closest to $\hat{\mathbf{x}}$. The most likely candidate codeword $\mathbf{x}_o$ for a received vector has the smallest $\delta_E(U,\hat{\mathbf{x}})$, that is

$$\mathbf{x}_o = \arg\min_{\hat{\mathbf{x}}\in S} \delta_E(U,\hat{\mathbf{x}}). \tag{5.5}$$

Hence, $\hat{\mathbf{x}} \in S$ closest to U is chosen as the ML decoder output.

In order to calculate $\delta_E(U,\hat{\mathbf{x}})$ for a codeword $\hat{\mathbf{x}}$, we first find the point in U' that is closest to $\hat{\mathbf{x}}$ and then check if this point is in U. Applying the first derivative test gives that the closest point in U' to $\hat{\mathbf{x}}$ is $\mathbf{p}_0 = c_0\mathbf{r} + d_0\mathbf{1}$ with

$$c_0 = \frac{\langle \mathbf{r},\hat{\mathbf{x}}\rangle - n\bar{\mathbf{r}}\bar{\hat{\mathbf{x}}}}{\langle \mathbf{r},\mathbf{r}\rangle - n\bar{\mathbf{r}}^2}$$

and

$$d_0 = \frac{\langle \mathbf{r},\mathbf{r}\rangle\,\bar{\hat{\mathbf{x}}} - \langle \mathbf{r},\hat{\mathbf{x}}\rangle\,\bar{\mathbf{r}}}{\langle \mathbf{r},\mathbf{r}\rangle - n\bar{\mathbf{r}}^2}.$$

In Figure 5.1, we depict the subset U in gray when $a_1 < 1 < a_2$ and $b_1 < 0 < b_2$. Four vertices **A**, **B**, **C**, **D** are also shown in the picture:

$$\mathbf{A} = (\mathbf{r}-b_1\mathbf{1})/a_1,$$

$$\mathbf{B} = (\mathbf{r}-b_2\mathbf{1})/a_1,$$

$$\mathbf{C} = (\mathbf{r}-b_2\mathbf{1})/a_2,$$

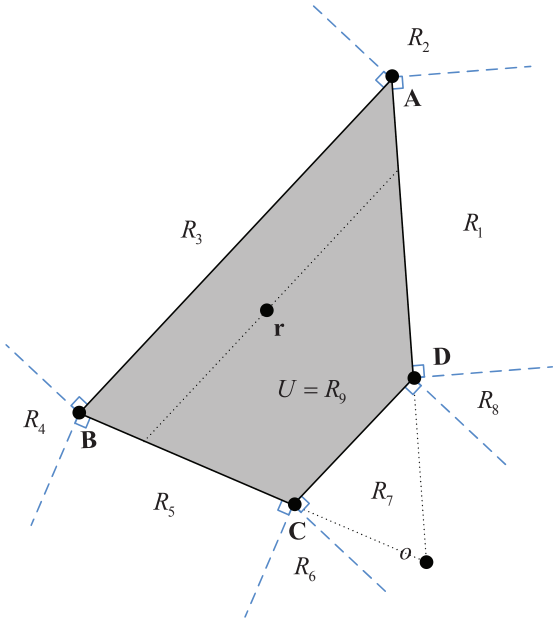


Figure 5.1: Subdivision of $U' = \{c\mathbf{r} + d\mathbf{1} | c, d \in \mathbb{R}\}$.

5

$$\mathbf{D} = (\mathbf{r} - b_1\mathbf{1}) / a_2.$$

Perpendicular lines (blue dash) in U' to sides of U through vertices are pictured in Figure 5.1. These perpendicular lines and the sides of U separate U' into 9 subsets, namely, $R_1, R_2, \ldots, R_9$. For instance, the perpendicular lines to side **BC** and **BC** itself form the boundaries of R_5. We use the notation R_9 in Figure 5.1 for the subset U for clerical convenience.

Theorem 10. *If $\mathbf{p}_0$ is in the subset R_i, $i = 1, \ldots, 9$, then the closest point in U to $\hat{\mathbf{x}}$ is*

$$\mathbf{p} = \begin{cases} \dfrac{\langle \mathbf{r} - b_1\mathbf{1}, \hat{\mathbf{x}} \rangle}{\|\mathbf{r} - b_1\mathbf{1}\|^2}(\mathbf{r} - b_1\mathbf{1}) & \text{if } i = 1, \\ \dfrac{\langle \mathbf{r} - b_2\mathbf{1}, \hat{\mathbf{x}} \rangle}{\|\mathbf{r} - b_2\mathbf{1}\|^2}(\mathbf{r} - b_2\mathbf{1}) & \text{if } i = 5, \\ (\mathbf{r} - (\bar{\mathbf{r}} - a_1\bar{\hat{\mathbf{x}}})\mathbf{1}) / a_1 & \text{if } i = 3, \\ (\mathbf{r} - (\bar{\mathbf{r}} - a_2\bar{\hat{\mathbf{x}}})\mathbf{1}) / a_2 & \text{if } i = 7, \\ \mathbf{A} & \text{if } i = 2, \\ \mathbf{B} & \text{if } i = 4, \\ \mathbf{C} & \text{if } i = 6, \\ \mathbf{D} & \text{if } i = 8, \\ \mathbf{p}_0 & \text{if } i = 9. \end{cases} \tag{5.6}$$

The ML decoding criterion is minimizing $\delta_E(\mathbf{p}, \hat{\mathbf{x}})$ among all candidate codewords.

Proof. If $\mathbf{p}_0$ is in the subset R_1, maximizing (5.3) is equivalent to minimizing the smallest squared Euclidean distance from the codeword $\hat{\mathbf{x}}$ to the line segment

$$\mathbf{AD} = \{(\mathbf{r} - b_1\mathbf{1})/a | 0 < a_1 \le a \le a_2\},$$

which is shown in Figure 5.1. Let θ be the angle between $\hat{\mathbf{x}}$ and $\mathbf{r} - b_1\mathbf{1}$. The point on $\mathbf{AD}$ closest to $\hat{\mathbf{x}}$ is $\mathbf{p} = \alpha(\mathbf{r} - b_1\mathbf{1})$ with

$$\alpha = (\|\hat{\mathbf{x}}\| \cos\theta)/\|\mathbf{r} - b_1\mathbf{1}\| = \langle \mathbf{r} - b_1\mathbf{1}, \hat{\mathbf{x}}\rangle / \|\mathbf{r} - b_1\mathbf{1}\|^2.$$

Similarly, when $\mathbf{p}_0$ is in the subset R_5, the point on $\mathbf{BC} = \{(\mathbf{r} - b_2\mathbf{1})/a | 0 < a_1 \le a \le a_2\}$ closest to $\hat{\mathbf{x}}$ is $\mathbf{p} = \alpha(\mathbf{r} - b_2\mathbf{1})$ with

$$\alpha = \langle \mathbf{r} - b_2\mathbf{1}, \hat{\mathbf{x}}\rangle / \|\mathbf{r} - b_2\mathbf{1}\|^2.$$

If $\mathbf{p}_0$ is in the subset R_3, the point $\mathbf{p} \in U$ that is closest to $\hat{\mathbf{x}}$ must be on the line segment

5

$$\mathbf{AB} = \{(\mathbf{r} - b\mathbf{1})/a_1 | b_1 \le b \le b_2\},$$

which is shown in Figure 5.1. The point on $\mathbf{AB}$ that is closest to $\hat{\mathbf{x}}$ is $\mathbf{p} = (\mathbf{r} - \beta\mathbf{1})/a_1$, with $\beta = \bar{\mathbf{r}} - a_1\bar{\hat{\mathbf{x}}}$, which follows from the first derivative test. The proof is similar when $\mathbf{p}_0$ is in the subset R_7, with the line segment $\mathbf{CD}$ taking the role of $\mathbf{AB}$.

If $\mathbf{p}_0$ is in the subset R_2, then the closest point in U to $\hat{\mathbf{x}}$ is the vertex $\mathbf{A} = (\mathbf{r} - b_1\mathbf{1})/a_1$, as can be observed from Figure 5.1. Similar results are found for the situations that $\mathbf{p}_0$ is in the subset R_4, R_6, and R_8, where the closest point in U to $\hat{\mathbf{x}}$ is $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$, respectively.

Obviously, the closest point in U to $\hat{\mathbf{x}}$ is $\mathbf{p}_0$ itself when $\mathbf{p}_0$ is in the subset $R_9 = U$. □

5.3. Special Cases

Several special values of a_1, a_2, b_1 and b_2 are considered, leading to typical cases for maximizing (5.3); these include the gain-only and offset-only cases. Not only ML decoding criteria are discussed, but also conventional decoding criteria as introduced in Chapter 2.

5.3.1. Gain-Only Case

In the gain-only case, i.e., $b = 0$, we simply have

$$\mathbf{r} = a(\mathbf{x} + \mathbf{v}),$$

where the gain, a, is unknown to both sender and receiver.

In Theorem 2 of [62], the following ML criterion was presented for the case that there is bounded gain ($0 < a_1 \le a \le a_2$) and no offset mismatch ($b = 0$):

$$L_{a_1,a_2}(\mathbf{r}, \hat{\mathbf{x}}) = \begin{cases} \delta_E(\mathbf{r}/a_1, \hat{\mathbf{x}}) & \text{if } \langle \mathbf{r}, \hat{\mathbf{x}}\rangle > \langle \mathbf{r}, \mathbf{r}\rangle / a_1, \\ \delta_E(\mathbf{r}/a_2, \hat{\mathbf{x}}) & \text{if } \langle \mathbf{r}, \hat{\mathbf{x}}\rangle < \langle \mathbf{r}, \mathbf{r}\rangle / a_2, \\ \|\hat{\mathbf{x}}\|^2 - \left(\frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\|\mathbf{r}\|}\right)^2 & \text{otherwise.} \end{cases} \tag{5.7}$$

This result can also be found from the general framework presented in the previous section by setting $b_1 = b_2 = 0$ in Theorem 10. Note that this gives indeed that $\mathbf{p} = \mathbf{r}/a_1$ if $\mathbf{p}_0 \in R_2 \cup R_3 \cup R_4$, which corresponds to the situation that

$$\frac{\|\hat{\mathbf{x}}\| \cos\varphi}{\|\mathbf{r}\|} = \frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\langle \mathbf{r}, \mathbf{r}\rangle} > 1/a_1,$$

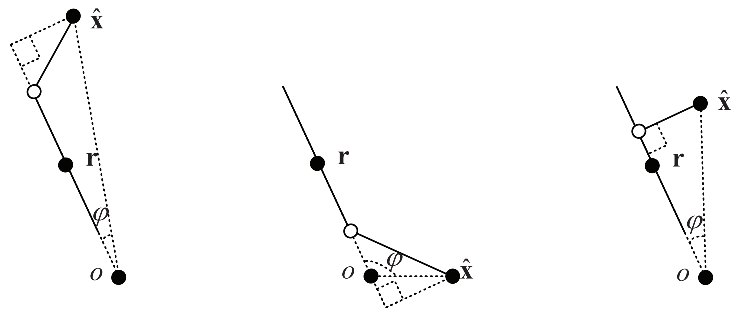
where φ is the angle between $\hat{\mathbf{x}}$ and $\mathbf{r}$. Similarly, note that $\mathbf{p} = \mathbf{r}/a_2$ if $\mathbf{p}_0 \in R_6 \cup R_7 \cup R_8$, which corresponds to the situation that

$$\frac{\|\hat{\mathbf{x}}\| \cos\varphi}{\|\mathbf{r}\|} = \frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\langle \mathbf{r}, \mathbf{r}\rangle} < 1/a_2.$$

Finally, note that $\mathbf{p} = \frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\|\mathbf{r}\|^2}\mathbf{r}$ if $\mathbf{p}_0 \in R_1 \cup R_5 \cup R_9$, which corresponds to the 'otherwise' case in (5.7), and that

$$\delta_E(\mathbf{p}, \hat{\mathbf{x}}) = \delta_E\left(\frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\|\mathbf{r}\|^2}\mathbf{r}, \hat{\mathbf{x}}\right) = \|\hat{\mathbf{x}}\|^2 - \left(\frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\|\mathbf{r}\|}\right)^2.$$

In Figure 5.2, we draw the three cases in (5.7), where the subset $\{\mathbf{r}/a \,|\, 0 < a_1 \le a \le a_2\}$ is a line segment in the direction of $\mathbf{r}$. The circle points are the closest points on this line segment to $\hat{\mathbf{x}}$.



(a) $\langle \mathbf{r}, \hat{\mathbf{x}}\rangle > \langle \mathbf{r}, \mathbf{r}\rangle / a_1$ (b) $\langle \mathbf{r}, \hat{\mathbf{x}}\rangle < \langle \mathbf{r}, \mathbf{r}\rangle / a_2$ (c) otherwise

Figure 5.2: The distance of a candidate codeword $\hat{\mathbf{x}}$ to the subset $\{\mathbf{r}/a \,|\, 0 < a_1 \le a \le a_2\}$: three cases in (5.7) assuming $a_1 < 1 < a_2$.

Next, we consider the situation that $a_1 \to 0$ and $a_2 \to \infty$, i.e., the only knowledge on the gain a is that it is a positive number, without further limitations. The subset $\{\mathbf{r}/a \,|\, a \in \mathbb{R}, a > 0\}$ is a ray from the origin in the direction of $\mathbf{r}$. In this case, it follows from the above that ML decoding can be achieved by minimizing

$$L_a(\mathbf{r}, \hat{\mathbf{x}}) = \begin{cases} \|\hat{\mathbf{x}}\|^2 - \left(\frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\|\mathbf{r}\|}\right)^2 & \text{if } \frac{\langle \mathbf{r}, \hat{\mathbf{x}}\rangle}{\langle \mathbf{r}, \mathbf{r}\rangle} > 0, \\ \|\hat{\mathbf{x}}\|^2 & \text{otherwise.} \end{cases} \tag{5.8}$$

One reason for this choice is that it behaves well with respect to an affine gain function $(a > 0)$, since

$$L_a(\mathbf{r}, \hat{\mathbf{x}}) = L_a(\mathbf{r}/a, \hat{\mathbf{x}}).$$

That is, scaling a vector $\mathbf{r}$ by a does not change the angle φ between $\hat{\mathbf{x}}$ and $\mathbf{r}$.

