On Combined Coding and Modulation

Von der Fakultät für Ingenieurwissenschaften der

Universität Duisburg-Essen

Abteilung Informatik und Angewandte Kognitionswissenschaft

zur Erlangung des akademischen Grades eines

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Anıl Mengi

aus Istanbul, Türkei

1. Gutachter: Prof. Dr.-Ing. A. J. Han Vinck

2. Gutachter: Prof. Dr.-Ing. John B. Anderson

Tag der mündlichen Prüfung: 21.05.2010

Acknowledgments

This thesis describes the results of my research carried out at the Institute for Experimental Mathematics (IEM), University of Duisburg-Essen, Germany, in the period from April 2005 till February 2010. There are many people who have supported and encouraged me during these years. I would like to take this opportunity to especially acknowledge the following people for their contributions, advice and friendship.

First of all, I would like to express my deep gratitude to my Doktorvater, Prof. Dr.-Ing. A. J. Han Vinck, who made this dissertation come into being. I am indebted for his constant assistance, guidance, encouragement and the tremendous support and opportunities he provided throughout my doctoral studies. He spent countless hours proofreading my research papers, discussing my research, evaluating contemporary topics for me to investigate, and providing me with countless ways to improve upon my dissertation. I thank him for encouraging me to participate and present our research papers in international conferences, which not only improved my confidence but also gave me an opportunity to get feedback from the leading researchers in the world. I immensely enjoyed and learned from every meeting I had with him, where, besides research, we discussed a variety of things in life.

Next, I am deeply grateful to Prof. Dr. Trung van Tran for his constant support, encouragement and being always there for discussions. A special thank goes to Prof. Dr. John B. Anderson for reviewing the previous version of this dissertation and providing valuable comments.

Every colleague at the Institute for Experimental Mathematics has supported this thesis in many ways. I would like to thank Marjan Bazrafshan, Dr. Vladimir Balakirsky, Dr. Yanling Chen, Pavol Svaba, Birgit Rieth, Anahit Ghazaryan and Dr. Ingo Janiszczak for creating a very nice working atmosphere and supporting my work at all times. Special thanks go out to Lukas Maas and Nihad Cosic with whom I had daily non-research discussions and fun.

I am grateful to Prof. Dr. Hendrik Ferreira and his research group at the University of Johannesburg, South Africa for their great hospitality and helpful comments.

Last but not least, I would like to express my heartfelt gratitude to my family for their unconditional love, to Pola Richter for her continuous support and to Dj. Ahmet Sisman for his close friendship.

Contents

Chapter 1

Introduction

The objective of the digital communications systems is to design a reliable and secure transmission of digital information over a noisy channel. The channel might be any physical medium, such as air, optical fiber, powerline or optical disks. Information sent through the channel has a source from which the information originates, and a destination to which the information is delivered. Typically, errors occur during the transmission. In order to reduce the error probability, the sequence of binary digits from the source is passed to the channel encoder. The channel encoder adds redundancy to the bit sequence in a controlled way, such that errors can be detected and corrected by a receiver. On average, q information bits are translated into n channel symbols. Then, a digital modulator is used at the output of the channel encoder, which serves as an interface to the communication channels. The modulator produces signal waveforms that carry information. A simple example may be given by on-off keying, where the presence of a carrier for a channel symbol duration represents a binary one, while its absence for the same duration represents a binary zero.

Throughout this thesis, the digital data are considered as a sequence of symbols in time. The transmission time of the sequence of channel symbols is written as T. Then, the information rate can be defined as q/T, i.e., bit per transmission time. In the treatment of channel coding as a separate operation independent of the modulation, the coded set of sequences generally has a smaller channel symbol duration than the uncoded set of sequences for the same information rate. Accordingly, the power spectrum density (PSD) of the channel signals changes essentially. On the other hand, if the modulation is designed in conjunction with the channel coding, error correction can be achieved without leading to any essential changes in the PSD, see [1–4].

In this thesis, two combined coding and modulation schemes are studied. Narrowband powerline communication (PLC) is considered as a practical application. Selection of modulation and coding schemes for narrowband band PLC is a difficult problem. According to the CENELEC standards [5], the frequency allowed for communication ranges from 3 to 148.5 kHz, and the maximum allowed transmitter output voltage is specified as follows. The maximum allowed peak voltage at the frequency band from 9 to 95 kHz is equal to 5 V for the broad-band transmitters (i.e., a 20-dB bandwidth of more than 5 kHz in width). For the frequency band 95 to 148.5 kHz, it should not exceed 0.63 V. As a consequence, the transmitters are output voltage limited and bandwidth limited. Modulation schemes, such as frequency shift keying (FSK) and orthogonal frequency division multiplexing (OFDM)

are currently considered as appropriate choices for the applications in the narrowband PLC, see [6]. Depending on the target application, each modulation technique has certain advantages. The FSK modulation is suited for the CENELEC standards, where the transmission of only one frequency per time unit leads to a constant envelope signal, and it is an attractive solution for the implementations that are oriented to low data rates. However, narrowband noise brings a serious difficulty to the system design. On the other hand, the OFDM modulation is a promising choice, because it provides high data rates, spectral efficiency and frequency diversity, which is effective against the narrowband noise. However, impulsive noise leads to a significant performance loss.

The thesis can be divided into two parts. In the first part, Chapters 3 and 4, combined coding and modulation scheme based on the single carrier modulation is proposed. An RLL encoder is introduced between the channel encoder and the constant envelope modulator to control the minimum channel symbol duration (the minimum duration in which the channel symbol stays constant) of a set of block waveforms defined in a constant time. As a single carrier modulation, noncoherent FSK and PSK are considered. Accordingly, it is shown that high coding gains can be achieved at the same information rate without leading to an essential change in the PSD. The maximum-likelihood (ML) receiver structures are derived and investigated for the additive white Gaussian noise (AWGN) and the impulsive noise channel models.

In the second part of the thesis, Chapters 5 and 6, OFDM modulation is considered. If the discrete Fourier transform (DFT) of the transmitted OFDM symbol contains a small number of zeros or known data, there is a similarity between the inverse DFT (IDFT) and RS encoder. In practice, not all subcarriers are used to carry information. Some subcarriers are set to zero or known data (pilot symbols) for different purposes, that include channel estimation, synchronization or cancelation of the DC value. An iterative impulsive noise suppression algorithm is proposed, which exploits the impulsive noise structure in the time and frequency domain and uses the existing redundancy to decode the errors. The simulation results show that the influence of impulsive noise can be essentially reduced.

1.1 Thesis Outline

The outline of this thesis is as follows.

Chapter 2 Channel Models

The channel models required for the next chapters are introduced in this chapter. The AWGN and impulsive noise channel models are studied.

Chapter 3 RLL Coded Modulation with a Constraint on the Minimum Channel Symbol Duration

In this chapter, a combined coding and modulation scheme based on single carrier modulation is presented. After analyzing the power spectrum characteristics of the channel signals, the evaluation of the code performance and the receiver design for the AWGN channel are

studied. It is shown that large coding gains can be achieved by a serial concatenation of Reed-Solomon (RS) code with an RLL code. Following a discussion of a non-binary RLL coded modulation scheme, ML block receiver for the noncoherent FSK transmission over the impulsive noise channel is derived and investigated.

Chapter 4 Runlength-Limited Code Construction Based on Minimum Hamming Distance

In this chapter, we describe construction algorithms for RLL codes with one-bit look-ahead encoding techniques having a minimum Hamming distance larger than one. The codes enable soft information to be available at the RLL decoder. Hence, a soft decoding can be used to improve the error rate performance of the system. We present several new high rate RLL codes and provide efficient bounds on the code sizes. The constructed codes are used in the combined coding and modulation scheme, which is described in Chapter 3.

Chapter 5 OFDM Modulation for Impulsive Noise Error Correction

In this chapter, we show that if the DFT of the transmitted OFDM symbol contains some redundant data, such as consecutive zeros or pilot symbols, in a circular way, there exists a similarity between the IDFT and Reed-Solomon encoder. We introduce a scheme for correcting the impulsive noise errors by first locating the possible impulsive noise errors by a threshold detector and then decoding the identified erasures based on the least-squares estimation.

Chapter 6 Iterative Impulsive Noise Suppression in OFDM

The analysis in Chapter 5 shows that the performance of the least-squares estimator receiver is limited. Therefore, an iterative impulsive noise decoding scheme is proposed in this chapter. The iterative scheme is an extension of the iterative noise suppression algorithm given in [15]. The extensions include 1) a clipping and nulling technique at the input of the iterative algorithm 2) a novel low complexity syndrome decoder which uses the redundancy that is transmitted for synchronization or other purposes. The decoder applies a successive impulsive noise error decoding instead of decoding all errors jointly. This approach has the advantage that the non-consecutive zeros or pilot tones can also be used in the decoding. Simulation results are provided to show the improvement in error rate.

Chapter 7 Summary

A summary of all investigated results can be found in this chapter.

Remark: The results in Chapters 3, 4, 5 and 6 have been published in [36–41].

Chapter 2

Channel Models

In this chapter, we introduce the basic channel models required for the next chapters.

2.1 Additive White Gaussian Noise Channel Model

A communication system affected by thermal noise is often modeled as an AWGN channel. In this model, the only impairment is a linear addition of white noise with a constant spectral density. AWGN may also be caused by other interferences, and due to the central limit theorem [7], a large amount of summarized random variables results in Gaussian distribution. The discrete time memoryless AWGN channel is depicted in Fig. 2.1. At time instant k, the channel accepts one sample c_k and maps it onto a sample r_k. The value g_k added by the channel is independent of the channel input. It can be considered as a Gaussian distributed random variable with zero mean and variance σ_g^2. Its one dimensional probability density function (PDF) is given as

$$p(g_k) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{\left(-\frac{g_k^2}{2\sigma_g^2}\right)}. \tag{2.1}$$

The signal-to-noise (SNR) is defined by

$$SNR := \frac{\varepsilon}{N_0} = \frac{\mathbf{E}\{|c_k|^2\}}{\mathbf{E}\{|g_k|^2\}}, \tag{2.2}$$

where $\varepsilon := \mathbf{E}\{|c_k|^2\}$ is the average power of the transmitted samples, and $N_0 := \mathbf{E}\{|g_k|^2\} := 2\sigma_g^2$ is the single sided noise power spectral density.

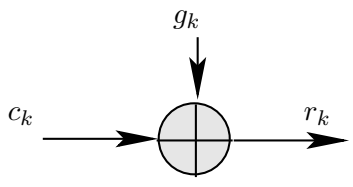

Figure 2.1: AWGN channel model.

2.2 Impulsive Noise Channel Model

In many communication channels, interference produced by man-made machinery are much more structured or impulsive than AWGN [8, 9]. In PLC, the noise includes both background and impulsive components [10]. Background noise is typically Gaussian, and its effects on the communication system are well understood. In the case of impulsive noise, however, a statistical model is needed to design a reliable communication. The impulsive noise is primarily caused by all kinds of switching operations in electrical devices [11]. By on/off switching, the capacitor in the electrical device suddenly connects/disconnects to the power circuit. It leads to large transient voltages on the circuit depending on the size of the capacitor. The generated impulses have durations of some microseconds up to a few milliseconds with random occurrence.

Within this thesis, the Middleton's additive white class A noise (AWCN) model is considered as a statistical model for the impulsive noise, see [8, 9]. In this model, Middleton classifies the impulsive noise into three general classes, i.e., class A, B and C. Among these three classes, class A is often used to model PLC noise environment [12,15]. In the class A model, the noise bandwidth has approximately the same size as the receiver's bandwidth, and so transient effects in the analog receiver stages can be ignored. The model covers both the impulsive and background noise. At the output of the channel, the received sample r_k at time instant k is given by

$$\begin{aligned} r_k &= c_k + i_k + g_k, && (2.3)\\ &= c_k + w_k, && (2.4) \end{aligned}$$

where c_k is the transmitted sample, i_k and g_k are the complex impulsive and Gaussian background noise samples, and w_k represents the complex channel noise sample that consists of the impulsive noise and the background noise samples. In particular, g_k's are independent and identically distributed (i.i.d.) complex Gaussian random variables with variance σ_g^2 and with PDF for the real variables given in Eq. (2.1). The complex impulsive noise samples i_k's are also i.i.d. with variance σ_i^2. The ratio between the variance of the Gaussian background noise and the variance of the impulsive noise is defined as

$$\Gamma = \frac{\sigma_g^2}{\sigma_i^2}. \tag{2.5}$$

The discrete time memoryless AWCN channel model is depicted in Fig. 2.2. The model can be interpreted as an infinite number of parallel AWGN channels each with variance

$$\sigma_{s_k}^2 = \sigma_g^2 + \frac{\sigma_i^2 s_k}{A}, \tag{2.6}$$

where s_k is the channel state indicating which channel is selected at time instant k, and A is the impulsive noise index. Since the channel is memoryless, the channel state s_k is an independent random variable not depending on any previous or succeeding states. Each channel at time instant k is selected with the Poisson distribution

$$P(s_k) = e^{-A}\frac{A^{s_k}}{s_k!}, \tag{2.7}$$

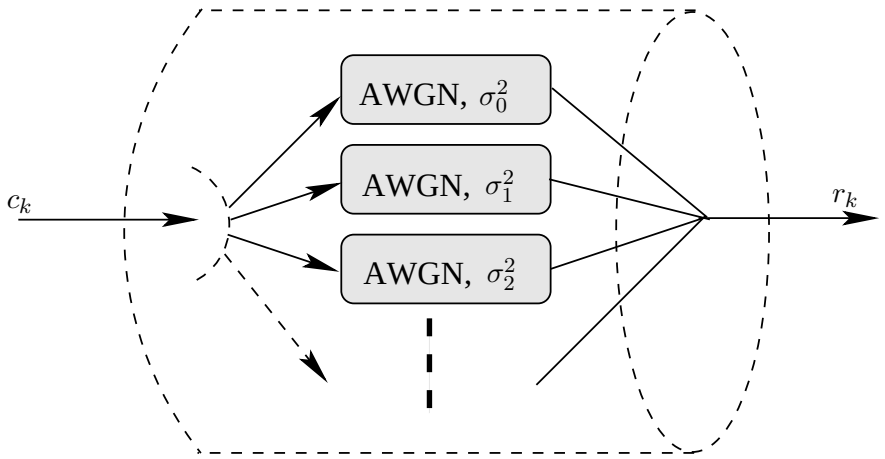

Figure 2.2: AWCN channel model.

where $s_k \geq 0$. In the case of $s_k = 0$, only background noise is present with a probability $P(0) = e^{-A}$ and variance $\sigma_0^2 = \sigma_g^2$.

The impulsive noise index A is given by the product of the average number of impulses per unit time and the mean duration of the emitted impulses entering the receiver. For small A, the noise has an impulsive character. For an impulse duration of 10^{-4} seconds and on average 100 impulses per second, we have $A = 10^{-2}$. Table 2.1 shows the three channel scenarios of the AWCN channel model. It can be seen that the channel is heavily disturbed for $A = 0.1$, since $1 - e^{-A} \approx 9.5\%$ of the samples are hit by the impulsive noise.

Table 2.1: The three channel scenarios depending on the impulsive noise index A.

Channel	A	$P(s_k > 0) = 1 - e^{-A}$
heavily disturbed	0.1	0.0952
medium disturbed	0.05	0.0488
weakly disturbed	0.01	0.01

Fig. 2.2 indicates that the PDF of the complex noise sample w_k is given by a weighted sum of the infinitely many Gaussian PDFs with increasing variance $\sigma_{s_k}^2$ and probability $P(s_k)$. Hence, the PDF can be written as

$$\begin{aligned} p(w_k) &= \sum_{s_k=0}^{\infty} P(s_k) p(w_k|s_k) \\ &= \sum_{s_k=0}^{\infty} \frac{e^{-A} A^{s_k}}{s_k!} \frac{1}{2\pi\sigma_{s_k}^2} e^{\left(-\frac{|w_k|^2}{2\sigma_{s_k}^2}\right)}. \end{aligned} \tag{2.8}$$

It can be observed in Eq. (2.8) that the real and the imaginary parts of the complex noise sample w_k are statistically dependent. This is due to the fact that at each time instant k, one channel with a complex Gaussian PDF $p(w_k|s_k)$ is selected. Thus, the real and the

imaginary parts of w_k have the same variance. On the other hand, the complex PDF of w_k is not equal to the product of the real and imaginary parts PDF's. Therefore, both parts are uncorrelated, but dependent, see [13].

Table 2.2 shows the calculated probabilities and channel noise variances for the first three states of AWCN channel model. It can be observed that as s_k increases, the probability $P(s_k)$ of selecting the s_kth channel decreases rapidly. Hence, truncating the infinite sum of Eq. (2.8) to a small number of states only leads to a small error.

Table 2.2: The probability $P(s_k)$ and the channel noise variance $\sigma^2_{s_k}$ for $A = 0.01$ and $\Gamma = 0.01$.

Channel state m	$P(s_k) = e^{-A}\frac{A^{s_k}}{s_k!}$	$\sigma^2_{s_k} = \frac{\sigma_i^2 s_k}{A} + \sigma_g^2$
$s_k = 0$	0.99	σ_g^2
$s_k = 1$	0.0099	$10^4\sigma_g^2 + \sigma_g^2$
$s_k = 2$	4.9502×10^{-5}	$2 \times 10^4\sigma_g^2 + \sigma_g^2$

If the channel state s_k is known at the receiver, then the additive channel noise sample w_k is a complex Gaussian distributed random variable with its PDF conditioned on the channel state

$$p(w_k|s_k) = \frac{1}{2\pi\sigma^2_{s_k}} e^{\left(-\frac{|w_k|^2}{2\sigma^2_{s_k}}\right)}. \tag{2.9}$$

Applying Eq. (2.4) yields the channel transition PDF as

$$p(w_k = r_k - c_k|s_k) = \frac{1}{2\pi\sigma^2_{s_k}} e^{\left(-\frac{|r_k - c_k|^2}{2\sigma^2_{s_k}}\right)}. \tag{2.10}$$

Equations (2.8) indicate that the error probability P_e for an uncoded transmission over the AWCN channel is a weighted sum of the error probabilities of the uncoded transmissions over the AWGN channel with variance $\sigma^2_{s_k}$, i.e.,

$$P_e = \sum_{s_k=0}^{\infty} P(s_k) P_e^{AWGN}(\sigma^2_{s_k}), \tag{2.11}$$

where $P_e^{AWGN}(\sigma^2_{s_k})$ is the error probability for the uncoded transmission over the AWGN channel with variance $\sigma^2_{s_k}$. By calculating the average noise variance σ_w^2 of the AWCN model as

$$\sigma_w^2 = \mathbf{E}\{\sigma^2_{s_k}\} = \sigma_g^2 + \frac{\sigma_i^2}{A}\mathrm{E}\{s_k\}, \tag{2.12}$$

the expected error probability $\mathbf{E}\{P_e\}$ for an uncoded transmission over the AWCN channel can be written as

$$\mathbf{E}\{P_e\} = P_e^{AWGN}(\sigma^2_{w_k}). \tag{2.13}$$

2.2.1 Two-state Impulsive Noise Channel Model

The AWCN channel model can be simplified to two-state impulsive noise channel model by considering only two states, namely $s_k = 0$ and $s_k = 1$. The model is shown in Fig. 2.3. For $s_k = 0$, only background noise is present with probability $P(s_k = 0) = e^{-A} \approx 1 - A$ and variance $\sigma_0^2 = \sigma_g^2$. For $s_k = 1$, the background noise and impulsive noise are together present with variance $\sigma_1^2 = \sigma_i^2/A + \sigma_g^2$. The probability of having $s_k = 1$ can be calculated as $P(s_k = 1) = 1 - e^{-A} \approx A$. The PDF of the channel noise sample w_k is given by

$$p(w_k) = (1 - A)\frac{1}{2\pi\sigma_0^2}e^{\left(-\frac{|w_k|^2}{2\sigma_0^2}\right)} + A\frac{1}{2\pi\sigma_1^2}e^{\left(-\frac{|w_k|^2}{2\sigma_1^2}\right)}. \tag{2.14}$$

If we consider binary modulation signals, the probability of error for binary phase shift keying (BPSK) is then given by

$$P_e = (1 - A)Q\left(\sqrt{\frac{E_b}{\sigma_0^2}}\right) + AQ\left(\sqrt{\frac{E_b}{\sigma_1^2}}\right), \tag{2.15}$$

where E_b is the transmitted signal energy per bit. For binary FSK, an additional loss of 3 dB should be included, additionally.

The average noise variance σ_w^2 of the two-state impulsive noise channel model is calculated as

$$\begin{aligned} \sigma_w^2 &= \mathrm{E}\{\sigma_{s_k}^2\} = \sigma_g^2 + \frac{\sigma_i^2}{A}A, \\ &= \sigma_g^2 + \sigma_i^2, \end{aligned} \tag{2.16}$$

where we expect A impulses ($s_k = 1$), on the average. Thus, the SNR of the two-state impulsive noise channel model is defined by

$$SNR := \frac{\mathrm{E}\{|c_k|^2\}}{2\sigma_w^2} = \frac{\mathrm{E}\{|c_k|^2\}}{2(\sigma_g^2 + \sigma_i^2)}. \tag{2.17}$$

In some communication channels, such as in PLC channel, the impulsive noise variance σ_i^2 may not be dependent on the background noise variance. Hence, instead of using the ratio

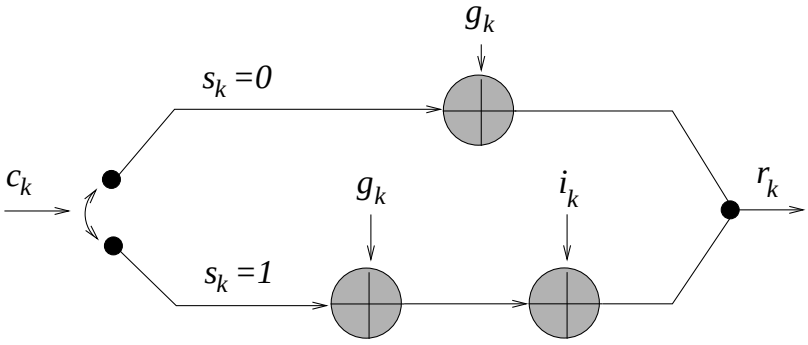

Figure 2.3: Two-state impulsive noise channel model

Γ in Eq. (2.5), we can also relate the impulsive noise power to the signal power. In that case, the ratio between the variance of the signal and the variance of the impulsive noise can be defined as

$$\Delta = \sigma_c^2/\sigma_i^2. \tag{2.18}$$

For $\Delta = 10^{-3}$, the impulsive noise variance is 30 dB larger than the signal variance.

Using the average signal-to-impulsive noise power ratio Δ, Eq. (2.15) can be rewritten as

$$P_e = (1-A)Q\left(\sqrt{2\text{SBNR}}\right) + AQ\left(\sqrt{A\Delta\frac{2\text{SBNR}}{\text{SBNR}+A\Delta}}\right), \tag{2.19}$$

where SBNR=$E_b/2\sigma_g^2$ is the signal-to-Gaussian background noise ratio. In Eq. (2.19), for small values of SBNR, the first term is dominant. Then, with increasing SBNR, the first term decreases, and the second term starts to dominate the error probability depending on the factors A and Δ.

Chapter 3

RLL Coded Modulation with a Constraint on the Minimum Channel Symbol Duration

Coding for bandwidth limited channels has been a very dynamic research area, since multilevel coding and trellis-coded modulation were introduced, see [1–4]. The principle of coded modulation is the interpretation of coding and modulation as a single entity to save transmitter power on a data-bit basis without changing the minimum channel symbol duration (the minimum duration in which the channel symbol stays constant) as it was shown by Ungerboeck [1]. The scheme in [1] is based on the set partitioning principles. The key point is finding subsets of the constellation in such a way that the signal points inside each partition are maximally separated. With $M = 2^m$ different signal points one can transmit m information bits per channel symbol without coding. In trellis coded modulation $(m-1)$ information bits are transmitted, and a $(m-1)/m$ trellis code is used to generate sequences with high Euclidean distances. The uncoded system with constellation size of $2^{(m-1)}$ is used as a reference system transmitting at the same information rate. Note that the minimum channel symbol duration in the uncoded system is kept the same as for the coded system.

In this chapter, we propose to combine the design of coding and modulation by controlling the minimum channel symbol duration of a set of block waveforms defined in a constant time. We show that high coding gains can be achieved at the same information rate without leading to an essential change in the PSD of the channel signals. As a practical application, we consider PLC channel, where there is a limitation on the output voltage and bandwidth.

The rest of the chapter is organized as follows. After a brief review of the basics of RLL codes, the look-ahead encoding technique is explained in Subsection 3.1.2. In section 3.2,

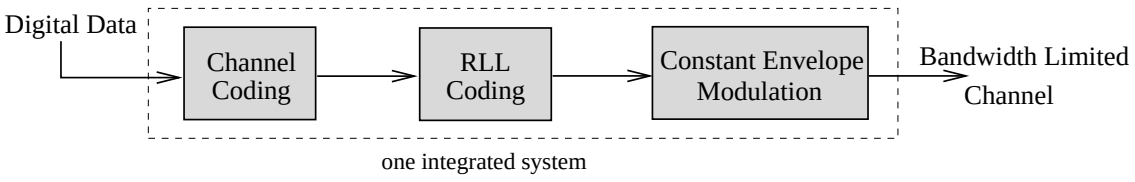

Figure 3.1: Block diagram of the coded modulation scheme based on RLL encoding.

the combined coding and modulation scheme based on RLL coding is presented. This is followed by the power spectral characteristics of channel signals. Section 3.4 illustrates the communication system under investigation and the design of the ML receiver for the AWGN channel. The evaluation of the code performance is also presented in this section. In Section 3.5, the concatenation of the RS code with the RLL code is studied. Following a discussion of an M-level RLL coded modulation scheme, the ML block receiver for the noncoherent 2-FSK transmission over the AWCN channel is derived and investigated in Section 3.8. Section 3.9 addresses the influence of the narrow band noise on the ML receiver structure. Finally, Section 3.11 concludes the chapter by possible modifications and remaining problems of the RLL coded modulation scheme.

3.1 Introduction to RLL Codes

In RLL coding, the data sequences are encoded into a restricted set of sequences that comply with the prescribed channel constraints. Runlength-limited simply means that the number of symbols (runs) in the intervals, where the signal stays constant, are bounded from below by the parameter d and from above by the parameter $\widehat{k}$. These bounds are called RLL constraints. The $(d, \widehat{k})$-constraint binary RLL sequence is characterized as follows.

- d-constraint: Each run of zeros or ones must have a length of at least $d + 1$.
- $\widehat{k}$-constraint: Each run of zeros or ones must have a length of at most $\widehat{k} + 1$.

An example of a sequence satisfying the $(d, \widehat{k}) = (1, 5)$ constraint is

$$0\,0\,1\,1\,1\,1\,1\,1\,0\,0\,1\,1\,0\,0.$$

In this thesis, no upper bound on the run of zeros or ones is considered. Thus, the parameter $\widehat{k}$ is omitted in the next sections. Fig. 3.2 illustrates a possible graph representation of the d-constraint. Any path through the graph defines an allowed sequence. Each sequence containing runs of zeros or ones less than $(d + 1)$ is a forbidden sequence.
RLL code has wide applications in the field of magnetic and optical disk technology, magnetic tape and fiber optic channels. A large number of related references can be found in [19, 26].

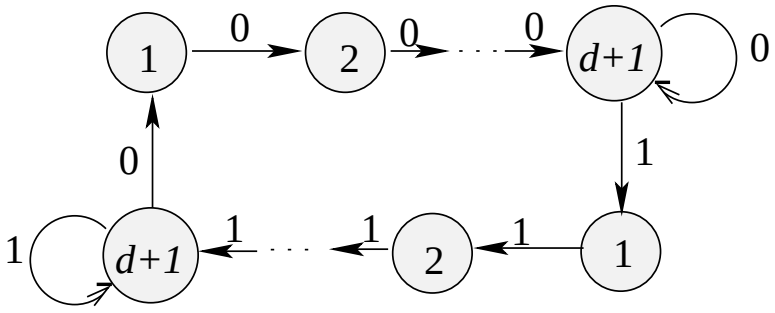

Figure 3.2: Graph representation of the d-constraint.

3.1.1 Asymptotic Information Rate of RLL Sequences

Let the source data be grouped into blocks of m symbols, and let a constraint encoder be used to translate these blocks into RLL codewords of length n symbols. The rate (efficiency) of the constraint coding is calculated as $R_r = m/n$. Shannon showed in his work [17] that the rate $R_r = m/n$ of an RLL encoder that satisfies the given d-constraint cannot exceed a quantity, which can be referred to as the asymptotic information rate of RLL sequences. It is given as

$$C_d = \lim_{n\to\infty} \frac{1}{n} \log_2 N(n,d), \tag{3.1}$$

where $N(n,d)$ is the number of RLL sequences of length n that satisfy the d-constraint. The asymptotic information rate of RLL sequences determines the number of information bits per channel bit that can maximally be carried by a constrained sequence. It was shown in [18] that the number of RLL sequences $N(n,d)$ of length $n > 0$ can be calculated by the enumeration of all sequences satisfying the d-constraint, and it is given as

$$\begin{aligned} (i)\ \ N(n,d) &= 2n, \ \ 1 \le n \le (d+2), \\ (ii)\ \ N(n,d) &= N(n-1,d) + N(n-d-1,d), \ \ n > d+2. \end{aligned} \tag{3.2}$$

Table 3.1 lists the number of RLL sequences as a function of the sequence length n and the parameter d. It can be seen that $N(n,d)$ depends on d, that is, the smaller d, the larger $N(n,d)$, and the larger information rate. Immink [26] shows that a characteristic equation of the form

$$z^{(d+1)} - z^d = 1 \tag{3.3}$$

can be obtained by assuming a solution of $N(n,d) = cz^n$ to Eq. (3.2). It also follows that if λ is the largest real root of Eq. (3.3), the maximum value of R_r that can be attained for values of d is

$$C_d = \lim_{n\to\infty} \frac{1}{n} \log_2 N(n,d) = \log_2 \lambda. \tag{3.4}$$

Example 3.1. For $d = 1$, the characteristic Eq. (3.3) can be written as $z^2 - z = 1$ with solutions $\lambda_1 = \frac{1}{2}(1+\sqrt{5})$ and $\lambda_2 = \frac{1}{2}(1-\sqrt{5})$. The largest real root is λ_1, and the $C_{d=1}$ is

$$C_{d=1} = \log_2 \frac{(1+\sqrt{5})}{2} \simeq 0.694.$$

Table 3.1: The number of RLL sequences as a function of the sequence length n and the parameter d.

$d\backslash n$	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	6	10	16	26	42	68	110	178	288	466	754	1220	1974	3194	5168
2	6	8	12	18	26	38	56	82	120	176	258	378	554	812	1190
3	6	8	10	14	20	28	38	52	72	100	138	190	262	362	500
4	6	8	10	12	16	22	30	40	52	68	90	120	160	212	280

It can be observed in Table 3.1 that the ratio $N(n,d)/n$ increases as n increases. Since the asymptotic information rate C_d can only be approached from the left side, any RLL code can be measured by the proximity of its rate to the asymptotic information rate. Table 3.2 shows C_d as a function of d.

Table 3.2: C_d as a function of d

	d=0	d=1	d=2	d=3	d=4	d=5
C_d	1	0.6942	0.5515	0.4650	0.4057	0.3620

3.1.2 Block-Decodable RLL Codes with Look-Ahead Encoding Technique

An important RLL encoding technique documented in the literature [20–25] is called the look-ahead encoding technique. A block-code is said to be look-ahead if the encoding is done as a function of not only the present and past inputs, but also of a finite number of inputs yet to come. The advantage of this type of encoding is that codes with higher rates can be constructed without a need of merging bits during the concatenation of the RLL codewords. At the decoder side, the RLL codewords can be decoded independently, which does not lead to an error propagation between the decoded codewords. In this chapter, we consider one-bit look-ahead encoding technique. The encoder makes a decision on the present word on the basis of the m-bit word itself, as well as the upcoming bit, and depending on the code construction, the last bit of the RLL codeword, see Fig. 3.3. The following example illustrates the one-bit look-ahead RLL encoder.

Example 3.2. The code table of ($d = 1$)-constraint RLL code at a rate $R_r = 3/5$ is shown in Table 3.3. It can be observed that the source words with indices {1, 2, 5, 6} are uniquely represented by a single codeword. These codewords can be concatenated freely without violating the runlength constraint, that is, solitary "ones" or "zeros" will never occur. The remaining four source words with indices {0, 3, 4, 7} are represented by two codewords. The choice depends on the first bit of the upcoming source word. If for example, the present source index is 7, and if the first bit of the upcoming source word is 1,

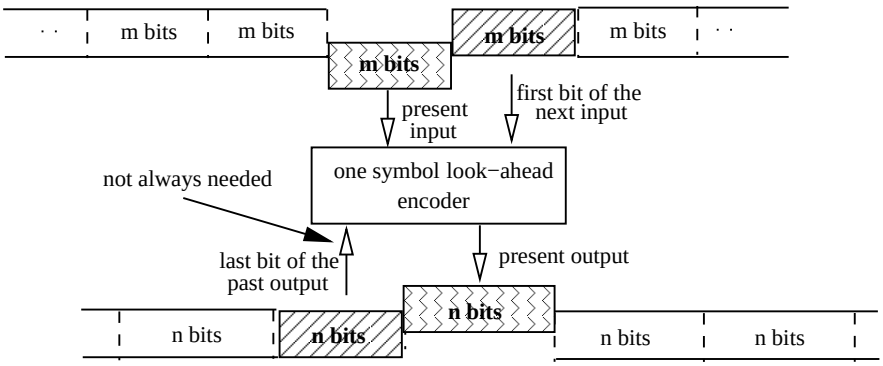

Figure 3.3: Schematic of look-ahead encoding.

then 11001 will be selected as the codeword of index 7. Thus, depending on the observed input, the encoder can decide on one alternative codeword, which maintains the $(d = 1)$-constraint in concatenation. At the receiver side, the received codewords can be decoded block-wise, which avoids the error propagation between the decoded codewords. Hence, we can conclude that the introduced code demonstrates a simple encoding and decoding structure. Note that the rate of the illustrated code is $R_r = 3/5 = 0.6 < C_{d=1} = 0.6942$ where $C_{d=1}$ is the asymptotic information rate of the $(d = 1)$-RLL code.

Table 3.3: Code table of the $R_r = 3/5$ $(d = 1)$-constraint RLL code.

		Next source bit	
index	Source word	0	1
0	000	00110	00001
1	001	00011	00011
2	010	00111	00111
3	011	01111	10011
4	100	10000	01100
5	101	11000	11000
6	110	11100	11100
7	111	11110	11001

In [20,21], Hollmann proposed a one symbol look-ahead code construction in terms of principal state-sets. The proposed construction is proved to be very efficient, and the rates of the constructed codes are close to the theoretical maximum. In his work, Hollmann described a rate 8/12, $(d = 1)$-constraint RLL code with a Hamming distance one, i.e., $d_{\min} = 1$. The code rate 8/12 is highly attractive as it is just 0.4% below the asymptotic information rate $C_{d=1}$.

In Chapter 4, we generalize the Hollmann's look-ahead RLL code construction algorithm to construct many novel block-decodable RLL codes with minimum Hamming distance larger than one. The constructed codes have several advantages in terms of the achievable code rate, minimum distance, and simple encoding and decoding structure. Table 3.4 shows some constructed RLL codes required for the rest of this chapter.

Table 3.4: Selected RLL codes

d	R_r	$d_{\min}$	C_d
1	2/3=0.66	1	0.6942
1	8/12=0.66	1	0.6942
1	6/11=0.5455	2	0.6942
1	8/14=0.5714	2	0.6942
1	12/19=0.6316	2	0.6942
1	9/18=0.5	3	0.6942
2	7/13=0.5385	1	0.5515
2	8/15=0.53	1	0.5515
2	6/13=0.4615	2	0.5515
2	8/17=0.4706	2	0.5515
2	10/21=0.4762	2	0.5515
2	6/16=0.375	3	0.5515
2	7/18=0.3839	3	0.5515
2	8/21=0.3810	3	0.5515
2	6/18=0.3333	4	0.5515
2	7/20=0.35	4	0.5515
3	3/7=0.4286	1	0.465
3	8/18=0.444	1	0.465
3	6/16=0.375	2	0.465
3	7/18=0.388	2	0.465
3	8/20=0.4	2	0.465
3	5/17=0.2941	3	0.465
3	6/19=0.3157	3	0.465
3	7/22=0.3181	3	0.465
3	5/18=0.2778	4	0.465
3	6/20=0.3	4	0.465
3	6/24=0.25	5	0.465

3.2 RLL Coded Modulation Principle

The key idea of the proposed RLL coded modulation scheme is to map the uncoded set of block waveforms into the set of RLL coded block waveforms such that the minimum Euclidean distance in the set is increased. This goal can be achieved by increasing the number of block waveforms over the corresponding uncoded system to be able to select block waveforms with desirable minimum Hamming distance. Note that the parameter $(d+1)$ is used to keep the minimum channel symbol duration (the minimum duration in which the channel symbol stays constant) the same. We describe this principle in the following.