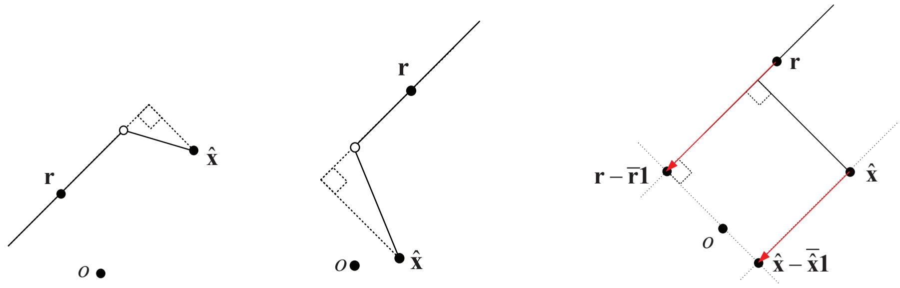


Figure 5.3: The distance of a candidate codeword $\hat{\mathbf{x}}$ to the line segment $\{\mathbf{r}-b\mathbf{1}\,|\,b_1 \le b \le b_2\}$: two cases in (5.9) assuming $b_1 < 0 < b_2$.

5.3.2. OFFSET-ONLY CASE

In the offset-only case, i.e., $a = 1$, we simply have

$$\mathbf{r} = \mathbf{x} + \mathbf{v} + b\mathbf{1},$$

where the offset b is unknown to both sender and receiver.

In Theorem 1 of [62], the following ML criterion was presented for the case that $a = 1$ and $b_1 \le b \le b_2$:

$$L_{b_1,b_2}(\mathbf{r},\hat{\mathbf{x}}) = \begin{cases} \delta_E(\mathbf{r}-b_1\mathbf{1},\hat{\mathbf{x}}) & \text{if } \bar{\mathbf{r}}-\bar{\hat{\mathbf{x}}} < b_1, \\ \delta_E(\mathbf{r}-b_2\mathbf{1},\hat{\mathbf{x}}) & \text{if } \bar{\mathbf{r}}-\bar{\hat{\mathbf{x}}} > b_2, \\ \delta_E(\mathbf{r}-(\bar{\mathbf{r}}-\bar{\hat{\mathbf{x}}})\mathbf{1},\hat{\mathbf{x}}) & \text{otherwise.} \end{cases} \tag{5.9}$$

This result also follows from the general setting presented in the previous section, by substituting $a_1 = a_2 = 1$. Note that the first case in (5.9) corresponds to the situation that $\mathbf{p}_0 \in R_1 \cup R_2 \cup R_8$, the second case to $\mathbf{p}_0 \in R_4 \cup R_5 \cup R_6$, and the last case to $\mathbf{p}_0 \in R_3 \cup R_7 \cup R_9$.

We illustrate the first two situations of $L_{b_1,b_2}(\mathbf{r},\hat{\mathbf{x}})$ in Figure 5.3(a) and 5.3(b), the last one in Figure 5.3(c), where $\{\mathbf{r}-b\mathbf{1}\,|\,b_1 \le b \le b_2\}$ is shown by a line segment passing through $\mathbf{r}$ with direction $\mathbf{1}$. The point in $\{\mathbf{r}-b\mathbf{1}\,|\,b_1 \le b \le b_2\}$ that is closest to $\hat{\mathbf{x}}$ is $\mathbf{r}-b_1\mathbf{1}$ or $\mathbf{r}-b_2\mathbf{1}$ for the situations in Figure 5.3(a) and 5.3(b). For the 'otherwise' case in (5.9), we consider in Figure 5.3(c) the normalized vectors $\hat{\mathbf{x}}-\bar{\hat{\mathbf{x}}}\mathbf{1}$ and $\mathbf{r}-\bar{\mathbf{r}}\mathbf{1}$ rather than $\hat{\mathbf{x}}$ and $\mathbf{r}$.

By letting $b_1 \to -\infty$ and $b_2 \to \infty$, we obtain from (5.9) that the criterion

$$\begin{aligned} L_b(\mathbf{r},\hat{\mathbf{x}}) &= \delta_E(\mathbf{r}-(\bar{\mathbf{r}}-\bar{\hat{\mathbf{x}}})\mathbf{1},\hat{\mathbf{x}}) \\ &= \sum_{i=1}^{n}(r_i - \hat{x}_i + \bar{\hat{\mathbf{x}}})^2 - n\bar{\mathbf{r}}^2 \\ &= \delta'_P(\mathbf{r},\hat{\mathbf{x}}) - n\bar{\mathbf{r}}^2, \end{aligned}$$

when there is no knowledge at all of the magnitude of the offset [62]. Noting that the last term $n\bar{\mathbf{r}}^2$ is irrelevant in the minimization process, we conclude that the modified Pearson criterion $\delta'_P(\mathbf{r},\hat{\mathbf{x}})$ achieves ML decoding in this case.

5.3.3. UNBOUNDED GAIN AND OFFSET CASE

In this subsection, an ML decoding criterion derived by Blackburn [61] for the situation when both the gain a and the offset b are unbounded ($a_1 \to 0$, $a_2 \to \infty$, $b_1 \to -\infty$, $b_2 \to \infty$) is reconsidered as a special case of the results presented in Section 5.2. In [61], Blackburn shows that an ML decoder chooses a codeword $\hat{\mathbf{x}}$ minimizing

$$l_{\mathbf{r}}(\hat{\mathbf{x}}) = \begin{cases} \sigma_{\hat{\mathbf{x}}}^2(1-\rho_{\mathbf{r},\hat{\mathbf{x}}}^2) & \text{when } \rho_{\mathbf{r},\hat{\mathbf{x}}} > 0, \\ \sigma_{\hat{\mathbf{x}}}^2 & \text{otherwise.} \end{cases} \tag{5.10}$$

His argument was that when the gain factor a and the offset term b are fully unknown, except for the sign of a, then maximizing (5.3) is equivalent to minimizing the smallest squared Euclidean distance from the codeword $\hat{\mathbf{x}}$ to the subset

$$U^+ = \{(\mathbf{r}-b\mathbf{1})/a | a, b \in \mathbb{R}, a > 0\},$$

which is a half-subspace of $U' \subset \mathbb{R}^n$. Note that when $a_1 \to 0$, $a_2 \to \infty$, $b_1 \to -\infty$, $b_2 \to \infty$, our U is indeed equal to Blackburn's set U^+. Note that $\mathbf{p}_0 = c_0\mathbf{r} + d_0\mathbf{1}$ is either in $R_9 = U = U^+$ or in R_7. By (2.4), c_0 and d_0 can be rewritten as

$$c_0 = \frac{\rho_{\mathbf{r},\hat{\mathbf{x}}}\sigma_{\hat{\mathbf{x}}}}{\sigma_{\mathbf{r}}} \tag{5.11}$$

and

$$d_0 = \bar{\hat{\mathbf{x}}} - c_0\bar{\mathbf{r}}. \tag{5.12}$$

In case $\mathbf{p}_0 \in R_9$, which happens if and only if $\rho_{\mathbf{r},\hat{\mathbf{x}}} > 0$, then Theorem 10 says $\mathbf{p} = \mathbf{p}_0 = c_0\mathbf{r} + d_0\mathbf{1}$. Note that

$$\begin{aligned}
&\delta_E(c_0\mathbf{r}+d_0\mathbf{1},\hat{\mathbf{x}}) \\
&= \sum_{i=1}^{n} [c_0 r_i + d_0 - \hat{x}_i]^2 \\
&= \sum_{i=1}^{n} \left[c_0(r_i - \bar{\mathbf{r}}) - (\hat{x}_i - \bar{\hat{\mathbf{x}}})\right]^2 \\
&= \sum_{i=1}^{n} \left[c_0^2(r_i-\bar{\mathbf{r}})^2 - 2c_0(r_i-\bar{\mathbf{r}})(\hat{x}_i-\bar{\hat{\mathbf{x}}}) + (\hat{x}_i-\bar{\hat{\mathbf{x}}})^2\right] \\
&= c_0^2\sigma_{\mathbf{r}}^2 - 2c_0\rho_{\mathbf{r},\hat{\mathbf{x}}}\sigma_{\mathbf{r}}\sigma_{\hat{\mathbf{x}}} + \sigma_{\hat{\mathbf{x}}}^2 \\
&= \left(\frac{\rho_{\mathbf{r},\hat{\mathbf{x}}}\sigma_{\hat{\mathbf{x}}}}{\sigma_{\mathbf{r}}}\right)^2\sigma_{\mathbf{r}}^2 - 2\left(\frac{\rho_{\mathbf{r},\hat{\mathbf{x}}}\sigma_{\hat{\mathbf{x}}}}{\sigma_{\mathbf{r}}}\right)\rho_{\mathbf{r},\hat{\mathbf{x}}}\sigma_{\hat{\mathbf{x}}}\sigma_{\mathbf{r}} + \sigma_{\hat{\mathbf{x}}}^2 \\
&= \sigma_{\hat{\mathbf{x}}}^2(1-\rho_{\mathbf{r},\hat{\mathbf{x}}}^2),
\end{aligned}$$

which is indeed the same as in (5.10) when $\rho_{\mathbf{r},\hat{\mathbf{x}}} > 0$.

In case $\mathbf{p}_0 \in R_7$, then Theorem 10 says $\mathbf{p} = \bar{\hat{\mathbf{x}}}\mathbf{1}$ since $a_2 \to \infty$. Hence,

$$\delta_E(\mathbf{p},\hat{\mathbf{x}}) = \delta_E(\bar{\hat{\mathbf{x}}}\mathbf{1},\hat{\mathbf{x}}) = \sigma_{\hat{\mathbf{x}}}^2.$$

This shows that Blackburn's criterion (5.10) indeed appears as a special case of our general setting.

5.4. Simulation Results

Thus far, we have discussed ML decoding for Gaussian noise channels with gain and offset mismatch. We have mentioned that MED is ML decoding for Gaussian noise channels in Section 2.3, while the MPD (2.23) is optimal for channels with gain and offset mismatch due to its intrinsic immunity to both gain and offset mismatch. In other words, the Pearson distance is invariant to changes in translation and scale (up to a sign) in two vectors, that is

$$\delta_P(a + b\mathbf{r}, \hat{\mathbf{x}}) = 1 - \rho_{a+b\mathbf{r},\hat{\mathbf{x}}} = 1 - \rho_{\mathbf{r},\hat{\mathbf{x}}} = \delta_P(\mathbf{r}, \hat{\mathbf{x}}).$$

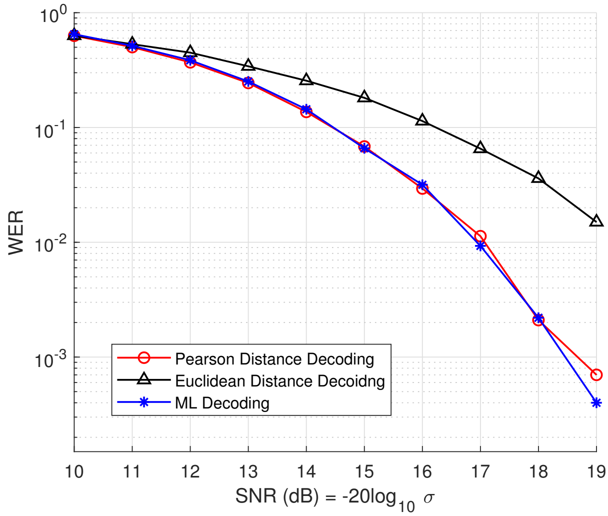


Figure 5.4: Word error rate (WER) against signal-to-noise ratio (SNR) when $q = 4$, $n = 8$, $a = 1.07$, and $b = 0.07$.

Figure 5.4 shows simulation results of MPD, MED, and ML decoding (5.10) when $q = 4$ and $n = 8$. The word error rate (WER) of 10,000 trials is shown as a function of the signal-to-noise ratio (SNR $= -20\log_{10}\sigma$). Results are given for 2-constrained codes [49, 51], while $a = 1.07$ and $b = 0.07$. The simulations indicate that for this case, Pearson distance based decoding has a comparable performance as ML decoding, while Euclidean distance based decoding performs considerably worse.

5.5. Conclusion

This chapter has derived a maximum likelihood decoding criterion for channels with Gaussian noise and gain and/or offset mismatch. The main result has been given in Theorem 10 in Section 5.2. In our channel model, gain and offset are restricted to certain ranges, $0 < a_1 \le a \le a_2$ and $b_1 \le b \le b_2$, which is a generalization of several prior art settings. For instance, by letting $a_1 \to 0$, $a_2 \to \infty$, $b_1 \to -\infty$, $b_2 \to \infty$, we obtain the same ML decoding criterion (5.10) as proposed by Blackburn for the case of unbounded gain and offset. We also provided geometric interpretations illustrating the main ideas.

Gain and offset mismatch are essential issues in many systems, but not the only ones. One could try to derive ML decoding criteria for the channel model in future work, including dependent noise or inter-symbol interference.

6

Noisy Channels with Slowly Varying Offset Mismatch

In this chapter, we consider noisy data transmission channels with unknown gain and varying offset mismatch. Minimum Pearson distance detection is used in cooperation with a difference operator, which offers immunity to such mismatch. Pair-constrained codes are proposed for unambiguous decoding, where, in each codeword, certain adjacent symbol pairs must appear at least once. We investigate the cardinality and redundancy of these codes. A simple systematic encoding algorithm of pair-constrained codes is proposed, whose redundancy is analyzed for memoryless uniform sources. Lastly, we discuss the error performance of the proposed scheme and provide options for future research.

6.1. Introduction

One can observe unknown gain and varying offset mismatch in many applications. Flash memory is an example. In Flash memory, physical features like the device temperature will result in rapid gain and offset variations of the retrieved signal. Memory cells closer to hotter areas on the chip may lose their charge faster than cells closer to colder areas. As a result, the retrieved signal is imposed on an unknown-time varying drift [81]. Another example is a fading and multi-path reception. The received power level may vary rapidly, which results in a time-varying fading or dynamic dc-offset [82].