Without loss of generality, we consider an uncoded system with BPSK modulation, where the information has a bit duration of τ seconds. We transmit q information bits in time $T = q\tau$ with a total energy qE_b, where E_b is the energy per bit. This corresponds to the set of $2^{T/\tau}$ block waveforms. The information rate is defined as bit per transmission time and written as $q/T = 1/\tau$. If a binary RLL encoder with an efficiency $R_r = q/n$ is applied, the set of all possible sequences which the RLL encoder can generate satisfies the $(d+1)$ constraint. The time interval T is partitioned into RLL symbols with duration τ' seconds. This corresponds to the set of $2^{TR_r/\tau'}$ sequences or block waveforms in a binary RLL code of length n, see Fig. 3.4. Since the comparisons are made on the basis of the same minimum channel symbol duration, we choose

$$\tau' = \frac{\tau}{(d+1)}. \tag{3.5}$$

Hence, the number of RLL sequences of length n is given by

$$2^{TR_r/\tau'} = 2^{TR_r(d+1)/\tau}. \tag{3.6}$$

Here, we can consider two cases.

(1) If we apply an RLL code with $R_r(d+1) > 1$, then the number of RLL block waveforms is larger than the number of uncoded block waveforms. Thus, we can increase the information rate, i.e., $R_r(d+1)/\tau > 1/\tau$.

(2) If we apply an RLL code with $d_{\min} > (d+1)$ and $R_r(d+1) = 1$, then coding gain can be achieved within the same minimum channel symbol duration and for the same information rate. Accordingly, the energy per RLL symbol is calculated as $E_s = E_b/(d+1)$, and the total energy in time T is equal to qE_b.
In the following sections, we consider the second case where we keep the information rate constant.

Example 3.3. In Table 3.4, we summarized some constructed RLL codes that satisfy $R_r(d+1) \geq 1$. Let us consider the $(d=1)$-constraint RLL code with $R_r = 9/18$ and $d_{\min} = 3$. Since the condition

$$2^{TR_r(d+1)/\tau} = 2^{T/\tau} \tag{3.7}$$

is satisfied for $R_r(d+1) = 1$, we can map a set of 2^9 uncoded information blocks to the set of 2^9 RLL coded blocks within the same block transmission time $T = 9\tau$ and the same minimum channel symbol duration $\tau' = \frac{\tau}{2}$. The minimum Euclidean distance is increased from $2\sqrt{E_b}$ to $\sqrt{6E_b}$. Hence, coding gain can be expected.

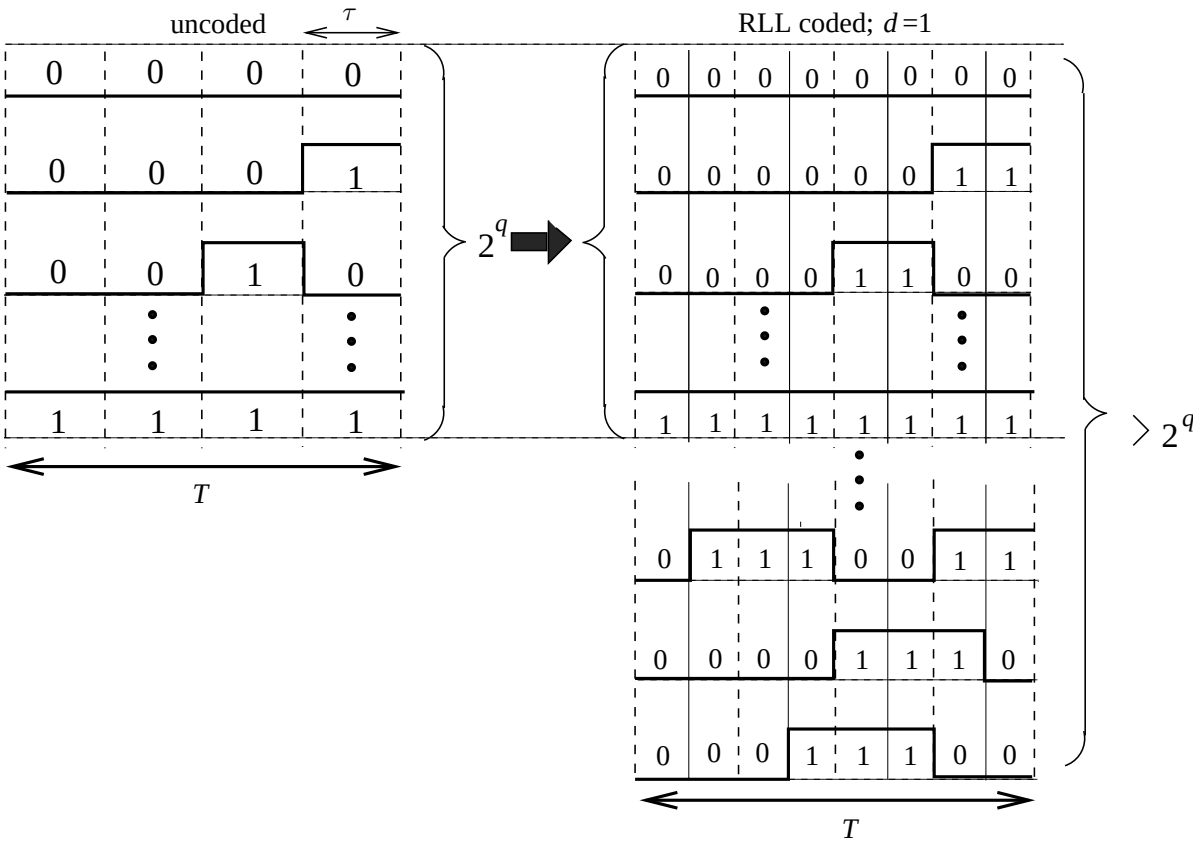

Figure 3.4: Binary waveforms of a set of uncoded and RLL coded sequences, $d = 1$.

3.3 Spectral Analysis of Transmitted Sequences

The PSD function of a signal shows the distribution of the average signal power versus frequency (power per Hz), and it offers a measure of the frequency occupation. In this section, we first study the PSD of the maxentropic (with maximum information content) RLL sequences. Later, we are concerned with the estimation of the spectral characteristics of RLL coded signals using BPSK modulation.

3.3.1 Spectral Density of Maxentropic RLL codes

RLL sequences with maximum information contents are defined as maxentropic RLL sequences. In a maxentropic RLL sequence, the runlength of length L_j has probability of occurrence [26]

$$Pr(L_j) = \lambda^{-j}, \quad j = (d+1, ...), \tag{3.8}$$

where λ is the largest real root of the characteristic equation given in Eq. (3.3). The proof of Eq. (3.8) can be found in [26]. It was shown in [42] that the calculation of the power spectra for the transmitted sequence of binary symbols, whose run-lengths $L_l \in (d+1, ...)$ are independently emitted with probability $Pr(L_l)$, is written as

$$H(w) = \frac{1}{\overline{L} sin^2 w/2} \frac{1 - |G(w)|^2}{1 + |G(w)|^2}, \tag{3.9}$$

where $G(w) = \sum_{l=d+1}^{\infty} Pr(L_l)e^{jwl}$ and $\overline{L} = \sum_{l=d+1}^{\infty} lPr(L_l)$. Truncating the infinite sum to large values only leads to minor errors. Using the run-length probability distribution for the maxentropic RLL sequences, the power spectra of baseband RLL sequences for various values of $(d+1)$ can be calculated. In Subsection 3.1.1, we showed that the rate of an RLL encoder that satisfies the given d-constraint cannot exceed a quantity, which can be referred as the asymptotic information rate of RLL sequences, C_d. If a number of $(d+1)$ code bits is transmitted in τ seconds, then the maximum information content for the RLL codes is expressed as $(d+1)C_d$ bits/τ. In magnetic recording, the quantity $(d+1)C_d$ is called as packing density.

In Fig. 3.5, we normalize the frequency scale so that the information rate stays constant, in which case the maxentropic RLL sequences with different d parameters have different minimum channel symbol duration. The power spectra of the baseband BPSK modulation is represented as $d = 0$. As d increases, the frequency occupation of the main lobe and the spectral density at zero frequency decreases. Furthermore, the peak in the spectrum increases and shifts in the direction of lower frequencies.

For the magnetic recording systems, Wolf [34] discussed how the factor $(d+1)C_d$ influences the recording density. He stated that, in the absence of noise, for increasing $(d+1)$ a larger amount of information can be stored at a minimum channel symbol duration, see Table 3.5. This observation is used in this thesis in a different way. In our case, increasing $(d+1)$ enables the RLL encoder to use an RLL code in which the Hamming distance between the RLL codewords is larger than $(d+1)$. This leads to a combined RLL encoder-modulator that establishes a correspondence between the information bits and the channel symbols

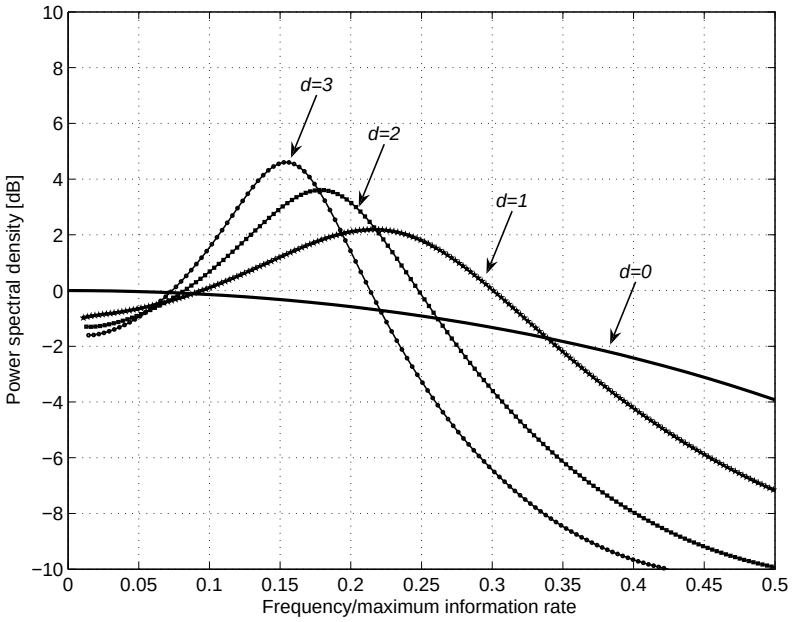

Figure 3.5: Spectral density of maxentropic RLL codes (same information rate).

without changing the minimum symbol duration. The receiver then performs a ML soft decoding at the demodulator output.

Table 3.5: C_d and $(d+1)C_d$ as a function of d

	d=0	d=1	d=2	d=3	d=4	d=5
C_d	1	0.6942	0.5515	0.4650	0.4057	0.3620
$(d+1)C_d$	1	1.388	1.654	1.860	2.028	2.172

3.3.2 Spectral Density Estimation of RLL coded modulation

We consider the estimation of the spectral characteristics of RLL coded signals by using Welch's method, see [43]. Welch's method involves the following main steps.

- First, the sequence is subdivided into successive data segments, where we allow that the data segments can overlap.
- The data segments are windowed with a specified window, such as a Hamming window, see [44]. The length of the window is chosen as the segment length.
- The periodogram is computed for each segment. It converts the signal from the time domain to the frequency domain applying the fast Fourier transform (FFT), and then computes the squared magnitude of the result.
- The expected value of the periodogram over the set of periodograms is calculated to form the estimate of the spectral density.

Fig. 3.6 compares the simulated power spectra of RLL coded BPSK modulation with the uncoded BPSK modulation for the same minimum channel symbol duration. Table 3.6 shows the parameters of the selected RLL codes with $d_{\min} = d + 2$. To ensure that each transmitted BPSK symbol contains an integral number of cycles of the carrier wave, the carrier frequency is chosen multiple of $1/\tau' = (d+1)/\tau$. We conclude from Fig. 3.6 that the PSD stays almost unchanged for the same minimum channel symbol duration.

In Fig. 3.7, we present the PSD of the uncoded 2-FSK modulation with $d = 0$ and $(d = 1)$-constraint RLL coded 2-FSK modulation for the same minimum channel symbol duration. Note that the minimum frequency separation between two frequencies for orthogonality is $(d+1)/\tau$ for noncoherent reception. It can be seen that the proposed coded FSK modulation scheme does not lead to any additional frequency occupation in PSD.

Table 3.6: Selected RLL codes for $d_{\min} = d + 2$.

d	R_r	$d_{\min}$	$R_r(d+1)$
1	9/18	3	1
3	6/24	5	1

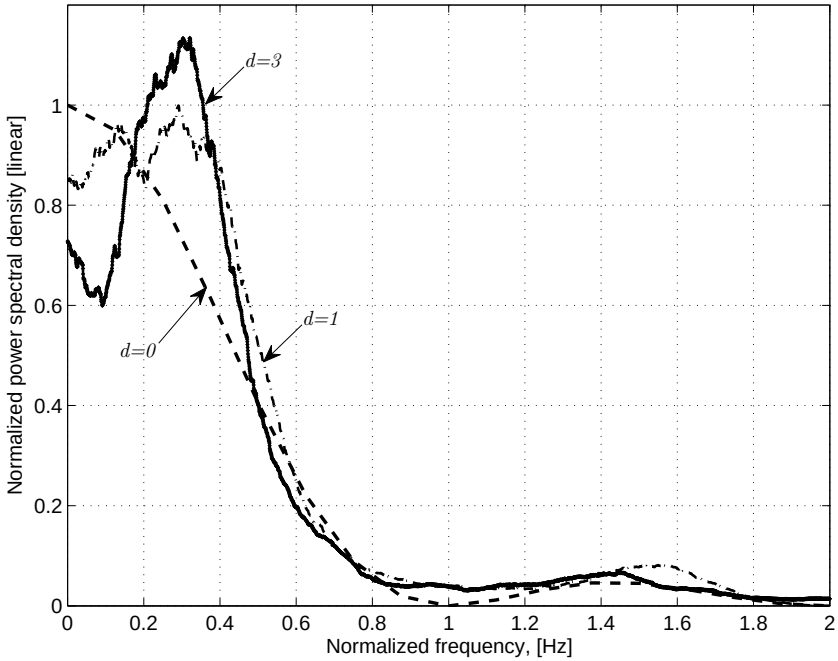

Figure 3.6: Simulated spectral density of RLL coded ($d_{\min} = d + 2$) and uncoded BPSK modulation with the same minimum channel symbol duration.

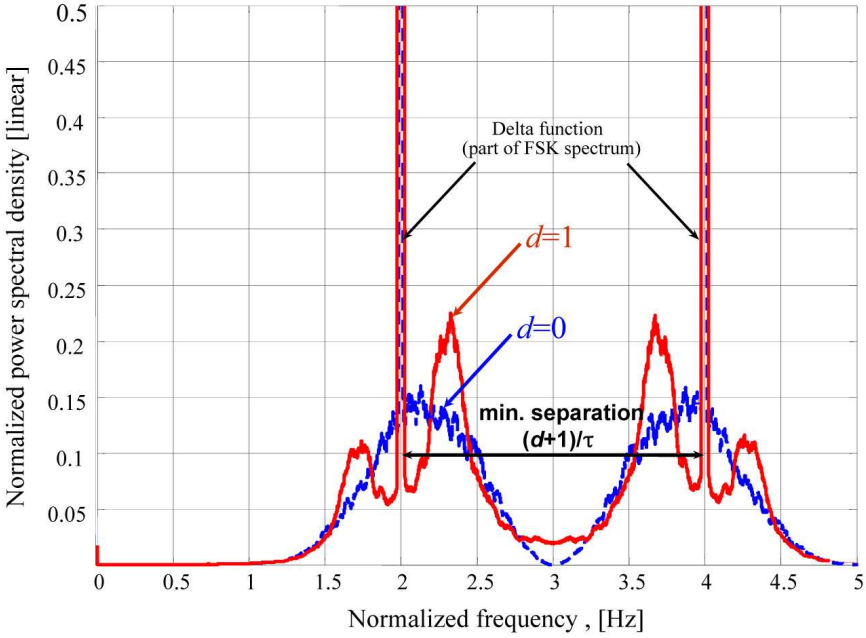

Figure 3.7: Simulated spectral density of RLL coded ($d_{\min} = d + 2$) and uncoded 2-FSK modulation with the same minimum channel symbol duration, τ is normalized to one.

3.4 Communication System

The communication system is depicted in Fig. 3.8. At the transmitter, one RLL symbol c_k is generated every τ' seconds, where the symbol length is written by $\tau' = \frac{\tau}{(d+1)}$, see Section 3.2. The generated symbols are passed through a transmit filter and sent through an AWGN channel. Throughout the thesis, we assume intersymbol interference (ISI)-free transmission over an ideal rectangular channel of bandwidth B. At the receiver, the matched-filter samples at the signaling rate $1/\tau' = (d+1)/\tau$ samples/s. If the integrate-and-dump circuit is used for the implementation of the matched filter, the integrator should be restored to its initial condition every τ' seconds. The sampling clock in the proposed RLL coded modulation scheme runs $(d+1)$ times faster than the clock in the uncoded modulation. With perfect timing and carrier-phase synchronization, the output of the sampling at time instant k can be written as

$$r_k = c_k + g_k, \tag{3.10}$$

where g_k denotes an i.i.d. Gaussian noise sample with zero mean and noise spectral density N_0. The SNR is defined by

$$SNR = \frac{E_s}{N_0} = \frac{E_b}{N_0(d+1)}, \tag{3.11}$$

where E_s is the channel symbol energy in a symbol duration τ'. Since $\tau' = \tau/(d+1)$, the information bit energy in a bit duration τ is calculated as $E_b = E_s(d+1)$.

3.4.1 The Maximum-Likelihood Receiver

The RLL encoder translates the information block into RLL codeword $\mathbf{c} \in \mathbf{C}$ of length n, where $\mathbf{c} = (c_1, ..., c_n)$. We assume that every RLL codeword is transmitted equally likely with the probability $P(\mathbf{c}) = 1/|\mathbf{C}|$, where $|\mathbf{C}|$ is the cardinality of $\mathbf{C}$. The set of all possible code symbols c_k depends on the modulation scheme adopted. For BPSK modulation, we obtain $c_k \in \{-1, +1\}$. After the transmission over the AWGN channel, soft-decision decoding can be applied. It means that the decoder operates on unquantized soft output samples $\mathbf{r} = (r_1, ..., r_n)$ of the matched filter. We wish to design a receiver that maximizes the conditional PDF $p(r_1, ..., r_n|c)$. Such a receiver is called the ML receiver.