We consider noisy channels with unknown gain and varying offset mismatch. Assume transmitting a codeword $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ from a codebook $S \subseteq [q]^n$. The received vector $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ is given by

$$\mathbf{r} = a(\mathbf{x} + \mathbf{v}) + b\mathbf{1} + c\mathbf{s}, \tag{6.1}$$

The material in this chapter has appeared in

- R. Bu and J. H. Weber, "Minimum Pearson Distance Detection Using a Difference Operator in the Presence of Unknown Varying Offset", *IEEE Communication Letters*, vol. 23, pp. 1115-1118, July, 2019.

where $\mathbf{1} = (1,1,\ldots,1)$ and $\mathbf{s} = (1,2,\ldots,n)$. The basic premises are that $\mathbf{x}$ is suffering from (i) additive Gaussian noise $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, where $v_i \in \mathbb{R}$ are i.i.d. noise samples with normal distribution $\mathcal{N}(0,\sigma^2)$, where $\sigma^2 \in \mathbb{R}$ denotes the noise variance, (ii) an unknown (positive) gain factor a, where $a \in \mathbb{R}$, $a > 0$, and (iii) an unknown varying offset, $b\mathbf{1} + c\mathbf{s}$, where $b, c \in \mathbb{R}$.

MPD detection [49] is intrinsically resistant to the gain a and offset b, where a and b may change from word to word but are constant for all transmitted symbols within a codeword. Here, we consider the situation in which the offset varies linearly within a codeword, where the slope of the offset, represented by the parameter c, is unknown. A detection scheme for channels with gain and such varying offset is investigated in [59], where, for the binary case, MPD detection is used in conjunction with mass-centered codewords, in such a way that the system is insensitive to both gain and varying offset, i.e., it is (a,b,c)-immune. However, this scheme is very expensive in terms of redundancy.

We show that the combination of MPD and a difference operator is (a,b,c)-immune as well. In addition, pair-constrained codes, where in each codeword certain adjacent symbol pairs must appear at least once, are proposed to achieve unambiguous decoding. The redundancy of pair-constrained codes is much lower than that of prior art mass-centered codes, which makes the new decoding scheme an attractive alternative for practical applications.

6

We start in Section 6.2 with a brief description of the prior art. Section 6.3 presents the backbone of the paper, where it is shown how an MPD detector can be used together with the difference operator. In Section 6.4, we introduce pair-constrained codes, and we compute their cardinality and redundancy. A systematic encoding algorithm is presented. The word error rate (WER) of the new scheme in the presence of additive noise is shown in Section 6.5. In Section 6.6, we discuss the results of this chapter and provide options for future research.

6.2. PRIOR ART

Below we discuss two prior art detection schemes and relevant properties.

The received vector $\mathbf{r}$ can be decoded with the well-known MED (2.16), which outputs the codeword

$$\mathbf{x}_o = \arg\min_{\hat{\mathbf{x}} \in S} \delta_E(\mathbf{r}, \hat{\mathbf{x}}). \tag{6.2}$$

Applying (6.1) to the (squared) Euclidean distance gives

$$\begin{aligned}\delta_E(\mathbf{r},\hat{\mathbf{x}}) &= \sum_{i=1}^{n} [a(x_i + v_i) + b + ci - \hat{x}_i]^2 \\ &= \sum_{i=1}^{n} (x_i' - \hat{x}_i)^2 + \sum_{i=1}^{n} (b+ci)^2 + 2b\sum_{i=1}^{n} x_i' + 2c\sum_{i=1}^{n} i x_i' - 2b\sum_{i=1}^{n} \hat{x}_i - 2c\sum_{i=1}^{n} i\hat{x}_i,\end{aligned} \tag{6.3}$$

where $x_i' = a(x_i + v_i)$. The mismatch may significantly affect the evaluation of δ_E, which can lead to a loss in noise margin or even a loss of the whole codeword. In the prior art, constrained coding techniques have been sought to offer a cure to the reported loss of performance caused by channel mismatch. For example, in [44], second-order spectral-null codes are proposed for the binary case, where x_i takes value from the alphabet $\{0,1\}$. By definition, all codewords $\mathbf{x}$ in a second-order spectral-null code have the property that the symbol sum

$$\sum_{i=1}^{n} x_i = \frac{n}{2},$$

and

$$\sum_{i=1}^{n} i x_i = \frac{n(n+1)}{4}.$$

After substituting the above conditions into (6.3), we see that it does not matter for the outcome of the detection process (6.2) or detection performance if we scale or translate the measurement with a constant independent of the variable $\hat{\mathbf{x}}$. Thus MED of second-order spectral-null codes is intrinsically resistant to channel mismatch as it is independent of the parameters a, b, and c. In that case, the detector is said to be (a,b,c)-immune.

Let $N_{sn^2}(n)$ denote the number of second-order spectral-null code codewords. It has been found in [44] that $N_{sn^2}(n) = 0$ if $n \mod 4 \neq 0$, and for asymptotically large n the number of second-order spectral-null codewords equals

$$N_{sn^2}(n) \simeq \frac{4\sqrt{3}}{\pi}\frac{2^n}{n^2},\ n \mod 4 = 0. \tag{6.4}$$

The redundancy, $r_{sn^2}(n)$, of second-order spectral-null codes is approximately

$$r_{sn^2}(n) = n - \log_2 N_{sn^2}(n) \simeq 2\log_2 n - 1.141, n \gg 1. \tag{6.5}$$

Another example of constrained code techniques is advocated in [59] with a less redundant option that also guarantees (a,b,c)-immunity. The Pearson distance offers immunity to gain and non-varying offset mismatch [49], and an MPD decoder chooses among all candidate codewords $\hat{\mathbf{x}} \in S$ the codeword $\mathbf{x}_o$ whose Pearson distance to the received vector $\mathbf{r}$ is smallest. However, if c is known to be zero in (6.1), using the MPD only makes the error performance (a,b)-immune [49].

In case of varying offset, mass-centered codes in combination with the MPD detector are advocated in [59] for the binary case, where the codebook $S_m \subseteq [2]^n$ is chosen such that each codeword $\mathbf{x} \in S_m$ satisfies

$$\sum_{i=1}^{n} \left(i - \frac{n+1}{2}\right) x_i = 0. \tag{6.6}$$

An example of mass-centered codes with length of $n = 5$ is given in Table 6.1.

We show that with the employment of mass-centered codes, the error performance of the MPD detector is insensitive to the parameter c, that is, (c)-immune. In the process of minimization $\delta_P(\mathbf{r},\hat{\mathbf{x}})$ the terms independent of $\hat{\mathbf{x}}$ is irrelevant. Applying (6.1) to the

Table 6.1: Mass-centered codes, $n = 5$.

0 0 0 0 0
0 1 0 1 0
1 0 0 0 1
1 1 0 1 1
0 0 1 0 0
0 1 1 1 0
1 0 1 0 1
1 1 1 1 1

Pearson distance measure gives

$$\begin{aligned}\delta_P(\mathbf{r},\hat{\mathbf{x}}) &= 1 - \frac{1}{\sigma_{\mathbf{r}}\sigma_{\hat{\mathbf{x}}}}\sum_{i=1}^{n}[a(x_i+\nu_i)+b+ci-\bar{\mathbf{r}}](\hat{x}_i-\bar{\hat{\mathbf{x}}}) \\ &= 1 - \frac{1}{\sigma_{\mathbf{r}}\sigma_{\hat{\mathbf{x}}}}\left[a\sum_{i=1}^{n}(x_i+\nu_i)(\hat{x}_i-\bar{\hat{\mathbf{x}}})+(b-\bar{\mathbf{r}})\sum_{i=1}^{n}(\hat{x}_i-\bar{\hat{\mathbf{x}}})+c\sum_{i=1}^{n}i(\hat{x}_i-\bar{\hat{\mathbf{x}}})\right] \\ &= 1 - \frac{a}{\sigma_{\mathbf{r}}\sigma_{\hat{\mathbf{x}}}}\sum_{i=1}^{n}(x_i+\nu_i)(\hat{x}_i-\bar{\hat{\mathbf{x}}}). \end{aligned} \tag{6.7}$$

where the last equation follows that: i) the '$b-\bar{\mathbf{r}}$'-dependent term is zero since

$$\sum_{i=1}^{n}(\hat{x}_i-\bar{\hat{\mathbf{x}}}) = \sum_{i=1}^{n}\hat{x}_i - n\bar{\hat{\mathbf{x}}} = 0;$$

and ii) the 'c'-dependent term is zero as well since by definition (6.6), we have

$$\sum_{i=1}^{n} i(\hat{x}_i-\bar{\hat{\mathbf{x}}}) = \sum_{i=1}^{n} i\hat{x}_i - \frac{n+1}{2}n\bar{\hat{\mathbf{x}}} = 0, \forall\hat{\mathbf{x}}\in S_m.$$

The remaining term of (6.7) depends only on a gain factor a, but it is irrelevant as it has no effect on the final choice of $\mathbf{x}_o$ that minimizes $\delta_P(\mathbf{r},\hat{\mathbf{x}})$. We conclude that the error performance of an MPD used in conjunction with the codebook S_m of the center of mass constrained codewords is (a,b,c)-immune.

Denote the cardinality of the binary mass-centered codes of length n as $N_o(n)$. For asymptotically large n, the redundancy of S_m is [59]

$$r_o(n) = n - \log_2 N_o(n) \approx \frac{3}{2}\log_2 n + \alpha, \tag{6.8}$$

where $\alpha = \log_2\sqrt{\pi/24} \approx -1.467$ for n odd and $\alpha \approx -0.467$ for n even. The redundancy of mass-centered codes is smaller than that of second-order spectral-null codes.

The two prior art detection methods discussed above have drawbacks in their high redundancy, and to alleviate the drawback, various alternatives are sought. In the next section, we propose and investigate a novel detection method with much less redundancy that also guarantees (a,b,c)-immunity.

6.3. MPD DETECTION USING A DIFFERENCE OPERATOR

Define the difference operator of a vector $\mathbf{u} \in \mathbb{R}^n$ as

$$\Delta\mathbf{u} = \mathbf{u}_{2,n} - \mathbf{u}_{1,n-1}, \tag{6.9}$$

where $\mathbf{u}_{i,j} = (u_i, u_{i+1}, \ldots, u_j)$ for all $1 \le i \le j \le n$.

For any codeword $\mathbf{x} \in S$, we call $\Delta\mathbf{x}$ its difference codeword. The difference codebook, ΔS, is defined by $\Delta S = \{\Delta\mathbf{x} | \mathbf{x} \in S\}$. This is a set of codewords of length $n-1$ over the alphabet $Q = \{-(q-1), \ldots, -1, 0, 1, \ldots, q-1\}$. We will later show that as long as the codebook S satisfies the pair constraints, the difference codebook ΔS can be one-to-one generated from S.

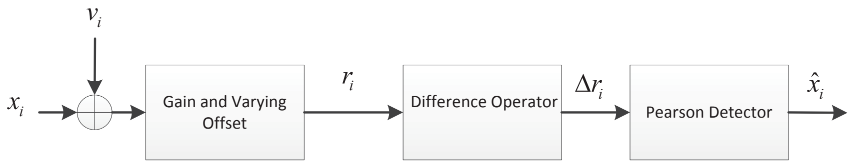


Figure 6.1: MPD detection using a difference operator.

We now show that the use of the difference operator will make Pearson distance based detection (a, b, c)-immune, which is pictured in Figure 6.1. Upon receipt of a vector $\mathbf{r}$, we find the difference vector $\Delta\mathbf{r}$ and then the MPD detector chooses the member in ΔS which has the smallest Pearson distance to $\Delta\mathbf{r}$, i.e.,

$$\Delta\mathbf{x}_o = \underset{\Delta\hat{\mathbf{x}} \in \Delta S}{\arg\min}\, L_p(\Delta\mathbf{r}, \Delta\hat{\mathbf{x}}), \tag{6.10}$$

Note that applying the difference operator (6.9) on the received vector gives

$$\begin{aligned} \Delta\mathbf{r} = &\; \mathbf{r}_{2,n} - \mathbf{r}_{1,n-1} \\ = &\; a(\mathbf{x}_{2,n} + \mathbf{v}_{2,n}) + b\mathbf{1} + c\mathbf{s}_{2,n} \\ &\; -(a(\mathbf{x}_{1,n-1} + \mathbf{v}_{1,n-1}) + b\mathbf{1} + c\mathbf{s}_{1,n-1}) \\ = &\; a(\mathbf{x}_{2,n} - \mathbf{x}_{1,n-1} + \mathbf{v}_{2,n} - \mathbf{v}_{1,n-1}) + c\mathbf{1} \\ = &\; a(\Delta\mathbf{x} + \Delta\mathbf{v}) + c\mathbf{1}, \end{aligned} \tag{6.11}$$

where each entry in $\Delta\mathbf{v}$ has the normal distribution $\mathcal{N}(0, 2\sigma^2)$. In the process of difference operator (6.11) the (b, c)-relevant offset is reduced to only the c-relevant offset. With c in the role of b, we can thus conclude that MPD detection in combination with the difference operator provides (a, b, c)-immunity.

As investigated in [49] and [51] for the case of $(a, b, 0)$-immunity, the codebook should satisfy certain properties in order to allow the use of MPD detection and to prevent ambiguous decoding options. For the case of (a, b, c)-immunity, a new class of codes with the required properties will be presented in the next section.

6.4. PAIR-CONSTRAINED CODES

In order to work well with an MPD detector, the codebook should satisfy the following two requirements [49, 51]: (i) it should not contain vectors $\mathbf{u}$ with $\sigma_{\mathbf{u}} = 0$, since it follows from Equation (2.4) that the Pearson distance is undefined for such $\mathbf{u}$; (ii) the presence of a vector $\mathbf{w}$ in the codebook implies that all vectors $c_1\mathbf{w}+c_2\mathbf{1}$ with $c_1>0$, $c_2\in\mathbb{R}$, and $(c_1,c_2)\neq(1,0)$, should not appear in the codebook because of Equation (2.25). In our case, these requirements must hold for ΔS since the MPD detector operates on the difference codebook. Furthermore, we have the apparent additional requirement that (iii) the codebook should be designed in such a way that the difference operator is a one-to-one map from S to ΔS. In conclusion, we have the following three properties to be satisfied.

- *Property 1*: $k\mathbf{1}\notin\Delta S$ for all $k\in\mathbb{R}$.
- *Property 2*: If $\Delta\mathbf{x}\in\Delta S$, then $c_1\Delta\mathbf{x}+c_2\mathbf{1}\notin\Delta S$ for all $c_1,c_2\in\mathbb{R}$ with $(c_1,c_2)\neq(1,0)$ and $c_1>0$.
- *Property 3*: $\Delta: S\to\Delta S$ is a bijection.

We propose a code satisfying these properties. Pair-constrained codes consist of q-ary n-length codewords, where one or more reference adjacent symbol pairs (s,t), $s,t\in[q]$, must appear at least once, i.e., for each codeword $\mathbf{w}$ there is an i, $1\leq i\leq n-1$ such that $w_i=s$ and $w_{i+1}=t$. In this work, we use a specific set of pair-constrained codes denoted by S_{pc}. The set S_{pc} contains all the vectors where both the adjacent symbol pairs $(0,q-1)$ and $(q-1,0)$ appear at least once, i.e., for each codeword $\mathbf{w}$ there are i and j, $1\leq i,j\leq n-1$ such that $w_i=0$, $w_{i+1}=q-1$, $w_j=q-1$, and $w_{j+1}=0$. This ensures that both the symbols '$q-1$' and '$-(q-1)$' appear at least once in each vector in ΔS_{pc}. This observation is key in showing that the proposed code satisfies the three properties mentioned above, which we will do next.

An example of pair-constrained codes and its difference codebook is given in Table 6.2. Binary codewords of length 5 are considered, where both the adjacent symbol pairs $(0,1)$ and $(1,0)$ appear at least once. Then both symbols 1 and -1 appear at least once in their corresponding difference codewords.

Proof. Property 1 follows immediately from the fact that each word in ΔS_{pc} contains the symbols '$q-1$' and '$-(q-1)$'.

Property 2 follows by a similar argument as used in [49] for so-called T-constrained codes, which has been introduced in Section 2.4.1. We adapt the arguments here to our setting for completeness. Suppose that both $\Delta\mathbf{x}\in\Delta S_{pc}$ and $c_1\Delta\mathbf{x}+c_2\mathbf{1}\in\Delta S_{pc}$ for some c_1 and c_2 as indicated in the property statement. Since $c_1>0$, the fact that both vectors contain the maximum symbol value '$q-1$' implies that $c_1(q-1)+c_2=q-1$, while the fact that both vectors contain the minimum symbol value '$-(q-1)$' implies that $-c_1(q-1)+c_2=-q+1$. Solving these two equations, we find $c_1=1$ and $c_2=0$ as the unique solution, which gives a contradiction and thus shows the result.

Property 3 easily follows by observing that for any $\mathbf{u},\mathbf{w}$ in any code S it holds that $\Delta\mathbf{u}=\Delta\mathbf{w}\iff\mathbf{u}=\mathbf{w}+k\mathbf{1}$ for some $k\in\mathbb{R}$. In case $S=S_{pc}$, the fact that $\mathbf{w}$ contains the minimum symbol value '0' implies that if $k>0$ then $u_i=w_i+k>0\ \forall i$, and if $k<0$ then there exists a position j such that $w_j=0$ and $u_j=w_j+k<0$. These observations contradict that $\mathbf{u}\in S_{pc}$, which implies $k=0$ and thus shows the bijective property for S_{pc}. □

Table 6.2: Pair-constrained codes and its difference codebook, $n = 5$.

0 0 0 1 0	0 0 1 -1
0 0 1 0 0	0 1 -1 0
0 0 1 0 1	0 1 -1 1
0 0 1 1 0	0 1 0 -1
0 1 0 0 0	1 -1 0 0
0 1 0 0 1	1 -1 0 1
0 1 0 1 0	1 -1 1 -1
0 1 0 1 1	1 -1 1 0
0 1 1 0 0	1 0 -1 0
0 1 1 0 1	1 0 -1 1
0 1 1 1 0	1 0 0 -1
1 0 0 0 1	-1 0 0 1
1 0 0 1 0	-1 0 1 -1
1 0 0 1 1	-1 0 1 0
1 0 1 0 0	-1 1 -1 0
1 0 1 0 1	-1 1 -1 1
1 0 1 1 0	-1 1 0 -1
1 0 1 1 1	-1 1 0 0
1 1 0 0 1	0 -1 0 1
1 1 0 1 0	0 -1 1 -1
1 1 0 1 1	0 -1 1 0
1 1 1 0 1	0 0 -1 1

It should be noted that not all pair-constrained codes are suitable to cooperate with an MPD detector. For example, when choosing the pairs $(1,2)$ and $(2,3)$ rather than $(0, q-1)$ and $(q-1, 0)$, the resulting code does not satisfy Property 3 if $q \geq 5$ and $n \geq 4$, since, e.g., both $(0,1,2,3,3,\ldots,3)$ and $(1,2,3,4,4\ldots,4)$ have the same difference vector $(1,1,1,0,0,\ldots,0)$. Weber et al. [51] have studied optimal Pearson codes, which are the largest codes contained in $[q]^n$ that can be correctly decoded in the zero-error case. The pair-constrained property is not a necessary but sufficient condition of codes for the MPD. Properties of optimal Pearson codes for the detection scheme with a difference operator are interesting topics for further research.

6.4.1. CARDINALITY

The cardinality of S_{pc} is denoted by $N(n)$. For the binary case, $q = 2$, we simply find that

$$N(n) = 2^n - 2n,$$

since S_{pc} consists of all sequences in $\{0,1\}^n$, except the sequences without or with only one transition of the $0 \to 1$ or $1 \to 0$ type, i.e., $(0,\ldots,0,1,\ldots,1)$, $(1,\ldots,1,0,\ldots,0)$, $(0,\ldots,0)$, and $(1,\ldots,1)$.

In general, we can calculate the number $N(n)$ as follows. Consider the complement set $\overline{S_{pc}}$ of S_{pc} in $[q]^n$, and let $M(n) = \left|\overline{S_{pc}}\right|$ denote the cardinality of this complement set. We have

$$M(n) = |K_2| + |K_3| - |K_1|,$$

where

$$K_1 = \{\mathbf{x} \in [q]^n \,|(x_{i-1}, x_i) \notin \{(0, q-1), (q-1, 0)\}, \forall i = 2, \dots, n\},$$
$$K_2 = \{\mathbf{x} \in [q]^n \,|(x_{i-1}, x_i) \neq (0, q-1), \forall i = 2, \dots, n\},$$
$$K_3 = \{\mathbf{x} \in [q]^n \,|(x_{i-1}, x_i) \neq (q-1, 0), \forall i = 2, \dots, n\}.$$

Let $a_n = |K_1|$. We consider the following partition of K_1:

$$K_1^* = \{\mathbf{x} \in K_1 \,|x_n \in \{1, \dots, q-2\}\},$$

$$K_1^\circ = \{\mathbf{x} \in K_1 \,|x_n \in \{0, q-1\}\},$$

and let $a_n^* = \left|K_1^*\right|$ and $a_n^\circ = \left|K_1^\circ\right|$. If we add a symbol x_{n+1} to the end of $\mathbf{x} \in K_1^*$, then x_{n+1} can be a random symbol from $\{0, \dots, q-1\}$. While adding a symbol x_{n+1} to the end of $\mathbf{x} \in K_1^\circ$ is different as x_{n+1} can be a random one from $\{1, \dots, q-2\}$, or the same symbol as x_n in order to avoid the undesired adjacent pairs. Then we have the recursive relations

$$a_n^* = (q-2)(a_{n-1}^* + a_{n-1}^\circ)$$

and

$$a_n^\circ = 2a_{n-1}^* + a_{n-1}^\circ,$$

from which it follows for all $n \geq 2$ that

$$\begin{aligned} a_n &= a_n^* + a_n^\circ \\ &= (q-2)(a_{n-1}^* + a_{n-1}^\circ) + 2a_{n-1}^* + a_{n-1}^\circ \\ &= (q-1)(a_{n-1}^* + a_{n-1}^\circ) + (q-2)(a_{n-2}^* + a_{n-2}^\circ) \\ &= (q-1)a_{n-1} + (q-2)a_{n-2} \end{aligned} \tag{6.12}$$

with initial conditions $a_0 = 1$ and $a_1 = q$.

Let $b_n = |K_2|$. Similarly, consider the following non-empty partition of K_2

$$K_2^* = \{\mathbf{x} \in K_2 \,|x_n \in \{1, \dots, q-1\}\},$$

$$K_2^\circ = \{\mathbf{x} \in K_2 \,|x_n = 0\},$$

and let $b_n^* = \left|K_2^*\right|$ and $b_n^\circ = \left|K_2^\circ\right|$. Thus we have the following recursive relations

$$b_{n+1}^* = (q-1)b_n^* + (q-2)b_n^\circ$$

$$b_{n+1}^\circ = b_n^* + b_n^\circ$$

from which it follows that

$$\begin{aligned} b_{n+1} &= b_{n+1}^* + b_{n+1}^\circ \\ &= (q-1)b_n^* + (q-2)b_n^\circ + b_n^* + b_n^\circ \\ &= q(b_n^* + b_n^\circ) - b_n^\circ \\ &= qb_n - b_{n-1}, \end{aligned} \tag{6.13}$$

Table 6.3: Codebook Sizes $N_o(n) - 2$ and $N(n)$.

n	$N_o(n) - 2$	$N(n), q = 2$	$N(n), q = 3$
4	2	8	12
5	6	22	54
6	6	52	214
7	18	114	790
8	16	240	2786
9	50	494	9516
10	46	1004	31746

with initial conditions $b_0 = 1$ and $b_1 = q$. The number of sequences in K_3 follows the same recurrence scheme as in K_2.

Since $N(n) = q^n - M(n)$ and $M(n) = |K_2| + |K_3| - |K_1| = 2b_n - a_n$, we have

$$N(n) = q^n + a_n - 2b_n, \tag{6.14}$$

from which we can derive the recursive relation

$$\begin{aligned} N(n) \quad = \quad & (2q-1)N(n-1) - (q^2 - 2q + 3)N(n-2) \\ & -(q^2 - 3q + 1)N(n-3) \\ & +(q-2)N(n-4) + 2q^{n-4} \end{aligned} \tag{6.15}$$

6

for all $n \geq 4$, with initial conditions $N(0) = 0$, $N(1) = 0$, $N(2) = 0$, and $N(3) = 2$. Relation (6.15) can be shown by replacing all $N(i)$, $n-4 \leq i \leq n$, by $q^i + a_i - 2b_i$, according to (6.14), and then (repeatedly) applying (6.12) and (6.13) on the a_i and b_i, $n-2 \leq i \leq n$, until expressions containing only a_{n-4}, a_{n-3}, b_{n-4}, and b_{n-3} are left. The results for the left-hand and right-hand sides are the same, which proves the claim.

Table 6.3 shows results of computations of $N(n)$ for binary and ternary codes. Also, for comparison purposes, it includes the sizes $N_o(n)$ of the binary mass-centered codes [59] mentioned in Section 6.2. It is easily verified that both the all-zero and all-one sequences are center-of-mass constrained. For unambiguous MPD detection, the all-zero and all-one sequences should be excluded so that the number of available codewords is $N_o(n) - 2$. Note that the remaining binary mass-centered sequences are all in the binary pair-constrained code of the same length. However, this code contains many other sequences as well, and therefore $N(n)$ considerably exceeds $N_o(n) - 2$ in the binary case.

6.4.2. REDUNDANCY

Since the redundancy of S_{pc} is equal to

$$r(n) = n - \log_q N(n), \tag{6.16}$$

it would be convenient for evaluation purposes to have an explicit expression for $N(n)$ rather than a recursive one. Here we will derive such an expression using generating functions, which are described in, e.g., [83].

We start by rewriting the recurrence (6.12) using the Kronecker delta symbol, such that it is valid for all $n \geq 0$ (assuming $a_n = 0$ for all $n < 0$):

$$a_n - (q-1)a_{n-1} - (q-2)a_{n-2} - \delta_{n0} - \delta_{n1} = 0. \tag{6.17}$$

Let the ordinary generating function of a_n be denoted by $A(z) = \sum_{n=0}^{\infty} a_n z^n$. Then we derive $A(z)$ by multiplying (6.17) by z^n and summing over n, which gives

$$\sum_{n=0}^{\infty} a_n z^n - (q-1)\sum_{n=0}^{\infty} a_{n-1} z^n - (q-2)\sum_{n=0}^{\infty} a_{n-2} z^n - 1 - z = 0.$$

We can rewrite the above equation as

$$A(z) - (q-1)zA(z) - (q-2)z^2 A(z) = 1 + z.$$

Hence, we have

$$A(z) = \frac{1+z}{1-(q-1)z-(q-2)z^2}. \tag{6.18}$$

Similarly, we can rewrite (6.13) as

$$b_n - qb_{n-1} + b_{n-2} - \delta_{n0} = 0, \tag{6.19}$$

for all $n \geq 0$ (assuming $b_n = 0$ for all $n < 0$), which leads to the ordinary generating function of b_n being

$$B(z) = \sum_{n=0}^{\infty} b_n z^n = \frac{1}{1-qz+z^2}. \tag{6.20}$$

Next, we find the power series of $A(z)$ and $B(z)$ by applying Taylor's theorem, in which a_n and b_n, respectively, appear as the coefficients of z^n. This results in

$$a_n = \frac{(q+\lambda-1)^n(\lambda+q+1) + (q-\lambda-1)^n(\lambda-q-1)}{2^{n+1}\lambda}, \tag{6.21}$$

where

$$\lambda = \sqrt{q^2+2q-7},$$

and

$$b_n = U_n(q/2), \tag{6.22}$$

where

$$U_n(x) = \frac{(x+\sqrt{x^2-1})^{n+1} - (x-\sqrt{x^2-1})^{n+1}}{2\sqrt{x^2-1}}$$

is the Chebyshev polynomial of the second kind and $U_n(1) = n+1$ [83]. Hence, combining with (6.14) leads to the explicit expression for $N(n)$ given as

$$N(n) = q^n - 2U_n(q/2) + \frac{(q+\lambda-1)^n(\lambda+q+1) + (q-\lambda-1)^n(\lambda-q-1)}{2^{n+1}\lambda}. \tag{6.23}$$

For example, we find for $q = 3$ that

$$N(n) = 3^n - \frac{(3+\sqrt{5})^{n+1} - (3-\sqrt{5})^{n+1}}{2^n\sqrt{5}} + \frac{1}{2}(1+\sqrt{2})^{n+1} + \frac{1}{2}(1-\sqrt{2})^{n+1},$$

which confirms the values in the most right column of Table 6.3.