Since the channel noise is assumed to be white and Gaussian with the PDF $p(r_k|c_k) = p(g_k)$ as given by Eq. (2.1), the joint PDF of $p(r_1, ..., r_n|c)$ may be expressed as a product of n marginal PDFs, i.e.,

$$p(r_1, ..., r_n|c) \quad = \quad \prod_{k=1}^{n} p(r_k|c_k), \tag{3.12}$$

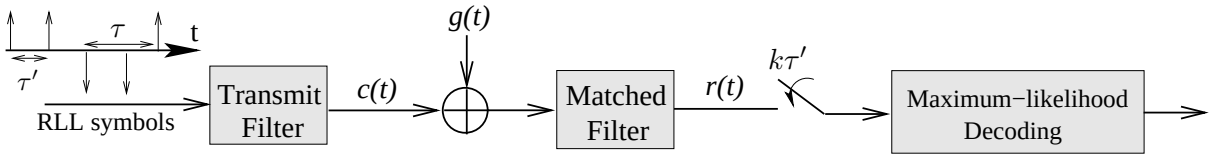

Figure 3.8: Communication system for the RLL coded BPSK modulation.

$$= \prod_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_g} e^{\left(-\frac{(r_k-c_k)^2}{2\sigma_g^2}\right)}, \tag{3.13}$$

$$= \left(\frac{1}{\sqrt{2\pi}\sigma_g}\right)^n e^{\left(-\sum_{k=1}^{n}\frac{(r_k-c_k)^2}{2\sigma_g^2}\right)}. \tag{3.14}$$

By taking the logarithm and neglecting all the factors independent of $\mathbf{c}$, the ML decoding rule for BPSK modulation can be written as

$$\max_{\mathbf{c}\in\mathbf{C}} p(\mathbf{r}|\mathbf{c}) = \min_{\mathbf{c}\in\mathbf{C}} \sum_{k=1}^{n} (r_k - c_k)^2. \tag{3.15}$$

In Eq. (3.15), $\sum_{k=1}^{n}(r_k - c_k)^2$ is the Euclidean square distance between the vectors $\mathbf{r}$ and $\mathbf{c}$. In other words, the ML decoder finds the closest signal to $\mathbf{r}$ in Euclidean space.

3.4.2 RLL Codeword Error Probability

The performance of the RLL coded modulation scheme is evaluated with respect to the RLL codeword error probability, P_{RLL}. It is defined as the probability $P_{RLL} = P(\mathbf{c}' \neq \mathbf{c})$ that the decoded RLL codeword $\mathbf{c}'$ is different from the transmitted codeword $\mathbf{c}$. In general, exactly calculating P_{RLL} is a difficult task. Hence, the union bound is employed which yields an asymptotically tight upper bound on P_{RLL}, and it is given as

$$P_{RLL} := P(\mathbf{c}' \neq \mathbf{c}) \leq \sum_{\mathbf{c}\in\mathbf{C}} P(\mathbf{c}) \sum_{\substack{\mathbf{c}'\in\mathbf{C} \\ \mathbf{c}'\neq\mathbf{c}}} P(\mathbf{c}\rightarrow\mathbf{c}'), \tag{3.16}$$

where $P(\mathbf{c})$ is the probability that the RLL codeword $\mathbf{c}$ is transmitted, and $P(\mathbf{c}\rightarrow\mathbf{c}')$ is the pairwise error probability (PEP). The PEP is the probability that, in the decision between two RLL codewords $\mathbf{c}\epsilon\mathbf{C}$ and $\mathbf{c}'\epsilon\mathbf{C}$, $\mathbf{c}'$ is erroneously decoded given that $\mathbf{c}$ was transmitted. We assume that every RLL codeword is transmitted equally likely with the probability $P(\mathbf{c}) = 1/|\mathbf{C}|$. Intersymbol interference-free signaling with transmission over the AWGN channel is assumed. After a matched filter detector, we apply ML decoder. Using the fact that the codewords having the minimum distance $d_{\min}$ have the highest two codeword error probability $P(d_{\min})$, we write the PEP as

$$\begin{aligned} P(\mathbf{c}\rightarrow\mathbf{c}') &\leq P(d_{\min}), \\ &\leq Q(\sqrt{\frac{d_{\min}2E_s}{N_0}}), \end{aligned} \tag{3.17}$$

where the Q-function is defined as $Q(\alpha) = 1/\sqrt{2\pi}\int_{\alpha}^{\infty} e^{-\frac{x^2}{2}}dx$, and E_s is the RLL bit energy in a duration τ'. BPSK modulation is considered, and since the system is designed such that $R_r(d+1) = 1$, the information bit energy E_b in a symbol duration τ can be calculated as $E_s = E_b R_r = E_b/(d+1)$. Thus, Eq. (3.17) can be rewritten as

$$P(\mathbf{c}\rightarrow\mathbf{c}') \leq Q(\sqrt{\frac{d_{\min}}{(d+1)}\frac{2E_b}{N_0}}). \tag{3.18}$$

Note that compared with uncoded BPSK modulation, the same amount of information is transmitted within the same block transmission time $T = q\tau$. Since the total energy in a block time T does not change, there is no efficiency loss compared with uncoded BPSK modulation in Eq. (3.18). The asymptotic coding gain G_{asy} for large SNR is given by

$$G_{\text{asy}} = 10\log_{10}(\frac{d_{\min}}{d+1}). \tag{3.19}$$

It can be seen that the PEP of the RLL coding is dependent on the parameters $d_{\min}$ and $(d+1)$. Large values of d could give larger improvement in $(d+1)C_d$, see Table 3.5. However, codes with larger values of d can be constructed at the expense of a lower efficiency and an increased complexity due to the increased RLL code length. Furthermore, the loss due to the parameter $(d+1)$ can be compensated by using higher minimum distance $d_{\min} > 1$. However, codes with desired $d_{\min}$ leads to a lower efficiency, see Table 3.4. From Eq. (3.16), the RLL codeword error probability can be upper bounded as

$$P_{RLL} \leq |\mathbf{C}|Q(\sqrt{\frac{d_{\min}}{(d+1)}\frac{2E_b}{N_0}}). \tag{3.20}$$

Fig. 3.9 depicts the simulated RLL codeword error probability as a function of E_b/N_0 for varying d and $d_{\min}$. The transmission over the AWGN channel is assumed for the all curves. As we discussed, the RLL codeword error probability depends mainly on the code parameters $d_{\min}$ and $(d+1)$. The error performance is driven by the expected coding gain $10\log_{10}(\frac{d_{\min}}{d+1})$.

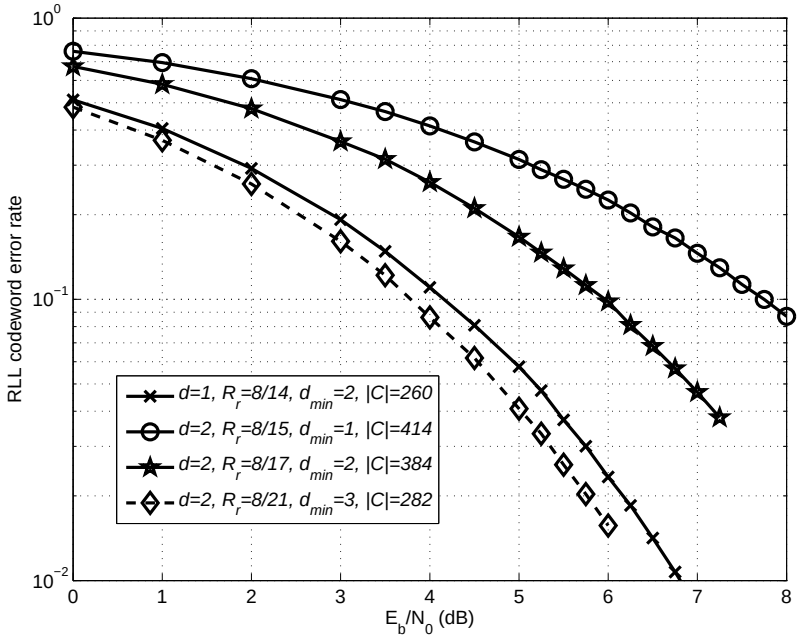

Figure 3.9: RLL coded BPSK transmission with varying d and $d_{\min}$ (the same minimum channel symbol duration).

3.5 RS-RLL Concatenated Coding Scheme

It can be observed in Table 3.4 that for $d_{\min} < (d+2)$, high rate codes can be selected such that the product $R_r(d+1)$ is larger than one. In this case, larger coding gains can be achieved by a serial concatenation of an error control code (ECC) with an RLL code. Accordingly, since a single bit error can cause a burst error at the RLL decoder, the outer ECC code must have the capability of correcting burst errors. A suitable example for this type of a code is RS code.

The basic transmitter is shown in Fig. 3.10. An advantage of this scheme is that since there is no error propagation between the RLL blocks, we can efficiently use the RS code error correction capability. The explanation of the modified scheme is as follows.

Without loss of generality, we again consider an uncoded system with BPSK modulation, where the information has a bit duration of τ seconds. We transmit q information bits in time $T = q\tau$ with a total energy qE_b. This corresponds to the set of $2^{T/\tau}$ block waveforms. If RS coding is used with a rate $R_{RS} = q/m$, we transmit m RS symbols with a minimum code symbol duration of $\widehat{\tau} = \tau\frac{q}{m}$ seconds in the same time T. The energy per RS symbol is $E_{RS} = q/mE_b$. A binary RLL encoder with $R_r = m/n$ is applied after the RS encoder. The set of all possible sequences which the RLL encoder can generate satisfies the $(d+1)$ constraint. The time interval T is partitioned into RLL symbols with duration τ' seconds. This corresponds to the set of $2^{TR_{RS}R_r/\tau'}$ block waveforms for q information bits. Since the comparisons are made on the basis of the same minimum channel symbol duration, we choose $\tau' = \frac{\tau}{(d+1)}$ seconds, see Fig. 3.11. Thus, a number of $TR_{RS}R_r(d+1)/\tau$ input symbols can be mapped onto the set of $2^{TR_{RS}R_r(d+1)/\tau}$ block waveforms. For $R_{RS}R_r(d+1) = 1$, the same amount of information with channel coding can be transmitted within the same minimum channel symbol duration compared with the uncoded modulation. The energy per RLL symbol is reduced to $E_s = E_b/(d+1)$, and the total energy in time T is equal to qE_b. The design criteria for the coded modulation scheme can be summarized as

$$R_{RS}R_r(d+1) = \frac{q}{m}\frac{m}{n}(d+1) = 1. \tag{3.21}$$

If Eq. (3.21) is satisfied, we can conclude that the essential PSD of the coded BPSK modulation scheme is almost the same as essential PSD of the uncoded BPSK modulation. Furthermore, the rate of the RS code can be lower bounded as

$$R_{RS} \geq \frac{1}{(d+1)C_d}, \tag{3.22}$$

see Table 3.5 for the quantity $(d+1)C_d$ as a function of d.

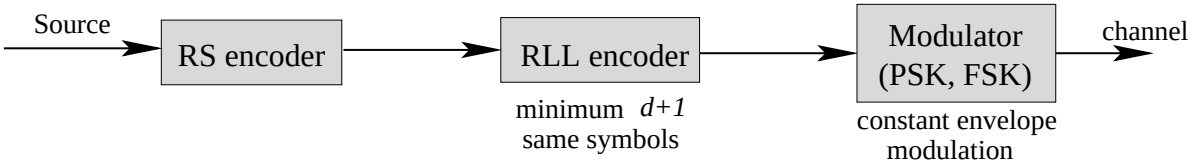

Figure 3.10: Block diagram of a basic transmitter for the concatenated encoding.

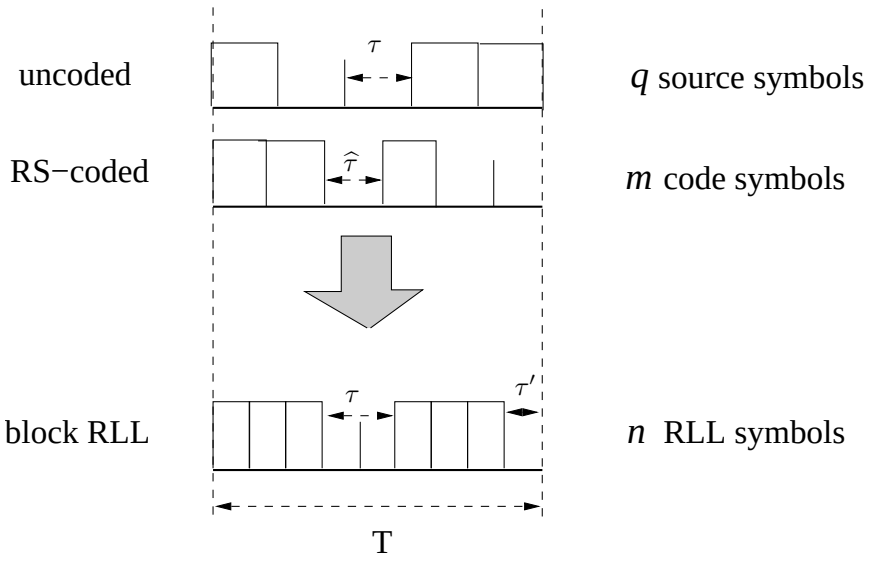

Figure 3.11: Illustration of the binary waveform of the RS-RLL coded sequences.

3.5.1 Design Example of the RS-RLL Concatenated Coding Scheme

A conventional RS code is a non-binary block code with code parameters $(2^p-1, 2^p-1-2t)$, where p is the number of bits in a symbol, 2^p-1 is the block length in symbols, 2^p-1-2t is the number of information symbols in the message, and $2t$ is the number of parity symbols. Thus, the rate can be calculated as $R_{RS} = q/m = (2^p - 1 - 2t)/(2^p - 1)$. The code is capable of correcting any combination of t or fewer errors. In the construction of RS codes, 2^p symbols are formed by the Galois field of 2^p, shortly GF(2^p). For more information on RS codes, we refer to [58].

Example 3.4. Let us consider Fig. 3.12 for the design example using RS code over GF(2^8). It can be seen that a RS(255,223) code with 8 bits per symbol is used as an outer code. This means that symbols are arranged in words of 255 symbols, of which 223 symbols are information and the other 32 are parity. Whenever the RS decoder detects 16 or fewer symbol errors, it can correct these errors. Each RS symbol is encoded further by a $(d = 2)$-constraint RLL code with $d_{\min} = 3$. The number of bits per RS symbol is chosen as the length of the input block of the RLL encoding. The rates of the RS and RLL codes are selected such that

$$R_{RS}R_r(d+1) = \frac{223}{255}\frac{8}{21}3 \approx 1.$$

Moreover, a concatenated decoding scheme with a soft RLL block-wise decoding followed by a hard RS decoding can be applied. In that way, a combination of RS and RLL code may be quite advantageous.

It should also be noted that the additional complexity introduced by the RLL encoder can be reduced by selecting small RLL codeword lengths, see Table 3.4.

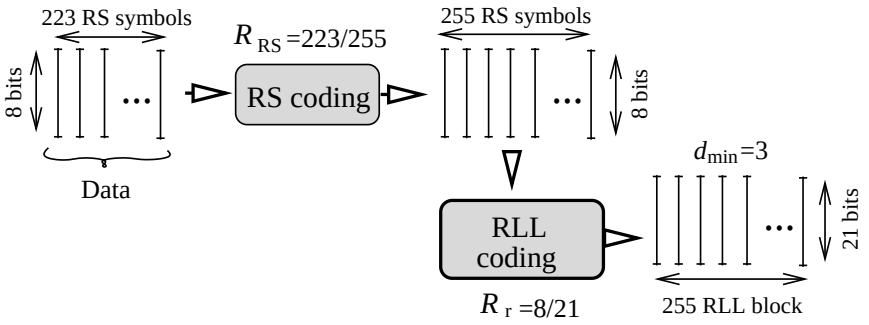

Figure 3.12: Bit allocation in the encoding of RS(255,223) and ($d = 2$)-constraint RLL codes.

3.5.2 The Performance of the RS-RLL Coded BPSK and Coherent 2-FSK Modulation

The RS codes are particularly useful for the burst error correction. The RS decoded symbol error probability, P_{sym}, in terms of the RLL codeword error probability, P_{RLL} is written

$$P_{sym} = 1 - \sum_{j=0}^{t} \binom{2^p - 1}{j} P_{RLL}^{j} (1 - P_{RLL})^{2^p - 1 - j}, \tag{3.23}$$

where $\binom{2^p-1}{j}$ denotes the binomial coefficient, and it is evaluated as $\binom{2^p-1}{j} = \frac{(2^p-1)!}{j!(2^p-1-j)!}$. An upper bound on the P_{RLL} for BPSK modulation is given in Eq. (3.20). A performance trade-off can be observed between the coding gains achieved by the RS code and the RLL code. For $\frac{d_{\min}}{(d+1)} = 1$, no coding gain can be achieved from the RLL code, since $10 \log_{10} \frac{d_{\min}}{(d+1)} = 0$. Thus, the performance is driven by the RS coding gain. Table 3.7 illustrates some of the constructed RLL codes for $\frac{d_{\min}}{(d+1)} = 1$.

Table 3.7: Selected RLL codes for $d_{\min} = d + 1$.

d	R_r	$d_{\min}$	$R_r(d+1)$	R_{RS}
1	8/14	2	8/7	223/255
2	8/21	3	8/7	223/255
3	6/20	4	6/5	53/63

Fig. 3.13 depicts the bit error rate (BER) as a function of E_b/N_0 for the transmission over the AWGN channel. The code efficiencies are chosen according to Table 3.7. It is apparent that the transmission with the concatenation of RS and RLL coding has a significant BER advantage over the uncoded BPSK transmission within the same minimum channel symbol duration. The improvement achieved by the proposed system is 4 dB at BER of 10^{-5}. Large coding gains are obtained by using a RLL code as an inner ECC. The RS(255,223)-RLL($d + 1 = 3$) with $d_{\min} = 3$ shows a slightly better performance due to its distance profile in the code. Note that since $R_{RS} R_r (d + 1) = 1$, the PSD of the RS-RLL coded BPSK modulation stays almost the same compared to the uncoded BPSK modulation.

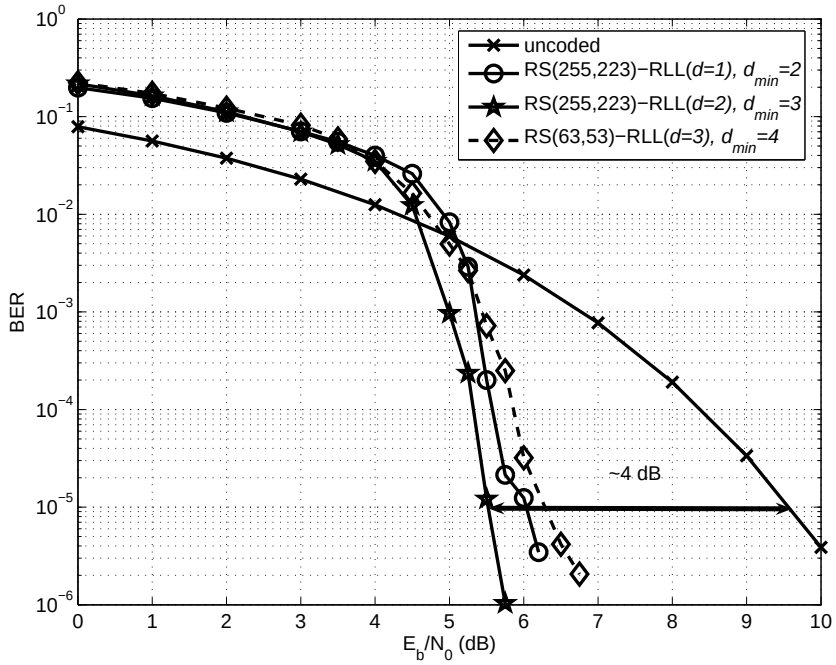

Figure 3.13: RS-RLL coded BPSK transmission with varying d and $d_{\min}$ (the same minimum channel symbol duration).