From (6.16) and the fact that $\log_q(1+x) \approx x/\ln q$ for small x, we obtain the approximate expression for the redundancy of S_{pc} given as

$$r(n) \approx \left[\frac{2U_n(q/2)}{q^n} - \frac{(q+\lambda-1)^n(\lambda+q+1)+(q-\lambda-1)^n(\lambda-q-1)}{2^{n+1}q^n\lambda}\right] / \ln q. \tag{6.24}$$

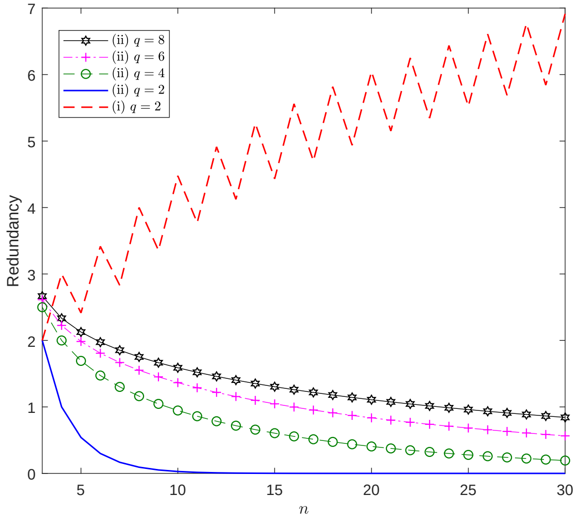


Figure 6.2: Redundancy versus codeword length n: (i) $r_o(n)$ for $q = 2$; (ii) $r(n)$ for $q = 2,4,6,8$.

Figure 6.2 shows the redundancy of S_{pc} as a function of the codeword length n for $q = 2,4,6,8$. As we can see, $r(n)$ approaches 0 as the codeword length increases, and the rate of convergence to 0 decreases as q grows. Also included in the figure is the redundancy of binary mass-centered codes, $r_o(n) = n - \log(N_o(n)-2)$, where $N_o(n)-2$ is the number of binary mass-centered sequences of length n without the all-zero and all-one words [59]. Note the significant difference between $r_o(n)$ and $r(n)$ for $q = 2$. With the increase of n, $r_o(n) = O(\log n)$ has an upward trend, while $r(n)$ experiences a downward trend to 0. For example, $r_o(10) \approx 4.5$ is more than 100 times larger than $r(10) \approx 0.028$. We conclude that the redundancy of the proposed pair-constrained codes gives a significant improvement compared to the corresponding mass-centered codes.

6.4.3. SYSTEMATIC CODING

Simple implementations of high-rate pair-constrained codes can be constructed with slight modifications of a systematic method used in [54]. An extremely simple fixed-length to fixed-length scheme is to fill the first $n-3$ positions in the code sequence $\mathbf{x}$ with information symbols and to reserve the last three symbols for reference purposes: $x_{n-2} = 0$, $x_{n-1} = q-1$, and $x_n = 0$. Due to the fixed last 3 symbols, which act as references, the redundancy of this method is 3, but it would preferably approach zero for large values of n. We propose a systematic variable-length to fixed-length scheme, $\mathscr{S}_{pc}$, for which the redundancy decreases in n. It is explained according to the flowchart in Figure 6.3 and reads as follows.

1. Take $n-3$ information symbols from the q-ary source and set these as $(x_1, x_2, \ldots, x_{n-3})$.
2. If $(x_{i-1}, x_i) = (0, q-1)$ and $(x_{j-1}, x_j) = (q-1, 0)$ for at least one $2 \le i, j \le n-3$, then choose x_{n-2}, x_{n-1}, and x_n to be information symbols.
3. If $(x_{i-1}, x_i) = (0, q-1)$ for at least one $2 \le i \le n-3$ and $(x_{j-1}, x_j) \ne (q-1, 0), \forall j = 2, \ldots, n-3$, then
   - if $x_{n-3} = q-1$, then set $x_{n-2} = 0$ and choose x_{n-1} and x_n to be information symbols;
   - otherwise, set $x_{n-2} = q-1$, $x_{n-1} = 0$ and choose x_n to be an information symbol.
4. If $(x_{i-1}, x_i) = (q-1, 0)$ for at least one $2 \le i \le n-3$ and $(x_{j-1}, x_j) \ne (0, q-1), \forall j = 2, \ldots, n-3$, then
   - if $x_{n-3} = 0$, then set $x_{n-2} = q-1$ and choose x_{n-1} and x_n to be information symbols;
   - otherwise, set $x_{n-2} = 0$, $x_{n-1} = q-1$ and choose x_n to be an information symbol.
5. If $(x_{i-1}, x_i) \notin \{(q-1, 0), (0, q-1)\}$, $\forall i = 2, \ldots, n-3$, then
   - if $x_{n-3} = 0$, set $x_{n-2} = q-1$, $x_{n-1} = 0$, and choose x_n to be an information symbol;
   - elseif $x_{n-3} = q-1$, set $x_{n-2} = 0$, $x_{n-1} = q-1$, and choose x_n to be an information symbol;
   - otherwise, set $x_{n-2} = 0$, $x_{n-1} = q-1$, and $x_n = 0$.

Since any code sequence obtained this way contains at least one $(q-1, 0)$ and $(0, q-1)$ adjacent symbol pairs, the code sequence $\mathbf{x}$ is indeed in S_{pc} and the information symbols can be uniquely retrieved from $\mathbf{x}$. Also, the $n-2$, $n-1$, or n information symbols can easily be retrieved from $\mathbf{x}$. Since the number of information symbols may vary from codeword to codeword, while the length of the codewords is fixed at n, this can be considered a variable-length to fixed-length coding procedure. The redundancy of this scheme is given in the following theorem.

Theorem 11. *For a memoryless uniform q-ary source, the redundancy of coding scheme $\mathscr{S}_{pc}$ is*

$$\frac{2U_{n-3}(q/2)}{q^{n-3}} - \frac{(q+\lambda-1)^{n-3}(\lambda+q+1)+(q-\lambda-1)^{n-3}(\lambda-q-1)}{2^{n-2}\lambda q^{n-3}} + 2\frac{(q-\gamma)^n(\gamma-q+2)+(q+\gamma)^n(\gamma+q-2)}{2^{n-2}\gamma q^{n-3}},$$

where $\lambda = \sqrt{q^2+2q-7}$ and $\gamma = \sqrt{q^2-4}$.

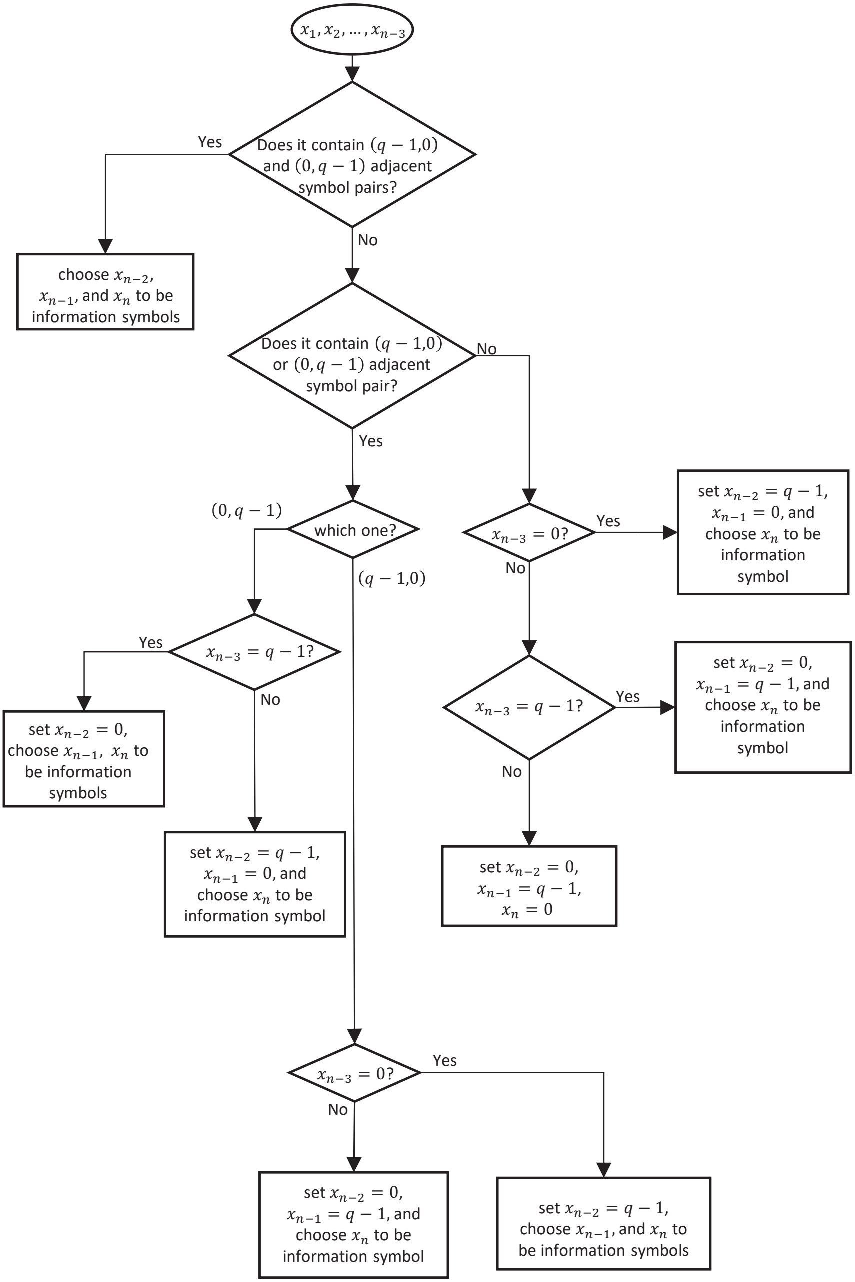


Figure 6.3: Flow chart of systematic variable-length to fixed-length coding scheme.

Proof. The result can be obtained using parameters defined in Section 6.4.1, with the observations that: (i) the a_n is the size of a set, K_1, containing sequences of length n without $(0, q-1)$ and $(q-1, 0)$ adjacent pairs; and (ii) the $b_n - a_n$ indicates the number of sequences of length n without one of these two adjacent pairs while another is included. Note that superscripts, such as asterisk and circle, represent the numbers of sequences in subsets of K_1, K_2, and K_3. Section 6.4.1 has given their definitions in detail.

The probability that a code sequence $\mathbf{x}$ has three redundant symbols is

$$\frac{a^*_{n-3}}{q^{n-3}}, \tag{6.25}$$

which is the probability of having an information sequence of length $n-3$ without $(0, q-1)$ and $(q-1, 0)$ adjacent pairs and $x_{n-3} \notin \{0, q-1\}$.

Further, we calculate the probability that $\mathbf{x}$ has two redundant symbols, which contains three situations as shown in Figure 6.3. The first part of it is

$$\frac{a^o_{n-3}}{q^{n-3}}, \tag{6.26}$$

which is the probability of having an information sequence of length $n-3$ without $(0, q-1)$ and $(q-1, 0)$ adjacent symbol pairs and $x_{n-3} \in \{0, q-1\}$. The second case of that $\mathbf{x}$ has two redundant symbols, is when $\mathbf{x}$ is without $(0, q-1)$ but at least one $(q-1, 0)$ and the last symbol is not zero. We consider the following non-empty partition of the set K_1° as

6

$$K_1^{\circ 1} = \{\mathbf{x} \in K_1 \,|\, x_n = 0\},$$

$$K_1^{\circ 2} = \{\mathbf{x} \in K_1 \,\big|\, x_n = q-1\},$$

and let $a_n^{\circ 1} = \left|K_1^{\circ 1}\right|$ and $a_n^{\circ 2} = \left|K_1^{\circ 2}\right|$. Note that $a_n^\circ = a_n^{\circ 1} + a_n^{\circ 2}$. Then we have that

$$\frac{b^*_{n-3} - a^*_{n-3} - a^{\circ 2}_{n-3}}{q^{n-3}} \tag{6.27}$$

is the probability of having an information sequence of length $n-3$ without $(0, q-1)$ but at least one $(q-1, 0)$ and $x_{n-3} \neq 0$. Similarly, for the last situation we have that

$$\frac{b^*_{n-3} - a^*_{n-3} - a^{\circ 1}_{n-3}}{q^{n-3}} \tag{6.28}$$

is the probability of having an information sequence of length $n-3$ without $(q-1, 0)$ but at least one $(0, q-1)$ and $x_{n-3} \neq q-1$. Thus the probability that $\mathbf{x}$ has two redundant symbols is

$$\frac{2b^*_{n-3} - 2a^*_{n-3}}{q^{n-3}}, \tag{6.29}$$

which is the summation of (6.26), (6.27) and (6.28).

The probability that $\mathbf{x}$ has only a redundant symbol is

$$\frac{b^\circ_{n-3} - a^{\circ 1}_{n-3}}{q^{n-3}} + \frac{b^\circ_{n-3} - a^{\circ 2}_{n-3}}{q^{n-3}}, \tag{6.30}$$

which is the probability of having an information sequence of length $n-3$: i) (the first term) without $(0,q-1)$ but at least one $(q-1,0)$ and $x_{n-3}=0$, and ii) (the second term) without $(q-1,0)$ but at least one $(0,q-1)$ and $x_{n-3}=q-1$. Hence, the redundancy is summation of three times the term in (6.25), two times the term in (6.29), and the term in (6.30), that is,

$$\begin{aligned}
&3\frac{a^*_{n-3}}{q^{n-3}}+2\frac{2b^*_{n-3}-2a^*_{n-3}}{q^{n-3}}+\frac{b^\circ_{n-3}-a^{\circ 1}_{n-3}}{q^{n-3}}+\frac{b^\circ_{n-3}-a^{\circ 2}_{n-3}}{q^{n-3}}\\
=&\frac{4b^*_{n-3}-a^*_{n-3}+2b^\circ_{n-3}-a^\circ_{n-3}}{q^{n-3}}\\
=&\frac{2b_{n-3}-a_{n-3}+2b^*_{n-3}}{q^{n-3}},
\end{aligned}$$

where the first equation follows that $a^\circ_n=a^{\circ 1}_n+a^{\circ 2}_n$ and the second equation follows that $a_n=a^*_n+a^\circ_n$ and $b_n=b^*_n+b^\circ_n$. Note that $b^*_n=\left|K^*_2\right|$, which has a recursive relation $b^*_{n+1}=qb^*_n-b^*_{n-1}$ with initial conditions $b^*_0=1$ and $b^*_1=q-1$. Using generating functions which follows the same approach as (6.17)- (6.18) results in

$$b^*_n=\frac{(q-\gamma)^n(\gamma-q+2)+(q+\gamma)^n(\gamma+q-2)}{2^{n+1}\gamma},\ \text{if } q>2, \tag{6.31}$$

where $\gamma=\sqrt{q^2-4}$, and $b^*_n=1$ if $q=2$. With (6.21) and (6.22) we obtain the expression stated in the theorem. □

The redundancy of $\mathscr{S}_{pc}$ as stated in Theorem 11 is a factor τ higher than the redundancy as stated in Equation (6.24), where τ and the redundancy of $\mathscr{S}_{pc}$ are pictured in Figure 6.4. For the binary case $q=2$, $\mathscr{S}_{pc}$ encoding algorithm has a redundancy of $\frac{2n-4}{2^{n-3}}$. However, pair-constraint requires at most two redundant symbols as the last symbol in binary sequences of length $n-3$ is either 0 or 1. We propose a less redundant algorithm for the binary case whose redundant symbols are located at x_{n-1} and/or x_n. It reads as follows.

1. Take $n-2$ information symbols from the binary source and set these as $(x_1,x_2,\ldots,x_{n-2})$.
2. If $(x_{i-1},x_i)=(0,1)$ and $(x_{j-1},x_j)=(1,0)$ for at least one $2\le i,j\le n-2$, then choose x_{n-1} and x_n to be information symbols.
3. If $(x_{i-1},x_i)=(0,1)$ for at least one $2\le i\le n-2$ and $(x_{j-1},x_j)\ne(1,0),\forall j=2,\ldots,n-2$, then set $x_{n-1}=0$ and choose x_n to be information symbols.
4. If $(x_{i-1},x_i)=(1,0)$ for at least one $2\le i\le n-2$ and $(x_{j-1},x_j)\ne(0,1),\forall j=2,\ldots,n-2$, then $x_{n-1}=1$ and choose x_n to be an information symbol.
5. If $(x_{i-1},x_i)\notin\{(1,0),(0,1)\},\forall i=2,\ldots,n-2$, then
   - if $x_{n-2}=0$, set $x_{n-1}=1$ and $x_n=0$;
   - otherwise, set $x_{n-1}=0$ and $x_n=1$.

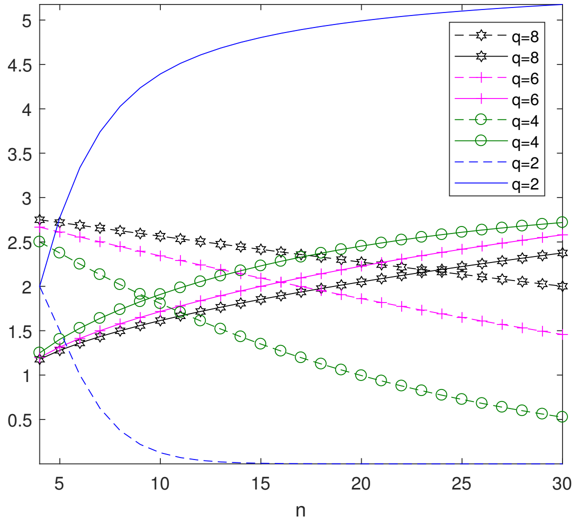


Figure 6.4: The factor τ (solid lines) and the redundancy of $\mathscr{S}_{pc}$ (dash lines) versus codeword length n.

6

It is easy to know that two sequences (all-zero and all-one sequences) of length $n-2$ have two redundant symbols, and $2(n-3)$ sequences of length $n-2$ have a redundant symbol. Thus the redundancy of this scheme is

$$\frac{2n-2}{2^{n-2}}, \tag{6.32}$$

which is the summation of two times the probability of having an information sequence of length $n-2$ with two redundant symbols and the probability of having an information sequence of length $n-2$ with a redundant symbol.

6.5. NOISE CONSIDERATIONS

This section investigates the error performance of the proposed detection over channels with gain and varying offset mismatch. The receiver uses the Pearson distance for the evaluation of the difference received word, where we assume that $\Delta\mathbf{x} \in \Delta S$ is sent, and received vector is $\Delta\mathbf{r} = \Delta\mathbf{x} + \Delta\mathbf{v}$. Note that we dropped the unknown parameters a, $b\mathbf{1}$, and $c\mathbf{s}$ since, as established above, the performance of MPD with a difference operator is (a, b, c)-immune. The receiver errs if there is at least one codeword $\Delta\hat{\mathbf{x}} \neq \Delta\mathbf{x}$, $\Delta\hat{\mathbf{x}} \in \Delta S$, such that

$$\delta_P(\Delta\mathbf{r}, \Delta\hat{\mathbf{x}}) < \delta_P(\Delta\mathbf{r}, \Delta\mathbf{x})$$

or

$$-\frac{1}{\sigma_{\Delta\hat{\mathbf{x}}}}\sum_{i=1}^{n-1}(\Delta x_i + \Delta v_i)(\Delta\hat{x}_i - \overline{\Delta\hat{\mathbf{x}}}) < -\frac{1}{\sigma_{\Delta\mathbf{x}}}\sum_{i=1}^{n-1}(\Delta x_i + \Delta v_i)(\Delta x_i - \overline{\Delta\mathbf{x}})$$

or

$$\sum_{i=1}^{n-1}(\Delta x_i + \Delta v_i)(a_i - \hat{a}_i) < 0, \tag{6.33}$$

where

$$a_i = \frac{\Delta x_i - \overline{\Delta \mathbf{x}}}{\sigma_{\Delta \mathbf{x}}}$$

and

$$\hat{a}_i = \frac{\Delta \hat{x}_i - \overline{\Delta \hat{\mathbf{x}}}}{\sigma_{\Delta \hat{\mathbf{x}}}}.$$

The distribution $\Delta \mathbf{v}$ has a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and $(n-1)\times(n-1)$ covariance matrix

$$\begin{bmatrix} 2\sigma^2 & -\sigma^2 & 0 & 0 & \dots & 0 \\ -\sigma^2 & 2\sigma^2 & -\sigma^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & -\sigma^2 & 2\sigma^2 & -\sigma^2 \\ 0 & \dots & 0 & 0 & -\sigma^2 & 2\sigma^2 \end{bmatrix}.$$

The left-hand side of (6.33) is a stochastic variable with distribution $\mathcal{N}(\alpha_1, \beta_1)$, where

$$\alpha_1 = \sum_{i=1}^{n-1} \Delta x_i (a_i - \hat{a}_i)$$

and

$$\beta_1 = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}(a_i - \hat{a}_i)(a_j - \hat{a}_j)\Sigma_{ij}.$$

Since $\sum_{i=1}^{n-1} a_i = \sum_{i=1}^{n-1} \hat{a}_i = 0$, we have

$$\alpha_1 = \sigma_{\Delta \mathbf{x}}(1 - \rho_{\Delta \mathbf{x}, \Delta \hat{\mathbf{x}}}) \tag{6.34}$$

and

$$\beta_1 = 2\sigma^2 \left[2(1-\rho_{\Delta \mathbf{x},\Delta \hat{\mathbf{x}}}) - \zeta(\Delta \mathbf{x}, \Delta \mathbf{x}) + \zeta(\Delta \mathbf{x}, \Delta \hat{\mathbf{x}}) + \zeta(\Delta \hat{\mathbf{x}}, \Delta \mathbf{x}) - \zeta(\Delta \hat{\mathbf{x}}, \Delta \hat{\mathbf{x}})\right], \tag{6.35}$$

where for two vectors $\mathbf{u}$ and $\mathbf{v}$ of length n,

$$\zeta(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}_{1,n-1}\cdot \mathbf{v}_{2,n} + \bar{\mathbf{u}} v_1 + \bar{\mathbf{v}} u_n - (n+1)\bar{\mathbf{u}}\bar{\mathbf{v}}}{\sigma_{\mathbf{u}}\sigma_{\mathbf{v}}}.$$

Note that $\zeta(\mathbf{u},\mathbf{v}) \neq \zeta(\mathbf{v},\mathbf{u})$. We define the square of the distance between the vectors $\Delta \mathbf{x}$ and $\Delta \hat{\mathbf{x}}$ by

$$d^2(\Delta \mathbf{x}, \Delta \hat{\mathbf{x}}) = \frac{\alpha_1^2}{\beta_1'}, \tag{6.36}$$

where $\beta_1' = \beta_1/(2\sigma^2)$. The distance $d^2(\Delta \mathbf{x}, \Delta \hat{\mathbf{x}})$ is not symmetric in the vector $\Delta \mathbf{x}$ and $\Delta \hat{\mathbf{x}}$, that is, in general $d^2(\Delta \mathbf{x}, \Delta \hat{\mathbf{x}}) \neq d^2(\Delta \hat{\mathbf{x}}, \Delta \mathbf{x})$. Since the probability that $\delta_P(\Delta \mathbf{r}, \Delta \hat{\mathbf{x}}) < \delta_P(\Delta \mathbf{r}, \Delta \mathbf{x})$ equals

$$Q\left(\frac{d(\Delta \mathbf{x}, \Delta \hat{\mathbf{x}})}{\sqrt{2}\sigma}\right),$$

the word error rate (WER) over all coded sequences $\Delta\mathbf{x}$ is upperbounded by

$$\text{WER} < \frac{1}{|\Delta S|} \sum_{\Delta\mathbf{x}\in\Delta S} \sum_{\Delta\hat{\mathbf{x}}\neq\Delta\mathbf{x}} Q\left(\frac{d(\Delta\mathbf{x},\Delta\hat{\mathbf{x}})}{\sqrt{2}\sigma}\right). \tag{6.37}$$

At high signal-to-noise ratios, the WER is approximated by its dominant term

$$WER \sim N_d \times Q\left(\frac{\Delta d_{min}}{2\sigma}\right), \tag{6.38}$$

where N_d is the average number of nearest neighboring codewords at minimum distance Δd_{min}, defining by

$$\Delta d^2_{min} = \min_{\Delta\mathbf{x},\Delta\hat{\mathbf{x}}\in\Delta S,\Delta\hat{\mathbf{x}}\neq\Delta\mathbf{x}} d^2(\Delta\mathbf{x},\Delta\hat{\mathbf{x}}). \tag{6.39}$$

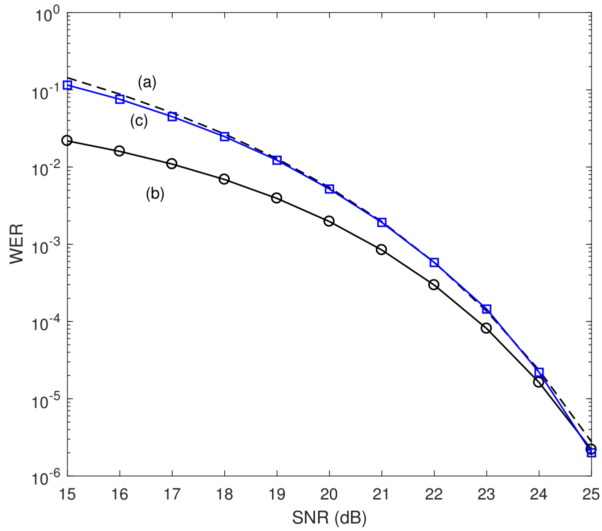


Figure 6.5: Word error rate (WER) as a function of the signal-to-noise ratio (SNR = $-20\log_{10}\sigma$), for $n = 7$ and $q = 2$ computed using upper bound (6.37), curve (a), and by using the approximation (6.38), curve (b). The square marked line (c) shows simulated results of the proposed scheme with channel parameters $a = 1.07$, $b = 0.07$, and $c = 0.04$.

Figure 6.5 shows WER results for $n = 7$ and $q = 2$, where we compared the WER upper bound (6.37), its approximation (6.38), and simulated results of pair-constrained codes using the proposed scheme. The channel parameters are $a = 1.07$, $b = 0.07$, and $c = 0.04$.

Figure 6.6 shows simulated WER results of a communication system using MED with parameters $n = 7$ and $q = 2$ a) without mismatch, and b) with mismatch $a = 1.07$, $b = 0.07$, and $c = 0.04$. The proposed scheme that MPD is used with a difference operator is also pictured in curve c). We may notice that the error performance is seriously affected by the mismatch. The error performance of the proposed scheme where offers intrinsic resistance to gain and varying offset mismatch at the cost of a reduced noise margin.

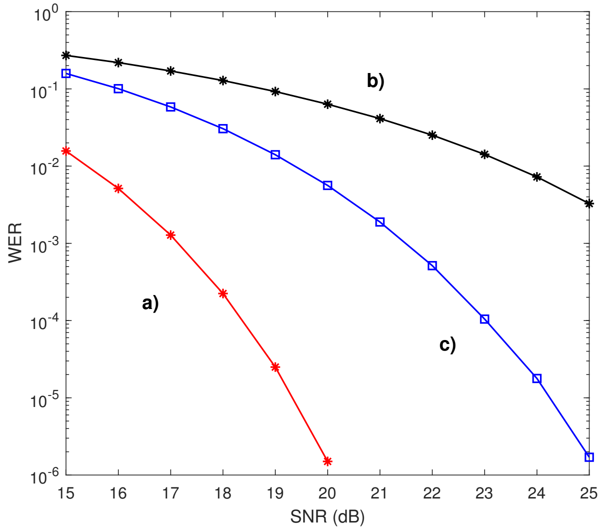


Figure 6.6: Word error rate (WER) for $n = 7$ and $q = 2$, a) without channel mismatch, b) with channel mismatch $a = 1.07$, $b = 0.07$, and $c = 0.04$ both using minimum Euclidean distance detection, and as a comparison c) for the proposed scheme that MPD is used with a difference operator.