For the RLL codes having $d_{\min} = 1$, RLL block-wise soft decoding cannot be applied. In this case, a concatenated decoding scheme with a RLL block-wise hard decoding followed by a hard RS decoding can be applied. Table 3.8 illustrates some of the constructed RLL codes for $d_{\min} = 1$.

Table 3.8: Selected RLL codes for $d_{\min} = 1$.

d	R_r	$d_{\min}$	$R_r(d+1)$	R_{RS}
1	8/12	1	8/6	191/255
2	8/15	1	8/5	159/255
3	8/18	1	6/(4.5)	145/255

Fig. 3.14 illustrates the BER performance for varying $d_{\min}$ parameters. The performance trade-off between the coding gains achieved by the RS and the RLL code is quite apparent. The curve with $d_{\min} = (d + 1)$ has the best BER performance. The RLL codewords are decoded using soft decision decoding. The uncorrectable errors can be decoded by the RS decoder using hard decision decoding.

In Fig. 3.15, we plotted the BER performance of coherent 2-FSK transmission. In all curves, the minimum channel symbol duration is kept constant. Since the FSK transmission has a constant envelope modulation, RS-RLL coded FSK modulation scheme can be suitable for the PLC, where there is a limitation on the maximum transmitted power and bandwidth.

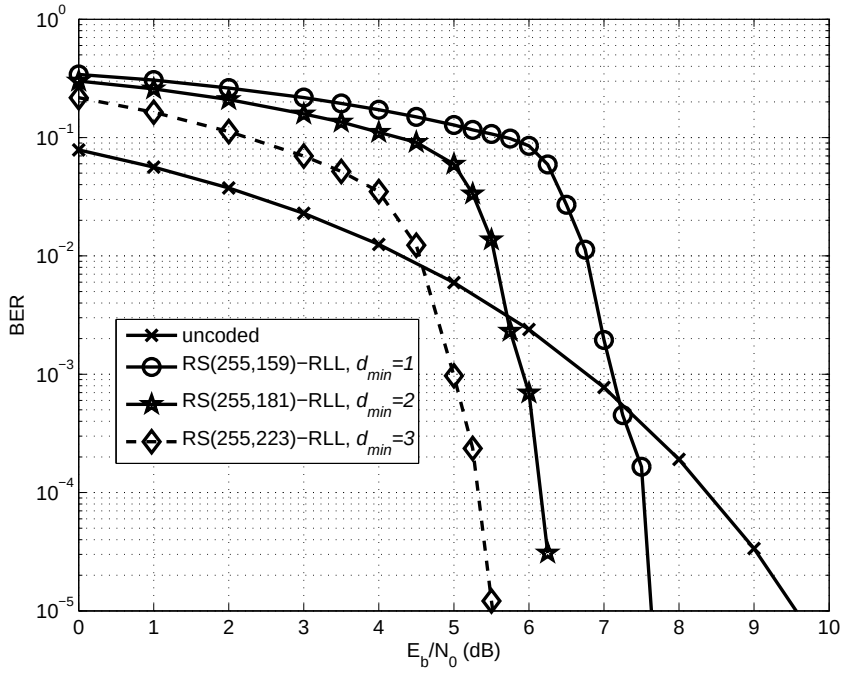

Figure 3.14: RS-RLL coded BPSK transmission with varying $d_{\min}$ for $(d+1) = 3$ (the same minimum channel symbol duration).

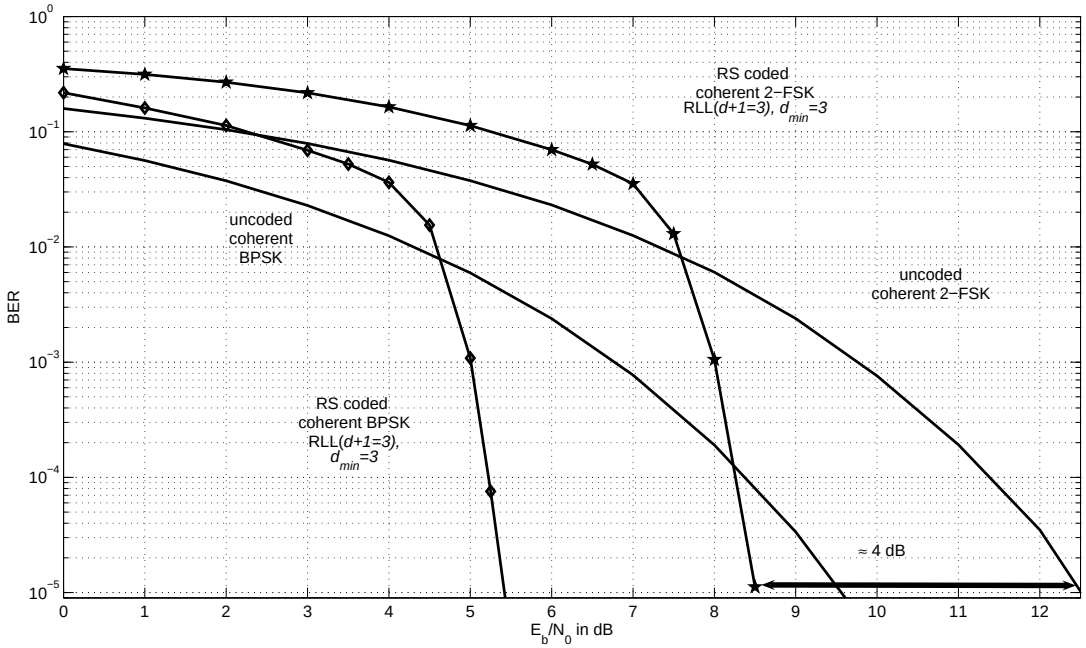

Figure 3.15: RS-RLL coded coherent 2-FSK and BPSK transmission (the same minimum channel symbol duration).

3.5.3 Bliss Encoding Scheme

It has been proposed in [28] and [27] to revert the order of application of the RS code and the RLL code to avoid the error propagation introduced by the RLL decoding. In Bliss encoding, the source data are first encoded into a long RLL codeword. This leads to a construction of long RLL encoded data stream using a high rate RLL encoder. Then, the RLL codewords are grouped into bytes, and they are encoded by the RS encoder. Since the parity bytes that are generated by the RS encoder do not obey the d-constrained, a second RLL encoder is applied to encode the parity bytes. The rate of the second RLL encoder is lower than the first RLL encoder. The basic block diagram of the encoder can be illustrated as in Fig. 3.16.

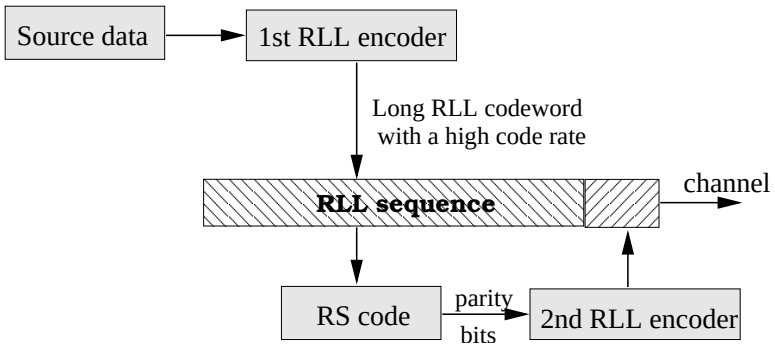

Figure 3.16: Basic block diagram of the Bliss encoding scheme.

The disadvantage of this scheme is that since no soft decoding is possible at the decoder side, large coding gains cannot be achieved. The second RLL decoder has to perform a hard decision decoding on the received parity bits. Afterwards, the RS decoder performs hard decision on symbols with a reduced symbol energy. Then, the first decoder delivers the source sequence. The advantage of the Bliss scheme depends on the construction of high rate RLL code which is applied in the first RLL encoder. In our proposed RLL code construction algorithm, we show that the constructed RLL code rates with $d_{\min} = 1$ are already very close to their theoretical limits, see Table 3.4.

3.6 The Performance of RS-RLL Coded Noncoherent FSK Modulation

In this section, we discuss the performance of the RS-RLL coded FSK modulation scheme over the AWGN channel with noncoherent reception. The basic system under investigation is shown in Fig. 3.17. Without loss of generality, we consider coded 2-FSK modulation, where the transmitter encodes the information using a block code $\boldsymbol{C}$ of length n with codewords $\mathbf{c} = (c_1, ..., c_n)$ and code symbols $c_k \in \{0, 1\}$. The 2-FSK modulator maps each symbol c_k onto one of the corresponding frequencies f_0 and f_1. The frequencies are chosen such that they are multiple of $1/\tau'$ in order to satisfy the orthogonality, where τ' denotes the duration of the one RLL symbol. Hence, the minimum frequency separation between the frequencies is $1/\tau' = (d+1)/\tau$, where τ is the bit duration of the uncoded system. We note that the transmission of only one frequency per time unit automatically leads to a constant envelope modulator output.

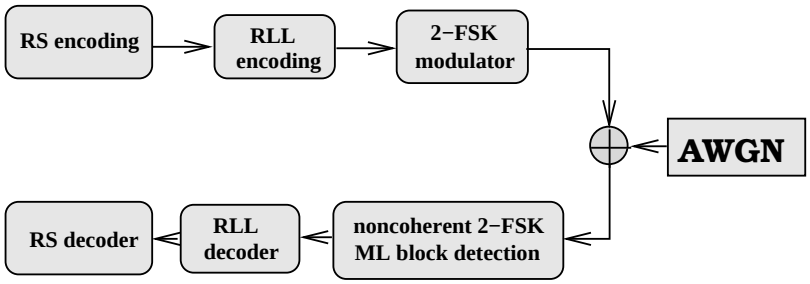

Figure 3.17: Basic block diagram of RS-RLL coded noncoherent 2-FSK transmission over AWGN channel.

3.6.1 Demodulation and ML Block Decoding of FSK symbols

Let us assume that the code symbols are transmitted over a memoryless AWGN channel. The receiver consists of a demodulation unit and of a maximum-likelihood block decoding unit. As a demodulation unit, a noncoherent correlation receiver is used [61]. For a transmitted codeword $\mathbf{c} \in \boldsymbol{C}$, the two correlators outputs, namely in-phase I and quadrature Q components at the transmitted frequency c_k and the time instant k are

$$r_{c_k,k,I} = \sqrt{E_s}\cos\phi + g_{c_k,k,I}, \tag{3.24}$$

$$r_{c_k,k,Q} = \sqrt{E_s}\sin\phi + g_{c_k,k,Q}, \tag{3.25}$$

where $E_s = E_b/(d+1)$ is the channel symbol energy, ϕ is an arbitrary phase introduced by the channel, and the noise samples $g_{c_k,k,\{I,Q\}}$ are complex Gaussian random variables with variance σ_g^2. For the frequency $l \neq c_k$, the signal components vanish independent of the values of the phase shift ϕ. In such a case, the other correlator outputs consist of complex Gaussian noise only, i.e.,

$$r_{l,k,I} = g_{l,k,I}, \qquad r_{l,k,Q} = g_{l,k,Q}, \qquad l \neq c_k. \tag{3.26}$$

Furthermore, using Eq. (2.1), the joint PDF for $r_{c_k,k,I}$ and $r_{c_k,k,Q}$ conditioned on ϕ is

$$p(r_{c_k,k,I}, r_{c_k,k,Q}|\ phi) = \frac{1}{2\pi\sigma_g^2} e^{-[(r_{c_k,k,I}-\sqrt{E_s}\cos\phi)^2+(r_{c_k,k,Q}-\sqrt{E_s}\sin\phi)^2]/2\sigma_g^2} \tag{3.27}$$

and for $l \neq c_k$, we have

$$p(r_{l,k,I}, r_{l,k,Q}) = \frac{1}{2\pi\sigma_g^2} e^{-(r_{l,k,I}^2+r_{l,k,Q}^2)/2\sigma_g^2}. \tag{3.28}$$

ML Block Decoding of Noncoherent 2-FSK modulation for the AWGN channel

The decoding of the transmitted RLL codeword is based on the received vector $\mathbf{r} = (r_1, ..., r_n)$. We assume that every RLL codeword is sent with equal probability $P(\mathbf{c}) = 1/|\mathbf{C}|$. The ML receiver chooses the codeword $\mathbf{c}$ that maximizes the conditional PDF $p(r_1, r_2, ..., r_n|\mathbf{c})$. The decoding metric $\varphi(\mathbf{c}, \mathbf{r})$ can be written as

$$\varphi(\mathbf{c}, \mathbf{r}) = \max_{\mathbf{c}\in\mathbf{C}} p(\mathbf{r}|\mathbf{c}). \tag{3.29}$$

The PDF $p(\mathbf{r}|\mathbf{c})$ can be obtained by averaging the PDFs $p(\mathbf{r}|\mathbf{c},\phi)$ over the PDF of the random carrier phase, i.e.,

$$p(\mathbf{r}|\mathbf{c}) = \int_0^{2\pi} p(\mathbf{r}|\mathbf{c},\phi)p(\phi)d\phi, \tag{3.30}$$

where we assume that the phase shift ϕ stays constant in the interval $0 \leq \phi \leq 2\pi$ for a block length n, and it has a uniform distribution, i.e., $p(\phi) = 1/2\pi$. We neglect the factors independent of the codeword $\mathbf{c}$, since they do not affect the maximization. As the channel is memoryless, Eq. (3.29) can be rewritten as

$$\varphi(\mathbf{c},\mathbf{r}) = \max_{\mathbf{c}\in\mathbf{C}} \int_0^{2\pi} \prod_{k=1}^{n} p(r_k|c_k,\phi)d\phi. \tag{3.31}$$

The joint PDF of $r_k = [r_{0,k,I}\ r_{0,k,Q}\ r_{1,k,I}\ r_{1,k,Q}]$ is a product of the marginal PDFs, i.e.,

$$p(r_k|c_k,\phi) = p(r_{c_k,k,\{I,Q\}}|c_k,\phi)p(r_{l,k,\{I,Q\}}|c_k). \tag{3.32}$$

Using Equations (3.27) and (3.28), Eq. (3.32) can be rewritten as

$$\begin{aligned} p(r_k|c_k,\phi) &= (\frac{1}{2\pi\sigma_g^2})^2 e^{-\frac{[(r_{c_k,k,I}-\sqrt{E_s}\cos\phi)^2+(r_{c_k,k,Q}-\sqrt{E_s}\sin\phi)^2]}{2\sigma_g^2}} e^{-\frac{(r_{l,k,I}^2+r_{l,k,Q}^2)}{2\sigma_g^2}}, \\ &= (\frac{1}{2\pi\sigma_g^2})^2 e^{-\frac{r_{c_k,k,I}^2+r_{c_k,k,Q}^2+r_{l,k,I}^2+r_{l,k,Q}^2+E_s}{2\sigma_g^2}} e^{\frac{\sqrt{E_s}(r_{c_k,k,I}\cos\phi+r_{c_k,k,Q}\sin\phi)}{\sigma_g^2}}. \end{aligned} \tag{3.33}$$

Substitution of this result in Eq. (3.31) and neglecting all factors independent of the codeword $\mathbf{c}$ yields ML decoding metric as

$$\varphi(\mathbf{c},\mathbf{r}) = \max_{\mathbf{c}\in\mathbf{C}} \int_0^{2\pi} \prod_{k=1}^{n} e^{\sqrt{E_s}(r_{c_k,k,I}\cos\phi+r_{c_k,k,Q}\sin\phi)}d\phi, \tag{3.34}$$

$$= \max_{\mathbf{c}\in\mathbf{C}} \int_0^{2\pi} e^{\sum_{k=1}^{n}(\sqrt{E_s}(r_{c_k,k,I}\cos\phi+r_{c_k,k,Q}\sin\phi))}d\phi, \tag{3.35}$$

$$= \max_{\mathbf{c}\in\mathbf{C}} I_0\left(n\sqrt{E_s}\sqrt{\left(\sum_{k=1}^{n} r_{c_k,k,I}\right)^2 + \left(\sum_{k=1}^{n} r_{c_k,k,Q}\right)^2}\right)d\phi, \tag{3.36}$$

where $I_0(x)$ is the modified Bessel function of order zero. Note that maximizing $P(\mathbf{r}|\mathbf{c})$ is equivalent to maximizing $\ln P(\mathbf{r}|\mathbf{c})$. By applying logarithmic maximization and the steepest-descent approximation $I_0(x) = \exp(x^2/4)$, the ML decoding metric can be written as

$$\varphi(\mathbf{c}) = \max_{\mathbf{c}\in\mathbf{C}} \left[\left(\sum_{k=1}^{n} r_{c_k,k,I}\right)^2 + \left(\sum_{k=1}^{n} r_{c_k,k,Q}\right)^2\right], \tag{3.37}$$

see Fig. 3.18 for the block diagram of the receiver.

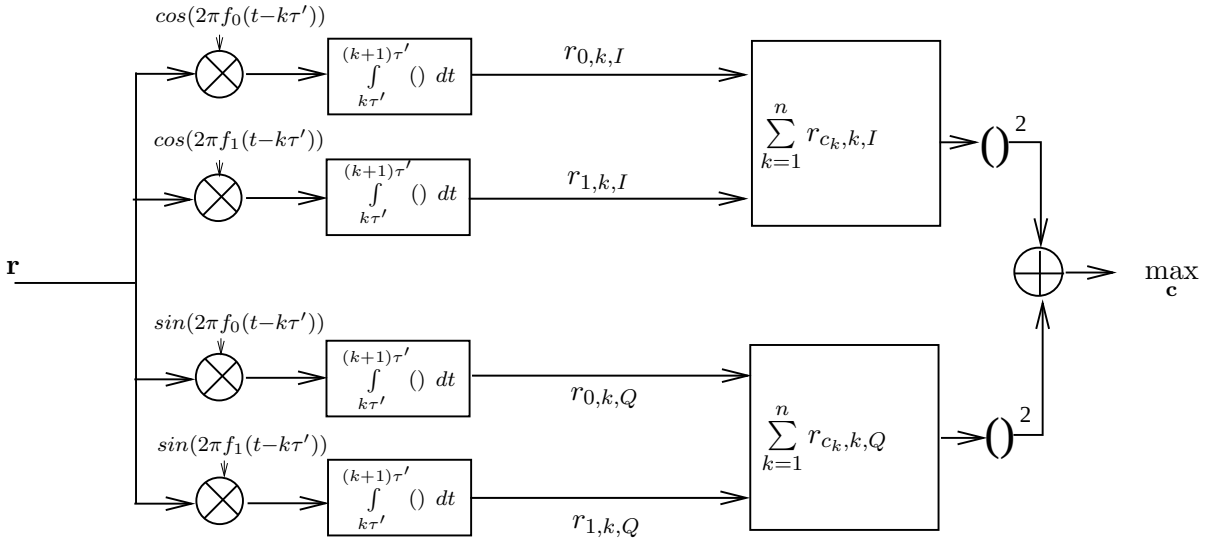

Figure 3.18: Block diagram of the block-wise noncoherent ML receiver for the AWGN channel.

Fig. 3.19 depicts the simulation results of noncoherent FSK transmission for the AWGN channel. In all curves, the minimum channel symbol duration is kept the same. It is apparent that the concatenation of RS and RLL coding has a significant advantage over the uncoded 2-FSK without changing the essential PSD of the channel symbols, see Fig. 3.7. The improvement achieved by the proposed system is 3 dB at BER of 10^{-3} and 5 dB at BER of 10^{-5}. Large coding gains are obtained by using the RLL code as an inner code, which can apply block-wise soft decision decoding. The RS(255,223)-RLL($d = 2$) with $d_{\min} = 3$ has the best performance. The RLL code corrects single bit errors, whereas the RS code corrects the remaining symbol errors.

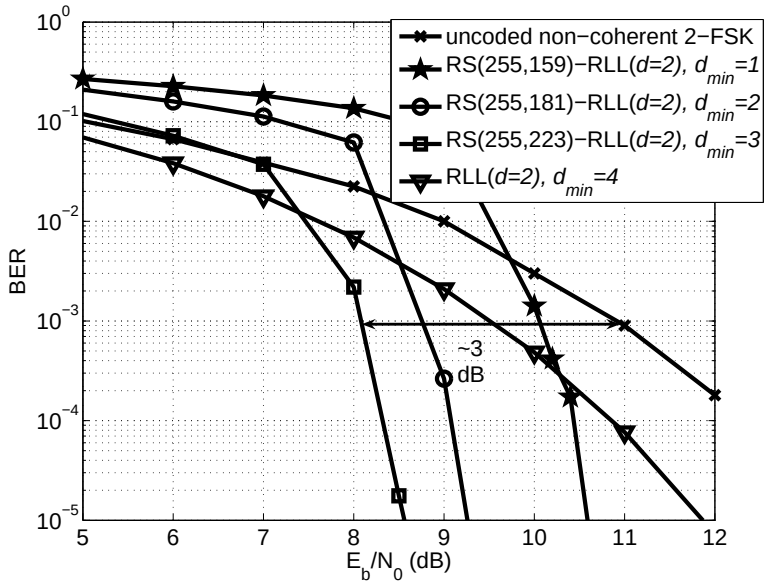

Figure 3.19: RS-RLL coded 2-FSK with varying $d_{\min}$ for the AWGN channel.

3.7 M-level RLL Coded M-ary FSK modulation

So far, we have discussed the use of binary RLL coded modulation schemes. For an M-level or M-ary symbol alphabet $\{0, 1, ..., M-1\}$, an M-ary d-constrained RLL sequence has the property that each run of symbols must have a length at least $d+1$. Extension of the binary RLL coded modulation scheme to M-ary RLL coded modulation scheme is as follows. We consider a concatenated coding scheme with RS coding in the outer stage and RLL coded FSK modulation in the inner stage, see Fig. 3.20. Each RS symbol with m bits per symbol is encoded by the M-ary d-constrained RLL encoder which translates the RS symbols of m bits into RLL codewords of n M-ary symbols. The rate of the RLL coding is calculated as $R_c = m/n$ bits per symbol. Afterwards, an M-ary FSK modulation scheme can be applied by assigning a transmission frequency for each M-ary symbol. The design criteria for the coded modulation scheme can be rewritten as

$$R_{RS} R_r (d+1) = \log_2 M. \tag{3.38}$$

If Eq. (3.38) is satisfied, we can conclude that the same amount of information with channel coding can be transmitted within the same minimum channel symbol duration compared with the uncoded modulation. Thus, the PSD of the M-ary RLL coded modulation scheme is almost the same as the PSD of the uncoded M-ary modulation scheme.

Table 3.9 shows the parameters of constructed ternary (M=3) and quaternary (M=4) RLL codes for $d_{\min} = 2$ and $d = 1$. We also added the rate of the RS encoder that we can accommodate without changing the minimum channel symbol duration.

Table 3.9: Selected multi-level RLL codes.

M	d	R_r	$d_{\min}$	R_{RS}
3	1	10/12=0.83 ($C_d = 1$)	2	0.95
4	1	11/10=1.1 ($C_d = 1.2$)	2	0.91

It should be noted that as M increases, the lower bound on the rate of the RS code also increases which decreases the RS coding gain. For the M-ary RLL coded modulation scheme, the lower bound on the rate of the RS code can be written as

$$R_{RS} \geq \frac{\log_2 M}{(d+1)C_d}. \tag{3.39}$$

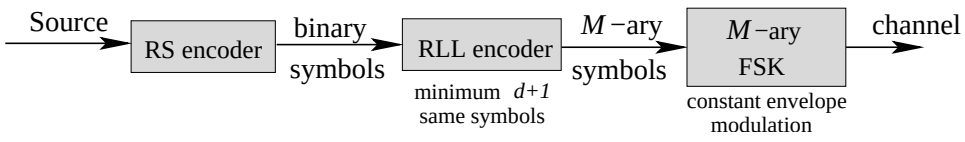

Figure 3.20: Block diagram of the M-ary RLL coded modulation scheme.

Table 3.10 shows $\frac{\log_2 M}{(d+1)C_d}$ as a function of M. A significant loss in the minimum achievable RS code rate can be observed. This implies a limitation on the M-level RLL coded modulation.

Table 3.10: The lower bound on the rate of the RS code for $d = 1$ and varying M.

M	$\frac{\log_2 M}{(d+1)C_d}$
2	0.7203
3	0.7925
4	0.8333

Fig. 3.21 illustrates the BER performance of the M-ary RLL coded noncoherent M-FSK transmission over the AWGN channel. The code rates are chosen according to Table 3.9. The minimum channel symbol duration is the same for the M-FSK transmission curves, and the frequencies are chosen so that they are multiple of $1/\tau'$ in order to satisfy the orthogonality, where τ' denotes the duration of the one RLL symbol. As expected, the 4-ary RLL coded FSK system outperforms the binary and the ternary coded FSK systems.

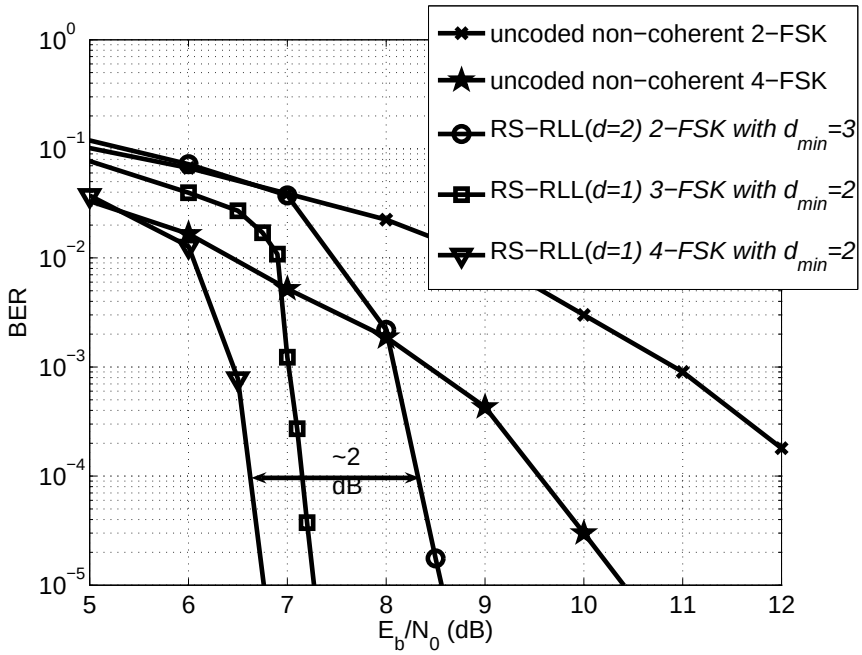

Figure 3.21: M-ary RLL coded noncoherent M-FSK with RS coding for the AWGN channel.

3.8 ML Block Decoding of Noncoherent 2-FSK Modulation for the AWCN Channel

In this section, the performance of the RS-RLL coded FSK modulation scheme for the block transmission over the AWCN channel is discussed. We also consider the influence of the narrow band noise on the receiver. Fig. 3.22 shows the basic block diagram of the communication scheme under investigation.

Let us first assume that the code symbols are transmitted over a memoryless AWCN channel, see Section 2.2 for the channel definition. The receiver consists of a demodulation unit and of a ML block decoder. As a demodulation unit, a noncoherent correlation receiver is used [61]. For a transmitted codeword $\mathbf{c} \in \boldsymbol{C}$, the two correlators outputs, namely in-phase I and quadrature Q components at the transmitted frequency c_k and the time instant k are

$$r_{c_k,k,I} = \sqrt{E_s}\cos\phi + w_{c_k,k,I}, \tag{3.40}$$

$$r_{c_k,k,Q} = \sqrt{E_s}\sin\phi + w_{c_k,k,Q}, \tag{3.41}$$

where E_s is the channel symbol energy, ϕ is an arbitrary phase introduced by the channel, $w_{c_k,k,\{I,Q\}}$ are the channel noise symbols at time instant k that consist of the impulsive noise and the background noise symbols, see Section 2.2. For the frequency $l \neq c_k$, the signal components vanish independent of the values of the phase shift ϕ. Hence, the other two correlator outputs consist of only noise.

The idea behind the decoder is to estimate the channel state vector $\mathbf{s} = (s_1, ..., s_n)$ based on the received vector $\mathbf{r} = (r_{0,1,\{I,Q\}}, r_{1,1,\{I,Q\}} ..., r_{0,n,\{I,Q\}}, r_{1,n,\{I,Q\}})$. Then, given $\mathbf{s}$, the receiver chooses the codeword $\mathbf{c}$ which maximizes the conditional PDF $p(\mathbf{r}|\mathbf{c}, \mathbf{s})$.

3.8.1 Estimation of Channel State Information

In this subsection, we design an channel state estimator based on the received vector $\mathbf{r}$. Since the real and the imaginary part of the impulsive noise have the same variance and PDF, the estimation unit calculates

$$\begin{aligned} R_k &= (r_{0,k,I} + jr_{0,k,Q}) + (r_{1,k,I} + jr_{1,k,Q}), \\ &= \sqrt{E_s}(\cos\phi + j\sin\phi) + \widehat{w}_k, \end{aligned} \tag{3.42}$$

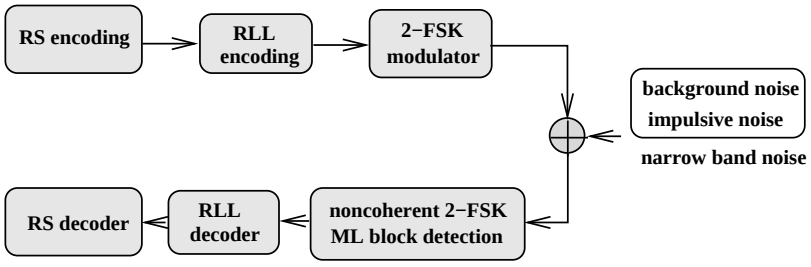

Figure 3.22: Basic block diagram of RS-RLL coded noncoherent 2-FSK transmission over AWCN channel.

where R_k is the complex value which is the sum of the correlator outputs for one symbol observation at time instant k, and $\widehat{w}_k$ is the complex noise symbol with zero-mean and variance $2\sigma^2_{s_k}$. Using the following relations

$$|R_k| \leq \sqrt{E_s} + |\widehat{w}_k| \tag{3.43}$$

and Eq. (2.8), the estimator selects the channel state s_k that maximizes the following

$$\varphi(s_k) = \max_{s_k=0,1,2} \frac{A^{s_k}}{s_k!\sigma^2_{s_k}} e^{(-\frac{(|R_k|-\sqrt{E})^2}{4\sigma^2_{s_k}})}, \tag{3.44}$$

where we neglected all factors independent of the maximization and truncated the number of channel states to three. In Eq. (3.44), $P(s_k) = e^{-A}\frac{A^{s_k}}{s_k!}$ is the probability of selecting channel s_k with the channel noise variance $\sigma^2_{s_k} = \sigma^2_g + \frac{\sigma^2_i s_k}{A}$. In estimating the vector $\mathbf{s}$, Eq. (3.44) must be computed for every signal interval k, where $k = 1, ..., n$. Fig. 3.23 shows the block diagram of the channel state estimator.

3.8.2 ML Decoding Rule

For a given channel state information vector $\mathbf{s}$, the ML decoder maximizes the conditional PDF $p(\mathbf{r}|\mathbf{c}, \mathbf{s})$. It can be obtained by averaging the PDFs $p(\mathbf{r}|\mathbf{c}, \mathbf{s}, \phi)$ over the PDF of the random carrier phase, i.e., it holds that

$$\max_{\mathbf{c}} p(\mathbf{r}|\mathbf{c}, \mathbf{s}) = \max_{\mathbf{c}} \int_0^{2\pi} \prod_{k=1}^{n} p(r_k|c_k, s_k, \phi)p(\phi)d\phi, \tag{3.45}$$

where s_k is the channel state while transmitting the code symbol c_k at time instant k. Moreover, we assume that the phase shift ϕ stays constant in the interval $0 \leq \phi \leq 2\pi$ for a block length n, and it has a uniform distribution, i.e., $p(\phi) = 1/2\pi$. We neglect the factors independent of the codeword $\mathbf{c}$, since they do not affect the maximization. As the channel is memoryless, Eq. (3.45) can be rewritten as

$$\max_{\mathbf{c}\in\mathbf{C}} \int_0^{2\pi} \prod_{k=1}^{n} p(r_k|c_k, s_k, \phi)d\phi. \tag{3.46}$$

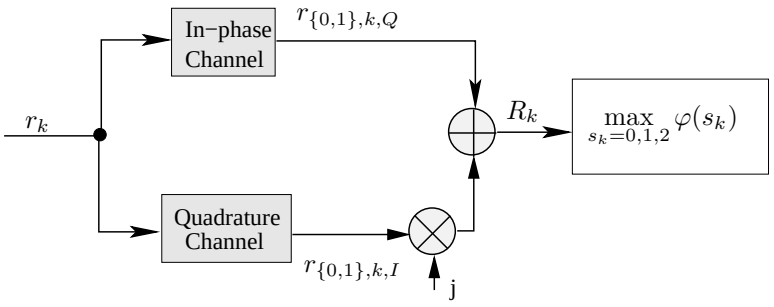

Figure 3.23: Block diagram of the channel state estimator for the noncoherent 2-FSK transmission over the AWCN channel.

The joint PDF of $r_k = [r_{0,k,I}\ r_{0,k,Q}\ r_{1,k,I}\ r_{1,k,Q}]$ may be expressed as a product of the marginal PDFs, i.e.,

$$p(r_k|c_k, s_k, \phi) = p(r_{c_k,k,\{I,Q\}}|c_k, s_k, \phi)p(r_{l,k,\{I,Q\}}|c_k, s_k); \quad l \neq c_k. \tag{3.47}$$

Using Equation (2.10), Eq. (3.47) can be rewritten as

$$\begin{aligned} p(r_k|c_k, s_k, \phi) &= (\frac{1}{2\pi\sigma_{s_k}^2})^2 e^{-\frac{[(r_{c_k,k,I}-\sqrt{E_s}\cos\phi)^2+(r_{c_k,k,Q}-\sqrt{E_s}\sin\phi)^2]}{2\sigma_{s_k}^2}} e^{-\frac{(r_{l,k,Q}^2+r_{l,k,I}^2)}{2\sigma_{s_k}^2}}, \\ &= (\frac{1}{2\pi\sigma_{s_k}^2})^2 e^{-\frac{r_{c_k,k,I}^2+r_{c_k,k,Q}^2+r_{l,k,I}^2+r_{l,k,Q}^2+E_s}{2\sigma_{s_k}^2}} e^{\frac{\sqrt{E_s}(r_{c_k,k,I}\cos\phi+r_{c_k,k,Q}\sin\phi)}{\sigma_{s_k}^2}} \end{aligned} \tag{3.48}$$

Substitution of this result in Eq. (3.45) and neglecting all factors independent of the codeword $\mathbf{c}$ yields ML decoding metric as

$$\max_{\mathbf{c}} p(\mathbf{r}|\mathbf{c}, \mathbf{s}) = \max_{\mathbf{c}\in\mathbf{C}} \int_0^{2\pi} \prod_{k=1}^{n} e^{\frac{\sqrt{E_s}(r_{c_k,k,I}\cos\phi+r_{c_k,k,Q}\sin\phi)}{2\sigma_{s_k}^2}} d\phi, \tag{3.49}$$

$$= \max_{\mathbf{c}\in\mathbf{C}} \int_0^{2\pi} e^{\sum_{k=1}^{n} \frac{(\sqrt{E_s}(r_{c_k,k,I}\cos\phi+r_{c_k,k,Q}\sin\phi))}{2\sigma_{s_k}^2}} d\phi, \tag{3.50}$$

$$= \max_{\mathbf{c}\in\mathbf{C}} I_0\left(n\sqrt{E_b}\sqrt{\left(\sum_{k=1}^{n}\frac{r_{c_k,k,I}}{2\sigma_{s_k}^2}\right)^2+\left(\sum_{k=1}^{n}\frac{r_{c_k,k,Q}}{2\sigma_{s_k}^2}\right)^2}\right) d\phi, \tag{3.51}$$

where $I_0(x)$ is the modified Bessel function of order zero. Note that maximizing $P(r|c)$ is equivalent to maximizing $\ln P(r|c)$. By applying logarithmic maximization and the steepest-descent approximation $I_0(x) = \exp(x^2/4)$, the ML decoding metric can be written as

$$\max_{\mathbf{c}} p(\mathbf{r}|\mathbf{c}, \mathbf{s}) = \max_{\mathbf{c}\in\mathbf{C}} \left[\left(\sum_{k=1}^{n}\frac{r_{c_k,k,I}}{\sigma_{s_k}^2}\right)^2+\left(\sum_{k=1}^{n}\frac{r_{c_k,k,Q}}{\sigma_{s_k}^2}\right)^2\right], \tag{3.52}$$

where the channel state s_k is estimated using Eq. (3.44). Fig. 3.24 shows the block diagram of the block-wise noncoherent ML receiver for the AWCN channel.