6

6.6. DISCUSSION

We have presented a scheme for channels with unknown gain and varying offset, where minimum Pearson distance detection is used in conjunction with a difference operator and pair-constrained codes. These codes have significantly less redundancy compared to the previously proposed mass-centered codes, making the new scheme an attractive alternative for practical applications. However, there are still some issues that need to be addressed.

The introduction of the difference operator is very effective to deal with the unknown varying offset, but it follows from the analysis in Section 6.5 that it doubles the noise power. Hence, in the error analysis, this extra 3 dB loss should be taken into account, and it makes the scheme less suitable for applications in which the noise is dominant over the (varying) offset. An interesting topic for further research is investigating to which extent the involvement of an error-correcting code in the scheme can help resolve this.

Another concern is the complexity of the proposed scheme. The use of the difference operator demands extra subtractions. However, the major problem is that the minimization operation (6.10) requires $|\Delta S|$ computations, which is impractical for codes with large cardinalities. In [49], it has been shown that the number of computations can be significantly reduced by considering the codebook as the union of a number of constant composition codes, which makes, at the expense of extra sorting operations, the number of options in the minimization equal to only the number of such subcodes. Similar complexity reduction could be explored for the setting under consideration here as well.

The material in this chapter has appeared in IEEE Communications Letters [84].

7

Conclusion and Future Work

All models are wrong, but some are useful.

George Box, 1978.

7.1. Conclusion

In this thesis, we have studied channels that are not only distorted by noise but also by another important channel impairment, *gain and/or offset mismatch*. Focus is put on four different models of gain and/or offset mismatch in Chapters 3 to 6. The main results are summarized as follows:

- A basic model of noisy channels with offset mismatch is introduced in Chapter 3, where an unknown offset is assumed to be in the all-one direction, that is, $b\mathbf{1}$. This model assumes that the offset mismatch is constant within one codeword block length and may vary block by block.

  Firstly, channels are considered for which both the noise and offset are bounded. For such cases, Euclidean distance based decoding (MED), modified Pearson distance based decoding (MMPD), and Maximum Likelihood (ML) decoding are considered. In particular, for each of these decoders, bounds are determined on the magnitudes of the noise and offset intervals, leading to a word error rate equal to zero.

  Secondly, for channels with Gaussian noise and offset, it is shown that an ML measurement is a weighted average of Euclidean distance and Pearson distance. The ML measurement tends towards the offset-resistant Pearson distance when the codeword length n is increasing.

  Thirdly, we propose a concatenated scheme and its corresponding decoding algorithm with Gaussian noise and offset mismatch. The concatenation is between a

Reed-Solomon (RS) code and a certain coset of a binary block code. The MMPD detection is used to decode the inner code that guarantees immunity to channel offset mismatch. Its output will be given to a two-stage hybrid decoding algorithm for the outer RS code. Simulation results demonstrate that the proposed concatenated scheme can achieve significant coding gain and maintain immunity to offset mismatch.

- Chapter 4 focuses on noisy channels with offset mismatch, where, most importantly, the offset is assumed to be signal dependent, i.e., $b_{\mathbf{x}}$. It means that the value of the offset may differ for distinct signal levels rather than being the same for all levels.

  Firstly, an ML decision criterion is derived, assuming uniform distributions for both the noise and the offset. In particular, for the proposed ML decoder, bounds are determined on the standard deviations of the noise and the offset, which lead to a word error rate equal to zero. For example, we assume that σ is the standard deviation of the noise and β_0 and β_1 are the standard deviations of the offsets of signal levels 0 and 1. With $2\sigma+\beta_0+\beta_1 \leq 1/\sqrt{3}$, the ML decoder achieves a WER equal to zero for a binary codebook.

  Later, an ML criterion is considered for the case of signals suffering from Gaussian noise and signal dependent offset, where the correlation between different offset random variables is considered as well. We show that an ML criterion in the prior art can be derived as a special case of our decoding criterion obtained by letting offsets be identically fully correlated distributed. Besides the ML criterion itself, an option to reduce the complexity for codebooks consisting of the union of constant weight sets is also considered. We provide a brief performance analysis, confirming the superiority of the newly developed ML decoder over classical decoders based on the Euclidean or Pearson distances.

- The attention of Chapter 5 is put on a channel model in which the retrieved data is corrupted by Gaussian noise, gain a and offset mismatch b. A general framework of ML decoding criteria for such channels is derived, where gain and offset are restricted to certain ranges, $0 < a_1 \leq a \leq a_2$ and $b_1 \leq b \leq b_2$. We also show that certain known criteria appear as special cases of this framework. For instance, by letting $a_1 \to 0$, $a_2 \to \infty$, $b_1 \to -\infty$, $b_2 \to \infty$, we obtain the same ML decoding criterion (5.10) as proposed by Blackburn for the case of unbounded gain and offset.

  In addition, we discuss geometric interpretations of the gain and offset mismatch. Scaling a vector of mean 0 by gain a does not change the mean but scales the standard deviation by a factor of a. The offset only changes the mean of a vector, where it is reasonable to normalize vectors to eliminate the effect of offset. It is not difficult to see that the Pearson distance provides such a measurement. Thus a minimum Pearson distance detection is intrinsically immune to the gain a and the offset b, i.e., (a, b)-immune.

- We consider channels with gain and slowly varying offset in Chapter 6, where an unknown slope, c, of varying offset is introduced. We demonstrate that the combination of minimum Pearson distance detection and a difference operator is (a, b, c)-immune. The codebook should satisfy certain properties to allow minimum Pearson distance detection and prevent ambiguous decoding options. A new class of codes, pair-constrained codes, with the required properties, are proposed for such decoding scheme. Pair-constrained codes consist of codewords, where certain adjacent symbol pairs must appear at least once. When n increases, the redundancy of these codes experiences a downward trend to 0. We can conclude that they have significantly less redundancy than the previously proposed mass-centered codes, making the new scheme an attractive alternative for practical applications. In addition, we propose a systematic variable-length to fixed-length encoding algorithm of pair-constrained codes, and its redundancy is analyzed for memoryless uniform sources.

7.2. FUTURE WORK

The thesis has investigated coding techniques for channels with various gain and/or offset mismatch models. Analysis of gain and/or offset mismatch is complicated. There are still many open questions and interesting research topics. Some recommendations for future work are:

- In Chapter 3, we have discussed decoding criteria for channels with noise and offset mismatch, both of which are bounded. Zero word error rate performance is achievable for various decoding criteria by different constraints on noise and offset characteristics. Investigations about how codebooks can be generated satisfying σ-bound and/or $\sigma + \beta$-bound will be of interest. Further, suppose σ, β are fixed, it would be interesting to explore what rates can be achieved for different decoding schemes satisfying zero WER conditions.

- We have presented a scheme for channels with unknown gain and varying offset in Chapter 6, where MPD is used in conjunction with a difference operator and pair constrained codes. The introduction of the difference operator is very effective in dealing with the unknown varying offset, but it has the drawback of making the system more sensitive to noise. This penalty can also be physically justified from (6.11) by noticing the doubling of the noise power. Hence, it makes the scheme less suitable for applications in which the noise is dominant over the (varying) offset. A topic for further research is investigating to what extent the involvement of an error-correcting code in the scheme can help resolve this.

- There are also different techniques for solving the offset issues rather than presented in this thesis. Instead of eliminating the offset mismatch itself, a greater focus on the asymmetric or unidirectional errors caused by the offset could produce some interesting findings. In noisy channels with offset mismatch, the message has a much higher possibility of suffering asymmetric or unidirectional errors. Thus, in certain applications, such as Flash memories, it can be modeled by an asymmetric channel. A considerable amount of codes [85–87] has been designed for use on asymmetric channels. Further studies regarding the role of distances used on asymmetric channels would be worthwhile to extend into our case. We

7

hope that these distances will provide tools to increase systems' error-correction ability to channel mismatch.

- It would be challenging to investigate the information-theoretic channel capacity of noisy channels with gain and/or offset mismatch. Small-scale research has emerged evaluating the capacity of channels with physical effects like voltage leakage, such as a multi-level flash memory with input-dependent additive Gaussian noise [10], a binary asymmetric channel cascaded with a Gaussian mixture channel [70].

  A difficulty of this topic arises because the offset mismatch is an additive and block-wise distortion. Thus, its exact numerical results may be unsolvable, but we may examine upper bounds on the capacity instead. The investigation could start by first considering the offset, which is constant within a block, and then include other offset types. Many questions have been brought up in need of further investigation.

BIBLIOGRAPHY

[1] J. Rydning and D. Reinsel, *Worldwide Global StorageSphere Forecast, 2021–2025: To Save or Not to Save Data, That Is the Question*, Tech. Rep. (International Data Corporation, 2021).

[2] F. Masuoka, M. Asano, H. Iwahashi, T. Komuro, and S. Tanaka, *A new flash E^2PROM cell using triple polysilicon technology,* in *International Electron Devices Meeting* (IEEE, 1984) pp. 464–467.

[3] S. Aritome, *NAND Flash Memory Technologies* (John Wiley & Sons, Inc., Hoboken, NJ, USA, 2015).

[4] Micron Technology, *MT29F64G08CBAB-NAND FLASH data sheet*, Tech. Rep. (Micron, 2010).

[5] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, *Introduction to flash memory,* Proceedings of the IEEE **91**, 489 (2003).

[6] Y.-J. Choi, K.-D. Suh, Y.-N. Koh, J.-W. Park, K.-J. Lee, Y.-J. Cho, and B.-H. Suh, *A high speed programming scheme for multi-level NAND flash memory,* in *Proc. IEEE Symposium on VLSI Circuits. Digest of Technical Papers* (Honolulu, HI, USA, 1996) pp. 170–171.

[7] K. Takeuchi, T. Tanaka, and H. Nakamura, *A double-level-V select gate array architecture for multilevel NAND flash memories,* IEEE Journal of Solid-State Circuits **31**, 602 (1996).

[8] G. Atwood, A. Fazio, D. Mills, and B. Reaves, *Intel strataflash memory technology overview,* Intel Technology Journal , 1 (1997).

[9] D. Ajwani, I. Malinger, U. Meyer, and S. Toledo, *Characterizing the performance of flash memory storage devices and its impact on algorithm design,* in *International Workshop on Experimental and Efficient Algorithms* (Springer, Berlin, Heidelberg, 2008) pp. 208–219.

[10] X. Huang, A. Kavcic, G. D. X. Ma, and T. Zhang, *Multilevel Flash memories: channel modeling, capacities and optimal coding rates,* International Journal on Advances in Systems and Measurements **6**, 364 (2013).

[11] F. Sala, K. A. S. Immink, and L. Dolecek, *Error control schemes for modern flash memories: an overview,* IEEE Consumer Electronics **4**, 66 (2015).

[12] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu, *Data retention in MLC NAND flash memory: Characterization, optimization, and recovery,* in *Proc. IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)* (Burlingame, CA, USA, 2015) pp. 551–563.

[13] M. Rajab, J.-P. Thiers, and J. Freudenberger, *Read threshold calibration for non-volatile flash memories,* in *Proc. IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)* (Berlin, Germany, 2019) pp. 109–113.

[14] P. Hansson, *When SSD Performance Goes Awry,* Tech. Rep. (Techspot, 2015).

[15] T. M. Cover and J. A. Thomas, *Elements of Information Theory, 2nd ed.* (John Wiley & Sons, 2012).

[16] W. Liu, J. Rho, and W. Sung, *Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories,* in *Proc. IEEE Workshop on Signal Processing Systems Design and Implementation (SIPS'06)* (Banff, Canada, 2006) pp. 303–308.

[17] B. Chen, X. Zhang, and Z. Wang, *Error correction for multi-level NAND flash memory using Reed-Solomon codes,* in *Proc. IEEE Workshop on Signal Processing Systems* (Washington, USA, 2008) pp. 94–99.

[18] F. Zhang, H. D. Pfister, and A. Jiang, *LDPC codes for rank modulation in flash memories,* in *Proc. IEEE International Symposium on Information Theory (ISIT)* (2010) pp. 859–863.

[19] S. Solda, D. Vogrig, A. Bevilacqua, A. Gerosa, and A. Neviani, *Analog decoding of trellis coded modulation for multi-level flash memories,* in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)* (Seattle, WA, USA, 2008) pp. 744–747.

[20] J. H. Weber, *Bounds and Constructions for Binary Block Codes Correcting Asymmetric or Unidirectional Errors,* Ph.D. thesis, Delft University of Technology (1989).

[21] M. Blaum and H. Van Tilborg, *On t-error correcting/all unidirectional error detecting codes,* IEEE Transactions on Computers **38**, 1493 (1989).

[22] N. Azizi, F. N. Najm, and A. Moshovos, *Low-leakage asymmetric-cell SRAM,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems **11**, 701 (2003).

[23] H. Deogun, R. R. Rao, D. Sylvester, and D. Blaauw, *Leakage-and crosstalk-aware bus encoding for total power reduction,* in *Proc. 41st Design Automation Conference* (San Diego, CA, USA, 2004) pp. 779–782.

[24] D. Ielmini, D. Sharma, S. Lavizzari, and A. L. Lacaita, *Reliability impact of chalcogenide-structure relaxation in phase-change memory (PCM) cells-part I: Experimental study,* IEEE Transactions on Electron Devices **56**, 1070 (2009).

[25] M. Boniardi, D. Ielmini, S. Lavizzari, A. L. Lacaita, A. Redaelli, and A. Pirovano, *Statistics of resistance drift due to structural relaxation in phase-change memory arrays,* IEEE Transactions on Electron Devices **57**, 2690 (2010).