In the following simulation results, the BER is plotted as a function of the signal-to-Gaussian background noise ratio, i.e.,

$$SBNR = E_b/2\sigma_g^2 \tag{3.53}$$

where $E_b = E_s(d+1)$ is the energy in a bit duration τ. Furthermore, the impulsive noise duration is chosen as channel symbol duration τ', and the impulsive noise index is selected as $A = 0.1$. With this, a heavily disturbed impulsive channel is described. The ratio between the variance of the Gaussian background noise and the variance of the impulsive noise is chosen as $\Gamma = \frac{\sigma_g^2}{\sigma_i^2} = 10^{-3}$.

In Fig. 3.25, the BER for the uncoded (no RS-RLL encoding) noncoherent 2-FSK transmission over AWCN channel is depicted. ML block decoder is applied at the receiver. The

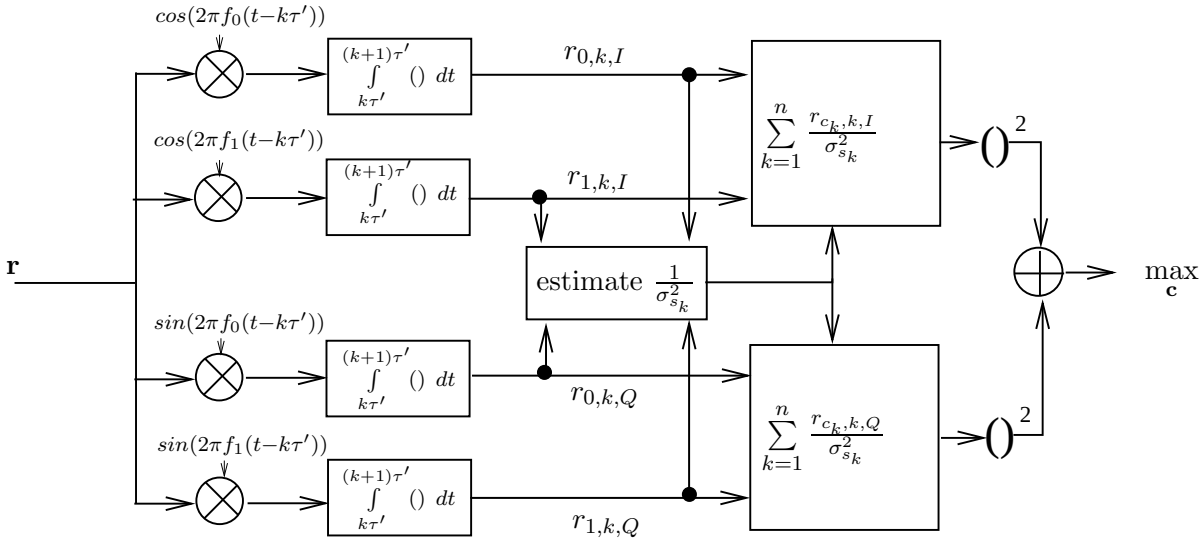

Figure 3.24: Block diagram of the block-wise noncoherent ML receiver for the AWCN channel.

block length n is chosen as eight. All curves obtained by the the uncoded 2-FSK transmission. Additionally, the noncoherent 2-FSK transmission over AWGN channel is plotted for comparison. A significant improvement can be observed by the noncoherent block decoding with channel state estimation. The AWCN performance with the proposed decoder follows the AWGN performance till 10 dB. In the region between 15-35 dB, the BER increases as the SBNR increases. This is due to the fact that, for increasing SBNR, the impulsive noise can hardly be distinguished, i.e., the estimator loses more and more information about the channel state and, therefore, approaches the performance of the receiver without channel state information.

Fig. 3.26 shows the AWCN performance of the RS-RLL coded 2-FSK system for the same minimum channel symbol duration. The code parameters are chosen according to Table 3.11. As a lower bound, the curves for the transmission over the AWGN channel are also presented. Since impulsive noise is expected to occur in every block, a large $d_{\min}$ in the RLL code can lead to an efficient use of the RS code. Therefore, RLL coding with $d_{\min} = 3$ can provide large coding gains without leading to an essential change in the PSD of the channel symbols. At a BER of 10^{-4}, we can observe only a 1 dB gap from the AWGN channel performance.

Table 3.11: RS and RLL code parameters for the noncoherent 2-FSK with ML block decoder for the AWCN channel.

d	R_r	$d_{\min}$	R_{RS}
2	8/15	1	159/255
2	8/17	2	181/255
2	8/21	3	223/255

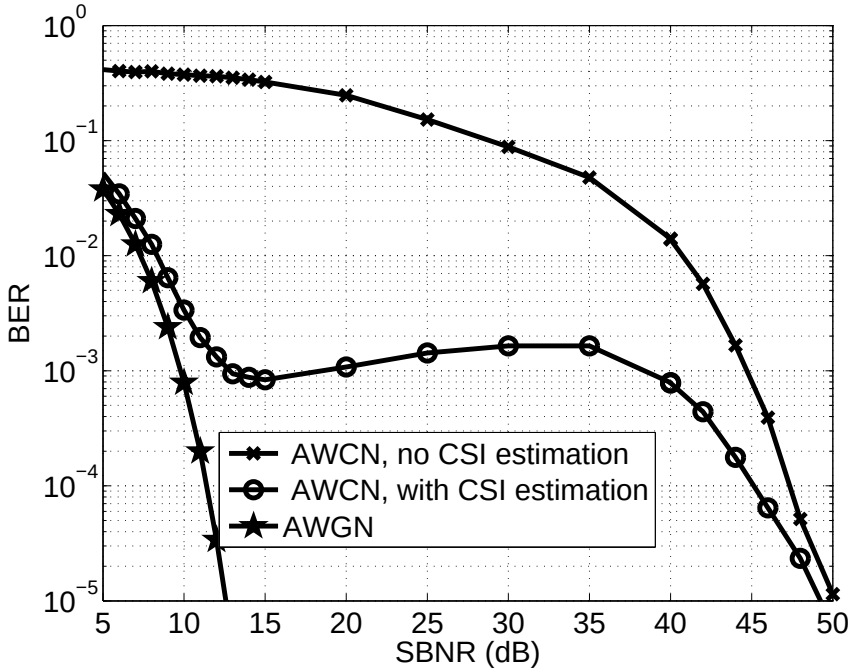

Figure 3.25: Uncoded noncoherent 2-FSK with ML block decoder for the AWCN channel ($A = 0.1, \Gamma = 10^{-3}$).

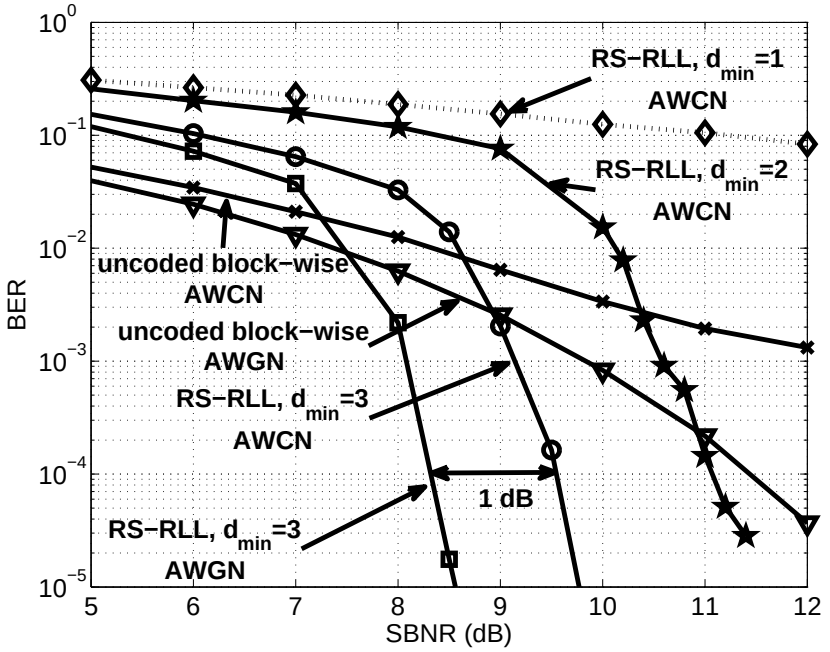

Figure 3.26: RS-RLL coded noncoherent 2-FSK with ML block decoder for the AWCN channel ($A = 0.1, \Gamma = 10^{-3}$).

3.9 ML Block Decoding of Noncoherent 2-FSK modulation for the Narrow Band Noise

The transmission quality of the powerline communication suffers heavily from the narrow band noise. Often, this type of noise is caused by television sets or computer terminals [14], and it can be permanent over a long period of time. We assume that when the narrow band noise exists, the SNR drops dramatically for one transmission frequency, i.e., one frequency is corrupted during the entire transmission, see Fig. 3.27. In the following analysis, a simple channel model is used. Non-white Gaussian noise η is added to the signal, where the variance of the noise at frequency f_0 and f_1 are $\sigma^2_{\eta_0}$ and $\sigma^2_{\eta_1}$, respectively. Moreover, the SNR at frequency f_0 and f_1 can be defined as $\mathrm{SNR}_0 = \frac{E_s}{2\sigma^2_{\eta_0}}$ and $\mathrm{SNR}_1 = \frac{E_s}{2\sigma^2_{\eta_0}}$, respectively. Under the assumption that the noise is Gaussian, the ML decoding metric can be calculated in the same way for the AWGN channel. However, the factors related with the SNR can not be neglected. Using Eq. (3.33), the ML decoding metric can be rewritten as

$$\max_{\mathbf{c}} p(\mathbf{r}|\mathbf{c}) = \max_{c\in C} \int_0^{2\pi} \prod_{k=1}^{n} e^{-\frac{E_s}{2\sigma^2_{\eta_{c_k}}}} e^{\frac{\sqrt{E_s}}{\sigma^2_{\eta_{c_k}}}(r_{c_k,k,I}\cos\phi + r_{c_k,k,Q}\sin\phi)} d\phi. \tag{3.54}$$

By applying logarithmic maximization and the steepest-descent approximation $I_0(x) = \exp(x^2/4)$, the ML decoding metric can be written as

$$\max_{\mathbf{c}} p(\mathbf{r}|\mathbf{c}) = -\sum_{k=1}^{n} \mathrm{SNR}_{c_k,k} + \left(\sum_{k=1}^{n} \sqrt{2\mathrm{SNR}_{c_k,k}} \frac{r_{c_k,k,I}}{\sigma_{\eta_{c_k}}}\right)^2 + \left(\sum_{k=1}^{n} \sqrt{2\mathrm{SNR}_{c_k,k}} \frac{r_{c_k,k,Q}}{\sigma_{\eta_{c_k}}}\right)^2. \tag{3.55}$$

where $\mathrm{SNR}_{c_k,k}$ represents the SNR at the transmitted frequency c_k for the time instant k. However, in the presence of severe narrow band noise, the energy of the interference surpasses the energy of the modulator, i.e., the demodulator will always have a large metric. Hence, no information can be extracted from the disturbed frequency. In that situation, the narrow band noise can be detected by a threshold detector, and the receiver ignores the disturbed frequency in the detection. Hence, in the case of 2-FSK, the decision is made observing the other frequency which can be seen as on-off keying (OOK). Consequently, the Euclidean distance between two signal points in the modified detection is reduced with a factor of $\sqrt{2}$.

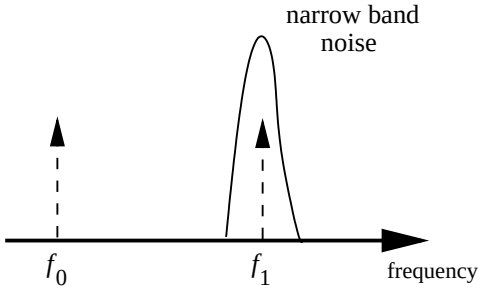

Figure 3.27: 2-FSK modulation and narrow band noise.

In Fig. 3.28, the simulated BER for the non-white Gaussian background noise is depicted. The x-axis denotes the SNR level at the frequency without narrow band noise. The results are obtained by assuming that the phase shift stays constant during a block transmission of 8 symbols. It can be seen that the OOK receiver has a good BER performance when the SNR-ratio between two frequencies exceeds 2 dB. The reduced Euclidean distance between two signal points gives rise to a loss of around 3,5 dB at BER 10^{-5} compared to the 2-FSK system with $\mathrm{SNR}_0/\mathrm{SNR}_1 = 0$ dB (AWGN channel).

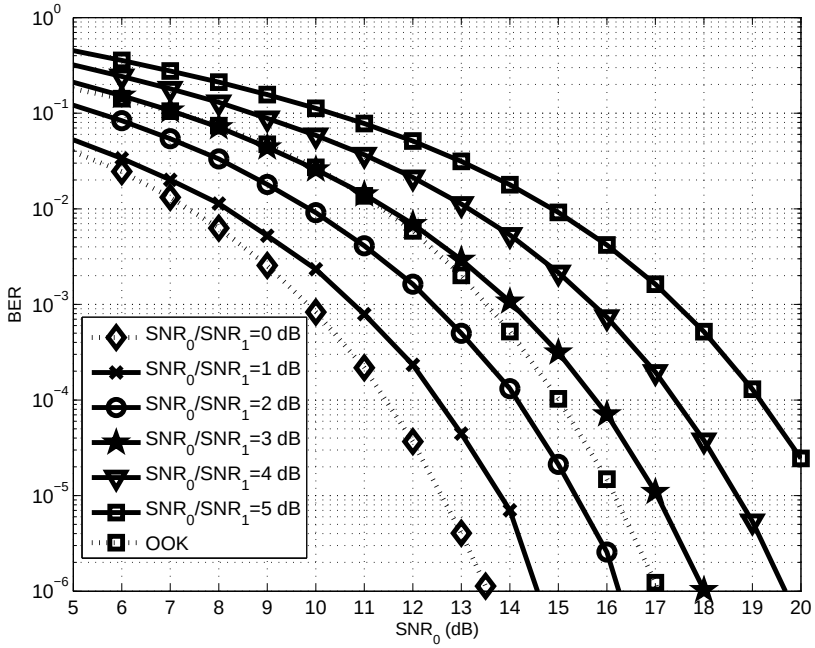

Figure 3.28: Uncoded noncoherent 2-FSK with ML block decoder for the narrow band noise channel.

3.10 RS-RLL Coded On-Off Keying for the AWCN Channel

As we discussed in Section 3.9, in the presence of narrow band noise, one transmission frequency is corrupted for a long period of time. Obviously, the optimum strategy can be to ignore the disturbed frequency in the detection. Then, the decision is made based on a single frequency, which can be considered as an OOK transmission. However, for the impulsive noise channel, the performance of block decoded OOK drops badly due to the dependency of the decision threshold on the signal amplitude and SNR. In order to overcome this problem, we simply declare a symbol erasure when the symbol is hit by the impulsive noise. In other words, the corrupted symbol is ignored in the block decoding. Since the impulsive index A is assumed to be 0.1, the impulse occurs once every 10 channel symbols, on the average. As a result, in a RS-RLL concatenated coded modulation during a transmission of one block or 21 RLL symbols, 2 erasures can be expected. Therefore, for a RLL block with $d_{\min} = 3$, a soft decision based RLL decoding can be expected to be able to correct the erasures for the high SBNR.

Fig. 3.29 illustrates the BER performances of the RS-RLL coded OOK and 2-FSK transmissions over the AWCN channel. For all curves, the ML block decoding is applied at the receiver. The BER performances over the AWGN channel are also added for comparison. At a BER of 10^{-4}, a 4.5 dB difference can be observed between the coded 2-FSK and the coded OOK transmission. In the OOK block decoding, noncoherent detection is done symbol-wise due to the threshold decision. Hence, this leads to a combination loss in block decoding. It can be also observed in Fig. 3.29 that erasure decoding for the symbols hit by impulsive noise performs quite well. Compared to the coded OOK Gauss bound, only a 1 dB gap can be observed.

3.11 Conclusion

A combined coding and modulation scheme was described which improves the error performance of constant envelope data transmission. RLL encoder is introduced between the RS encoder and the constant envelope modulator to control the minimum channel symbol duration. It was shown that the same amount of information can be transmitted within the same minimum channel symbol duration with coding gains of 4 and 5 dB at BER of 10^{-5} compared with the uncoded BPSK and 2-FSK modulation, respectively. Since the PSD of the signals is related to the minimum channel symbol duration, the PSD stays almost the same compared with the uncoded modulation. Furthermore, the scheme can be generalized to a multilevel transmission. However, a loss in the RS code rate can be observed as the symbol alphabet increases, see Section 3.7. It should also be noted that the additional complexity introduced by the RLL encoder can be reduced by selecting small RLL codeword lengths, see Table 3.4.

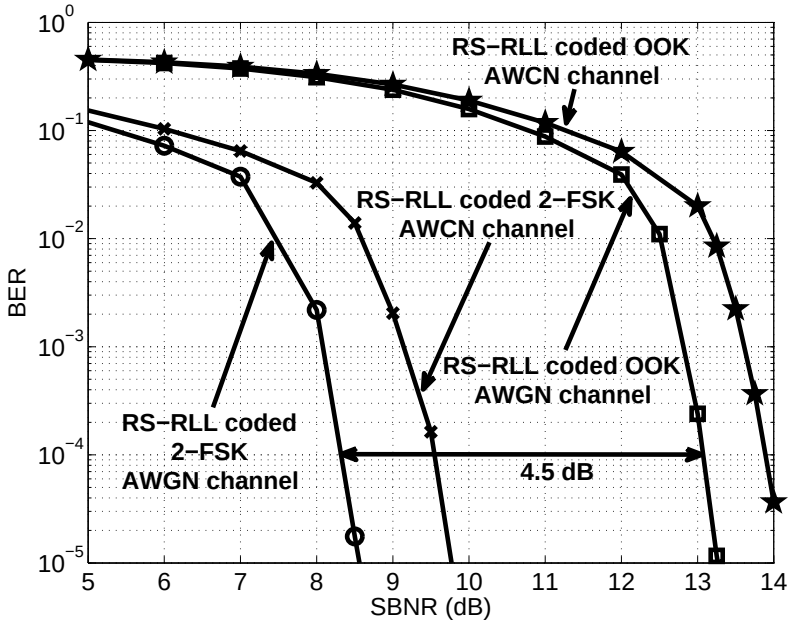

Figure 3.29: Simulated BER of RS-RLL coded 2-FSK and RS-RLL coded OOK for the AWCN channel ($A = 0.1,\ \sigma_g^2/\sigma_i^2 = 10^{-3}$).

As an application of the proposed scheme, we considered low frequency range PLC, where the output voltage and the bandwidth of the transmitters are limited. The performance of the PLC system is severely degraded by the permanent frequency disturbances and impulsive noise. We gave the derivation of the receiver structure using the Middleton Class-A model and showed that by estimating the channel state information, performance improvement can be obtained. Moreover, we addressed the effect of the narrow band noise on the 2-FSK transmission scheme. Simulation results showed that under a strong frequency disturbance, the decisions can be made based on a single frequency.