[26] J. P. A. Pérez, S. C. Pueyo, and B. C. López, *Automatic gain control* (Springer, 2011).

[27] K. A. S. Immink, *Coding schemes for multi-level Flash memories that are intrinsically resistant against unknown gain and/or offset using reference symbols,* Electronics Letters **50**, 20 (2014).

[28] Z. Mei, K. Cai, and X. He, *Deep learning-aided dynamic read thresholds design for multi-level-cell flash memories,* IEEE Transactions on Communications **68**, 2850 (2020).

[29] Z. Mei, K. Cai, and X. Zhong, *Neural network-based dynamic threshold detection for non-volatile memories,* in *Proc. IEEE International Conference on Communications (ICC)* (Shanghai, China, 2019) pp. 1–6.

[30] H. Choe, J. Jee, S. C. Lim, S. M. Joe, I. H. Park, and H. Park, *Machine-learning-based read reference voltage estimation for NAND flash memory systems without knowledge of retention time,* IEEE Access **8**, 176416 (2020).

[31] X. Zhong, K. Cai, Z. Mei, and T. Q. S. Quek, *Deep learning-based decoding of linear block codes for spin-torque transfer magnetic random access memory,* IEEE Transactions on Magnetics **57**, 1 (2021).

[32] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, *Rank modulation for flash memories,* IEEE Transactions on Information Theory **55**, 2659 (2009).

[33] H. Zhou, A. Jiang, and J. Bruck, *Error-correcting schemes with dynamic thresholds in nonvolatile memories,* in *Proc. IEEE International Symposium on Information Theory (ISIT)* (St. Petersburg, Russia, 2011) pp. 2143–2147.

[34] H. Zhou and J. Bruck, *Balanced modulation for nonvolatile memories,* arXiv: 1209.0744 (2012).

[35] F. Sala, R. Gabrys, and L. Dolecek, *Dynamic threshold schemes for multi-level non-volatile memories,* IEEE Transactions on Communications **61**, 2624 (2013).

[36] K. A. S. Immink, *Coding schemes for multi-level channels with unknown gain and/or offset,* in *Proc. IEEE International Symposium on Information Theory (ISIT)* (Istanbul, Turkey, 2013) pp. 709–713.

[37] D. Slepian, *Permutation modulation,* Proceedings of the IEEE **53**, 228 (1965).

[38] K. A. S. Immink and K. Cai, *Composition check codes,* IEEE Transactions on Information Theory **64**, 249 (2018).

[39] A. Jiang, M. Schwartz, and J. Bruck, *Correcting charge-constrained errors in the rank-modulation scheme,* IEEE Transactions on Information Theory **56**, 2112 (2010).

[40] A. Mazumdar, A. Barg, and G. Zemor, *Constructions of rank modulation codes,* IEEE Transactions on Information Theory **59**, 1018 (2012).

[41] H. Zhou, M. Schwartz, A. A. Jiang, and J. Bruck, *Systematic error-correcting codes for rank modulation,* IEEE Transactions on Information Theory **61**, 17 (2014).

[42] Y. Yehezkeally and M. Schwartz, *Limited-magnitude error-correcting Gray codes for rank modulation,* IEEE Transactions on Information Theory **63**, 5774 (2017).

[43] M. Horovitz and T. Etzion, *Local rank modulation for flash memories,* IEEE Transactions on Information Theory **65**, 1705 (2018).

[44] K. A. S. Immink, *Spectrum shaping with binary DC^2-constrained channel codes,* Philips Journal of Research **40**, 40 (1985).

[45] Y. Xin and I. Fair, *Algorithms to enumerate codewords for DC^2-constrained channels,* IEEE Transactions on Information Theory **47**, 3020 (2001).

[46] T. G. Swart, J. H. Weber, and K. A. S. Immink, *Prefixless q-ary balanced codes with fast syndrome-based error correction,* IEEE Transactions on Information Theory **64**, 2431 (2018).

[47] K. A. S. Immink and J. H. Weber, *Very efficient balanced codes,* IEEE Journal on Selected Areas in Communications **28**, 188 (2010).

[48] J. H. Weber and K. A. S. Immink, *Knuth's balanced codes revisited,* IEEE Transaction on Information Theory **56**, 1673 (2010).

[49] K. A. S. Immink and J. H. Weber, *Minimum Pearson distance detection for multilevel channels with gain and/or offset mismatch,* IEEE Transactions on Information Theory **60**, 5966 (2014).

[50] M. S. Chandan and P. G. Poddar, *Minimum Pearson distance based detection for data transmission over fading wireless channels,* in *Proc. IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)* (2017) pp. 2149–2153.

[51] J. H. Weber, K. A. S. Immink, and S. R. Blackburn, *Pearson codes,* IEEE Transactions on Information Theory **62**, 131 (2016).

[52] J. H. Weber and K. A. S. Immink, *Properties of binary Pearson codes,* in *Proc. IEEE International Symposium on Information Theory and Its Applications (ISITA)* (Singapore, 2018) pp. 605–609.

[53] H. Ai, *Pearson Codes Design with Error-Control Capabilities*, Master's thesis, Delft University of Technology (2016).

[54] J. H. Weber, T. G. Swart, and K. A. S. Immink, *Simple systematic Pearson coding,* in *Proc. IEEE Internationl Symposium of Information Theory (ISIT)* (Barcelona, Spain, 2016) pp. 385–389.

[55] C. Cao and I. Fair, *Capacity-approaching variable-length Pearson codes,* IEEE Communications Letters **22**, 1310 (2018).

[56] K. A. S. Immink and J. H. Weber, *Hybrid minimum Pearson and Euclidean distance detection,* IEEE Transactions on Communications **63**, 3290 (2015).

[57] K. A. S. Immink, K. Cai, and J. H. Weber, *Dynamic threshold detection based on Pearson distance detection,* IEEE Transactions on Communications **66**, 2958 (2018).

[58] D. Chase, *Class of algorithms for decoding block codes with channel measurement information,* IEEE Transactions on Information Theory **18**, 170 (1972).

[59] K. A. S. Immink and V. Skachek, *Minimum Pearson distance detection using mass-centered codewords in the presence of unknown varying offset,* IEEE Journal on Selected Areas in Communications **34**, 2510 (2016).

[60] V. Skachek and K. A. S. Immink, *Minimum Pearson distance detection in the presence of unknown slowly varying offset,* in *Proc. IEEE International Symposium on Information Theory (ISIT),* pp. 195–199.

[61] S. R. Blackburn, *Maximum likelihood decoding for multilevel channels with gain and offset mismatch,* IEEE Transactions on Information Theory **62**, 1144 (2016).

[62] J. H. Weber and K. A. S. Immink, *Maximum likelihood decoding for Gaussian noise channels with gain or offset mismatch,* IEEE Communications Letters **22**, 1128 (2018).

[63] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding: Second Edition* (John Wiley & Sons, 2006) pp. 1–263.

[64] R. W. Hamming, *Error detecting and error correcting codes,* The Bell System Technical Journal **29**, 147 (1950).

[65] G. Bouwhuis, A. H. J. Braat, J. Pasman, G. van Rosmalen, and K. A. S. Immink, *Principles of Optical Disc Systems* (Adam Hilger, Boston, MA, USA, 1985).

[66] J. H. Weber, R. Bu, K. Cai, and K. A. S. Immink, *Binary block codes for noisy channels with unknown offset,* IEEE Transactions on Communications **68**, 3975 (2020).

[67] M. Grigoriu, *Applied Non-Gaussian Processes: Examples, Theory, Simulation, Linear Random Vibration, and MATLAB Solutions* (PTR Prentice Hall, Englewood Cliffs, NJ, 1995).

[68] A. D'Onofrio, *Bounded Noises in Physics, Biology, and Engineering* (Springer, New York, NY, 2013).

[69] I. S. Reed and G. Solomon, *Polynomial codes over certain finite fields,* Journal of the society for industrial and applied mathematics **8**, 300 (1960).

[70] K. Cai and K. A. S. Immink, *Cascaded channel model, analysis, and hybrid decoding for spin-torque transfer magnetic random access memory,* IEEE Transactions on Magnetics **53**, 1 (2017).

[71] E. Berlekamp, *Nonbinary BCH decoding,* IEEE Transactions on Information Theory **14**, 242 (1968).

[72] J. Massey, *Shift-register synthesis and BCH decoding,* IEEE Transactions on Information Theory **15**, 122 (2003).

[73] C. Argon and S. W. McLaughlin, *An efficient chase decoder for turbo product codes,* IEEE Transactions on Communications **52**, 896 (2004).

[74] H. Pozidis, N. Papandreou, A. Sebastian, T. Mittelholzer, M. BrightSky, C. Lam, and E. Eleftheriou, *A framework for reliability assessment in multilevel phase-change memory,* in *IEEE International Memory Workshop* (2012) pp. 143–146.

[75] X. Kou, J. Schmalhorst, A. Thomas, and G. Reiss, *Temperature dependence of the resistance of magnetic tunnel junctions with MgO barrier,* Applied physics letters **88**, 212115 (2006).

[76] K. Cai and K. A. S. Immink, *Detection of noisy and corrupted data using clustering techniques,* in *Proc. IEEE International Symposium on Information Theory and Its Applications (ISITA)* (2018) pp. 135–138.

[77] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas* (Princeton university press, 2009).

[78] R. Svitek and S. Raman, *DC offsets in direct-conversion receivers: characterization and implications,* IEEE Microwave Magazine **6**, 76 (2005).

[79] E. Biglieri, G. Caire, and G. Taricco, *Coding for the fading channel: A survey,* Signal Processing **80**, 1135 (2000).

[80] K. W. Cattermole, *Principles of digital line coding,* IEEE Journal on Electronics **55**, 3 (1983).

[81] H. Pozidis, T. Mittelholzer, N. Papandreou, T. Parnell, and M. Stanisavljevic, *Phase change memory reliability: A signal processing and coding perspective,* IEEE Translation Journal on Magnetics **51**, 1 (2015).

[82] L. Bharath and P. G. Poddar, *Minimum Pearson distance based detection for MIMO systems,* Procedia Computer Science **171**, 2195 (2020).

[83] P. Flajolet and R. Sedgewick, *Analytic Combinatorics* (Cambridge University Press, 2009).

[84] R. Bu and J. H. Weber, *Minimum Pearson distance detection using a difference operator in the presence of unknown varying offset,* IEEE Communications Letters **23**, 1115 (2019).

[85] Y. Saitoh and H. Imai, *Multiple undirectional byte error-correcting codes,* IEEE Transactions on Information Theory **37**, 903 (1991).

[86] Y. Saitoh and H. Imai, *Some codes for correcting and detecting unidirectional byte errors,* IEEE Transactions on Computers **42**, 547 (1993).

[87] L. Pezza, L. G. Tallini, and B. Bose, *Variable length unordered codes,* IEEE Transactions on Information Theory **58**, 548 (2012).

Curriculum Vitæ

Renfei BU

Renfei Bu was born in Changsha, Hunan, China in 1993. She received a B.Sc. degree in communication engineering from Hunan University, Changsha, China in 2015. After that she received an M.Sc. degree in information and communication engineering from the National University of Defense Technology, Changsha, China under the supervision of Prof. Shilian Wang in 2017. She has been involved in many research on underwater acoustic sensor networks. In October 2017, she joined the Department of Applied Mathematics at the Faculty of Electrical Engineering, Mathematics, Computer Science at Delft University of Technology as a Ph.D. student under the supervision of Prof. K. I. Aardal and Dr. J. H. Weber. Her research interests include signal processing and channel coding.

Publications

12. **R. Bu** and J. H. Weber, *Decoding of concatenated codes for noisy channels with unknown offset*, Proc. of the Fourty-first WIC Symposium on Information Theory and Signal Processing in Benelux, the Netherlands, May, 2021.

11. **R. Bu**, J. H. Weber and K. A. S. Immink, *Maximum likelihood decoding for channels with Gaussian noise and signal dependent offset*, IEEE Transactions on Communications, vol. 69, pp. 85-93, Sep., 2020.

10. **R. Bu**, J. H. Weber and K. A. S. Immink, *Maximum likelihood decoding for channels with uniform noise and signal dependent offset*, Proc. of IEEE International Symposium on Information Theory, August, 2020.

9. J. H. Weber, **R. Bu**, K. Cai and K. A. S. Immink, *Binary block codes for noisy channels with unknown offset*, IEEE Transactions on Communications, vol. 48, pp. 3975-3983, Apr., 2020.

8. **R. Bu** and J. H. Weber, *Minimum Pearson distance detection using a difference operator in the presence of unknown varying offset*, IEEE Communications Letters, vol. 23, pp. 1115-1118, July, 2019.

7. **R. Bu** and J. H. Weber, *Comparisons of detection schemes for channels with unknown varying offset*, Proc. of the Eleventh Asia-Europe Workshop on Concepts in Information Theory, Rotterdam, the Netherlands, July, 2019.

6. **R. Bu** and J. H. Weber, *Maximum likelihood decoding for multi-level cell memories with scaling and offset mismatch*, Proc. of IEEE International Conference on Communications (ICC), Shanghai, China, May, 2019.

5. **R. Bu** and J. H. Weber, *Decoding criteria and zero WER analysis for channels with bounded noise and offset*, Proc. of IEEE International Conference on Communications (ICC), Shanghai, China, May, 2019.

4. **R. Bu** and J. H. Weber, *Maximum likelihood decoding for channels with uniform noise and offset*, Proc. of the Thirty-ninth WIC Symposium on Information Theory and Signal Processing in Benelux, Enschede, the Netherlands, May, 2018.

3. **R. Bu**, S. Wang and H. Wang, *Fuzzy logic vector–based forwarding routing protocol for underwater acoustic sensor networks*, Transactions on Emerging Telecommunications Technologies, vol. 29, no. 3, e3252, March, 2018.

2. H. Wang, S. Wang, **R. Bu** and E. Zhang, *A novel cross-layer routing protocol based on network coding for underwater sensor networks*, Sensors, vol. 17, no. 8, p. 1821, August, 2017.

1. **R. Bu**, S. Wang and X. Yu, *Blind channel estimation and phase ambiguity elimination in MIMO-OFDM UWA communications*, Proc. of IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Hong Kong, China, August, 2016.