In addition to the points mentioned above, the following open problems remain for further research:

- The synchronization problem has not been studied in this chapter. As the parameter d of the RLL code increases, the sampling speed also increases which can be a limiting factor as in magnetic recording.
- Soft decision decoding of Reed Solomon codes might be introduced in the concatenated coding scheme so that soft information can be passed into the RS decoder.
- Instead of RS codes, convolutional codes might be used in the concatenated coding scheme. Since convolutional codes are not suitable for burst error correction, an interleaver can be used between the convolutional encoder and the RLL encoder.

Chapter 4

Runlength-Limited Code Construction Based on Minimum Hamming Distance

The number of bits between two consecutive transitions is called runlength. A sequence of 0's and 1's which satisfies a particular set of rules on the placement of symbols is called a constraint sequence. The d-constraint runlength-limited sequence is characterized as follows.

- d-constraint: Each run of zeros or ones must have a length of at least $d + 1$.

A variety of approaches to construct RLL codes is available in the literature. These include methods for constructing fixed and variable length codes and techniques which employ look-ahead encoding. A construction algorithm for the fixed length RLL codes with a one-symbol look-ahead encoding technique was proposed by Hollmann [20, 21]. This algorithm has the advantage that the constructed RLL codes have high rates, simple encoder tables and a block-wise decoder. They are highly suitable in conjunction with symbol error correcting codes, such as Reed-Solomon (RS) codes.

In this chapter, we describe construction algorithms for RLL codes with one-symbol look-ahead encoding techniques having a minimum Hamming distance, shortly $d_{\min}$, larger than one. The designed codes enable soft information to be available at the RLL decoder. Hence, a soft decoding can be used to improve the error rate performance of the system. Our proposed constructions are based on Hollmann's code construction algorithm. We present several new high rate RLL codes with specified d-constraint, $d_{\min}$ and code rate. We also provide efficient bounds on the code sizes. The constructed codes are used in the RLL coded modulation scheme which is described in Chapter 3.

The rest of the chapter is organized as follows. In Section 4.1, we describe and generalize the RLL code construction approach given in [20, 21]. The RLL code construction methods for $d_{\min} > 1$ are proposed in Section 4.2. The final section summarizes the main results obtained in this chapter.

4.1 Look-Ahead RLL Code Construction In Terms of Principal State-Sets

In this section, the look-ahead RLL code construction in [20] is described for $d = 1$ and generalized for $d = 2$ and $d = 3$. For ease of presentation, we will first focus on the design of RLL codes with $d = 1$.

4.1.1 One-Bit Look-Ahead RLL Code Construction For $d = 1$

Suppose that we want to design a block-decodable look-ahead RLL code at a given rate of $R_r = m/n$. Since the RLL encoded block complies with the d-constraint, the binary strings in Table 4.1 can not occur at the output of the encoder. As an example, the forbidden strings

Table 4.1: Forbidden strings.

$\{010\}$	$\cdots$	$\{01^d0\}$
$\{101\}$	$\cdots$	$\{10^d1\}$

for $d = 1$ can be written as $\{010\}$, $\{101\}$. Hollmann [20, 21] showed that a classification of codewords can be done according to their begin and end types, where they are defined according to the forbidden strings and labeled by the number of initial and final zeros (or ones). The begin and end types together constitute the type of the codeword. Consider the following example.

Example 4.1. The begin and end types for the $(d = 1)$-constraint are shown in Tables 4.2 and 4.3. Since the forbidden strings for $d = 1$ can be written as $\{010\}$ and $\{101\}$, it is enough to consider four begin and four end types.

Table 4.2: The begin types for $d = 1$.

Begin type	Initial part of the codeword
$z1$	01
$z2$	00
$o1$	10
$o2$	11

Table 4.3: The end types for $d = 1$.

End type	Final part of the codeword
$z1$	10
$z2$	00
$o1$	01
$o2$	11

The type of codeword can be represented as **begin-end**. For example, the codeword (01100) is represented as **z1-z2**. We now show that codeword types can be used to formulate successor rules, principal state-sets and code classes.

Successor relations: In the one symbol look-ahead code, encoding is done such that the concatenation of two RLL codewords also satisfies the given d-constraint. Such a concatenated sequence of length $2n$ is called admissible. If a concatenated sequence $\mathbf{cc}'$ of a codeword $\mathbf{c}$ with a codeword $\mathbf{c}'$ is admissible, then the codeword $\mathbf{c}'$ is a successor of the codeword $\mathbf{c}$. For example, since the concatenation $(01100 - 01110)$ is admissible for the $(d = 1)$-constraint, the codeword (01110) is a successor of the codeword (01100). The successor relations of a code can be described by the type of the codewords. For example a

codeword with end type $o1$ can be only followed by the codewords that have begin types of $o1$ or $o2$, while codewords that end with $z2$ can be followed by three begin types without violating the $(d = 1)$-constraint. Table 4.4 summarizes the successor relations of the $(d = 1)$-constraint.

Table 4.4: Successor relations of $(d = 1)$-constraint.

End type	begintype			
$z1$	$z1$	$z2$	-	-
$z2$	$z1$	$z2$	-	$o2$
$o1$	-	-	$o1$	$o2$
$o2$	-	$z2$	$o1$	$o2$

Principal state-sets and cover lists: The principal state-sets and the cover lists can be designed using the successor relations. The principal state-sets are described in terms of end types, and the cover lists are described in terms of begin types. They are chosen such that each list from the cover lists can be concatenated freely to any state from the principal states. Let us consider Table 4.5 which illustrates the cover lists and principle state-sets for the $(d = 1)$-constraint. It can be seen that the begin type $z2$ forms a cover list, i.e.,

Table 4.5: Cover list and principal state-sets.

Cover lists	Principal state-sets
$z2$	$z2$
$o2$	$o2$
$z1$ & $o1$	$z1$ & $o1$

$z2$ is a successor to each principal state-set. In other words, each of the principal state-set contains at least one end type which can be concatenated freely with the begin type $z2$. This describes a look-ahead assignment. One of the end type is chosen depending on the upcoming begin type in the cover list.

Potential codeword class types: The next step is to obtain the set of potential codeword class types by forming codewords with respect to the cover lists and principal state-sets. Each cover list is combined with the principal state-sets. For example, the set $\{(\mathbf{z2\text{-}z1}), (\mathbf{z2\text{-}o1})\}$ is a potential codeword class type, since we cascaded the cover list z_2 with the principal state-set ($z1$ & $o1$). After listing the potential codeword class types, the total number of available classes, i.e., $N_c(n, d)$, should be verified by calculating the number of codeword classes of each type. Table 4.6 illustrates the potential codeword class types according to the principal state-sets and cover lists together with the number of codeword classes of each type of length 5, see also Table 3.3. It can be seen that there are 10 codeword classes of length 5. Thus, a rate $\log_2(10/5)$ look-ahead code is possible with the above construction.

Table 4.6: Code class types

Index	Code class type	Code	multiplicity
1	(**z2-z2**)	00000	1
2	(**z2-o2**)	00011 00111	2
3	(**o2-z2**)	11100 11000	2
4	(**o2-o2**)	11111	1
5	(**z2-z1**) or (**z2-o1**)	00110 or 00001	1
6	(**o2-z1**) or (**o2-o1**)	11110 or 11001	1
7	(**z1-z2**) or (**o1-z2**)	01100 or 10000	1
8	(**z1-o2**) or (**o1-o2**)	01111 or 10011	1

In Table 4.6, some codewords can uniquely identify the source word, while some codewords have alternatives. The ability of looking one bit ahead and back enables the encoder to decide on an alternative codeword which maintains the $(d = 1)$-constraint in the concatenation. Therefore, each decoding is done block-wise, which simplifies the decoding complexity and avoids the propagation between the blocks. Note that not all the possible code class types are shown in Table 4.6, since their multiplicities are equal to zero. In general, $(d = 1)$-constraint RLL code can be constructed using the potential codeword class types as given in Table 4.7.

Table 4.7: Code class types for the construction of $(d = 1)$-constraint RLL code.

Subset	Code class types
1	$\begin{pmatrix} (\mathbf{z2\text{-}z2}) \\ (\mathbf{z2\text{-}o2}) \\ (\mathbf{o2\text{-}z2}) \\ (\mathbf{o2\text{-}o2}) \end{pmatrix}$
2	$\begin{pmatrix} (\mathbf{z2\text{-}z1}) \\ (\mathbf{o2\text{-}z1}) \end{pmatrix}$ or $\begin{pmatrix} (\mathbf{z2\text{-}o1}) \\ (\mathbf{o2\text{-}o1}) \end{pmatrix}$
3	$\begin{pmatrix} (\mathbf{z1\text{-}z2}) \\ (\mathbf{z1\text{-}o2}) \end{pmatrix}$ or $\begin{pmatrix} (\mathbf{o1\text{-}z2}) \\ (\mathbf{o1\text{-}o2}) \end{pmatrix}$
4	(**z1-z1**) or (**z1-o1**) or (**o1-z1**) or (**o1-o1**)

The Total Number of Codeword Classes For $d = 1$

A recursive expression without a proof is given in [26] to calculate the number of codeword classes, $N_c(n, 1)$. In this subsection, we obtained the same result using Table 4.7. It can be observed that the set of $(d = 1)$-constraint RLL code class types can be divided into four subsets. The total number of codeword classes, $N_c(n, 1)$, can be found by counting the number of codewords in each subset. For $n > 2$, $N_c(n, 1)$ can be calculated by the relation

$$\begin{aligned} N_c(n,1) &= N(n-2,1) + N(n-3,1) + \lfloor N(n-4,1)/4 \rfloor, \\ &= N(n-1,1) + \lfloor N(n-4,1)/4 \rfloor, \end{aligned} \tag{4.1}$$

where $N(n < 0, 1) = 0$, $N(0, 1) = 2$ and see Eq. (3.2) and Table 3.1 for $N(n > 0, 1)$. The proof of Equation (4.1) is as follows.

Subset-1. Each class type in subset-1 in Table 4.7 consists of one codeword starting and ending with 00 or 11. For $n > 2$, the codewords of length n can be constructed by repeating the first and the last bit of the $(d = 1)$-constraint RLL codeword of length $n - 2$. Thus, there are $N(n - 2, 1)$ codewords and $N(n - 2, 1)$ classes of such. For example, there are altogether six codewords of length five starting and ending with 00 or 11. They can be formed by the RLL codewords of length three, i.e., $\{(0\mathbf{000}0), (0\mathbf{001}1), ..., (1\mathbf{111}1)\}$.

Subset-2. Each class type in subset-2 consists of two codewords starting with 00 or 11 and ending with 01 or 10. For $n > 3$, the codewords of length n can be constructed by repeating the first bit and appending 01 or 10 (depending on the last bit) at the end of the $(d = 1)$-constraint RLL codeword of length $n - 3$. Thus, there are $N(n - 3, 1)$ codewords and $N(n - 3, 1)/2$ classes of such. For example, there are altogether four codewords of length five starting with 00 or 11 and ending with 01 or 10. They can be formed by the RLL codewords of length two, i.e., $\{(0\mathbf{00}01), (0\mathbf{01}10), (1\mathbf{10}01), (1\mathbf{11}10)\}$.

Subset-3. Each class type in subset-3 consists of two codewords starting with 01 or 10 and ending with 00 or 11. For $n > 3$, the codewords of length 5 can be constructed by appending 10 or 01 at the beginning and repeating the last bit of the $(d = 1)$-constraint RLL codeword of length $n - 3$. Thus, there are $N(n - 3, 1)$ codewords and $N(n - 3, 1)/2$ classes of such. For example, there are altogether four codewords of length five starting with 01 or 10 and ending with 00 or 11. They can be formed by the RLL codewords of length two, i.e., $\{(01\mathbf{10}0), (01\mathbf{11}1), (10\mathbf{00}0), (10\mathbf{01}1)\}$.

Subset-4. Each class type in subset-4 consists of four codewords starting and ending with 01 or 10. For $n > 4$, the codewords of length n can be constructed by appending 01 or 10 at the beginning and at the end of the $(d = 1)$-constraint RLL codeword of length $n - 4$. Thus, there are $N(n - 4, 1)$ codewords and $\lfloor N(n - 4, 1)/4 \rfloor$ classes of such. For example, there are altogether two codewords of length five starting and ending with 01 or 10. They can be formed by the RLL codewords of length one, i.e., $\{(10\mathbf{0}01), (10\mathbf{1}10)\}$.

Rate efficiency: The rate efficiency is defined with respect to the asymptotic information rate of the code, and it is given by

$$E(n,d) = \frac{R_r}{C_d}, \tag{4.2}$$

see Table 3.2 for C_d as a function of d.

Table 4.8 shows $N_c(n,1)$ and the source word length m of the RLL code as a function of codeword length. The source word length is calculated by truncating the number of codewords to the smallest power of two. Since the rate $R_r = m/n$ cannot exceed the asymptotic information rate C_d, the upper bound on the maximum source word length m, defined as $\lfloor nC_d \rfloor$, is also added. It can be observed in Table 4.6 that the smallest rate 2/3, $(d = 1)$-constraint RLL block-decodable code has a codeword length $n = 3$ with a very high rate efficiency $E(3,1) = 0.9603$. The code rates can achieve the upper bound on the maximum source word length m, except in the case of $n = 13$.

Table 4.8: The code size $N_c(n,1)$ and the source word length m for $(d = 1)$-constraint RLL block codes of length n.

n	$N_c(n,1)$	$m = \lfloor \log_2 N_c(n,1) \rfloor$	$\lfloor nC_{d=1} \rfloor$
3	4	2	2
4	6	2	2
5	10	3	3
6	17	4	4
7	27	4	4
8	44	5	5
9	72	6	6
10	116	6	6
11	188	7	7
12	305	8	8
13	493	8	9
14	798	9	9

4.1.2 One-bit Look-Ahead RLL Code Construction For $d = 2$

In this subsection, we generalize the one-bit look-ahead code construction algorithm to the $(d = 2)$-constraint. The begin and end types can be illustrated as in Tables 4.9 and 4.10. Using the codeword types, the cover lists and the principal state-sets can be formed as in Table 4.11. Then, the potential codeword classes can be obtained by forming the codewords with respect to the cover list and principal state-sets, see Table 4.12.

Table 4.9: The begin types for $d = 2$.

Begin type	Initial part of the codeword
$z1$	01
$z2$	001
$z3$	000
$o1$	10
$o2$	110
$o3$	111

Table 4.10: The end types for $d = 2$.

End type	Final part of the codeword
$z1$	10
$z2$	100
$z3$	000
$o1$	01
$o2$	011
$o3$	111

Table 4.11: Principal state-sets and cover list.

Cover lists	Principal State-sets
$z3$	$z3$
$o3$	$o3$
$z1$ & $o1$	$z1$ & $o2$
$z1$ & $o2$	$z2$ & $o1$
$z2$ & $o1$	
$z2$ & $o2$	

The Total Number of Codeword Classes For $d = 2$

Due to the tedious calculations, Eq. (4.1) can only be generalized to $d = 2$ by considering the classes that are formed by one or two codewords. The number of codeword classes for $d = 2$ can be lower bounded as

$$\begin{aligned} N_c(n,2) &\geq N(n-4,2) + (N(n-5,2) + N(n-6,2))/2 + N(n-6,2), \\ &\geq N(n-3,2) + (N(n-5,2) + N(n-6,2))/2, \end{aligned} \tag{4.3}$$

where $N(n < 0, 2) = 0$, $N(0, 2) = 2$, and see Eq. (3.2) and Table 3.1 for $N(n > 0, 2)$. The proof of Eq. (4.3) is as follows.

Subset-1. Each class type in subset-1 in Table 4.12 consists of one codeword starting and ending with 000 or 111. For $n > 4$, the codewords of length n can be constructed from the $(d = 2)$-constraint RLL codeword of length $n - 4$ by repeating the first and the last bit two times. Thus, there are $N(n - 4, 2)$ codewords and $N(n - 4, 2)$ classes of such.

Subset-2. Each class type in subset-2 consists of two codewords. For $n > 5$, the codewords of length n starting with 01 or 10 can be constructed by appending three bits to the beginning and repeating the last bit two times at the end of the $(d = 2)$-constraint RLL codeword of length $n - 5$. For $n > 6$, the codewords of length n starting with 001 or 110 can be constructed by appending four bits to the beginning and repeating the last bit two times at the end of the $(d = 2)$-constraint RLL codeword of length $n-6$. Thus, there are altogether $N(n-5, 2)+N(n-6, 2)$ codewords and $(N(n-5, 2)+N(n-6, 2))/2$ classes of such.

Table 4.12: Code class types for $(d = 2)$-constraint.

Index	Code class types
1	$\begin{pmatrix} \textbf{(z3-z3)} \\ \textbf{(z3-o3)} \\ \textbf{(o3-z3)} \\ \textbf{(o3-o3)} \end{pmatrix}$
2	$\begin{pmatrix} \textbf{(z1-z3)} \\ \textbf{(z1-o3)} \\ \textbf{(z2-z3)} \\ \textbf{(z2-o3)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(o1-z3)} \\ \textbf{(o1-o3)} \\ \textbf{(o2-z3)} \\ \textbf{(o2-o3)} \end{pmatrix}$
3	$\begin{pmatrix} \textbf{(z3-z2)} \\ \textbf{(o3-z2)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(z3-o1)} \\ \textbf{(o3-o1)} \end{pmatrix}$
4	$\begin{pmatrix} \textbf{(z3-z1)} \\ \textbf{(o3-z1)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(z3-o2)} \\ \textbf{(o3-o2)} \end{pmatrix}$
5	$\begin{pmatrix} \textbf{(z3-o1)} \\ \textbf{(o3-o1)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(z1-z2)} \\ \textbf{(z2-z2)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(o1-z2)} \\ \textbf{(o2-z2)} \end{pmatrix}$
6	$\begin{pmatrix} \textbf{(z3-z1)} \\ \textbf{(o3-z1)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(z1-o2)} \\ \textbf{(z2-o2)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(o1-o2)} \\ \textbf{(o2-o2)} \end{pmatrix}$
7	$\begin{pmatrix} \textbf{(z1-z1)} \\ \textbf{(z2-z1)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(z1-o2)} \\ \textbf{(22-o2)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(o1-z2)} \\ \textbf{(o2-z2)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(o1-o1)} \\ \textbf{(o2-o1)} \end{pmatrix}$
8	$\begin{pmatrix} \textbf{(z1-z2)} \\ \textbf{(z2-z2)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(z1-o1)} \\ \textbf{(z2-o1)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(o1-z1)} \\ \textbf{(o2-z1)} \end{pmatrix}$ or $\begin{pmatrix} \textbf{(o1-o2)} \\ \textbf{(o2-o2)} \end{pmatrix}$

Subset-3. Each class type in subset-3 consists of two codewords starting with 000 or 111 and ending with 100 or 01. For $n > 5$, the codewords of length n ending with 01 can be constructed by appending two bits to the beginning and three bits to the last bit of the $(d = 2)$-constraint RLL codeword of length $n - 5$. For $n > 6$, the codewords of length n ending with 100 can be constructed by appending two bits to the beginning and four bits to the last bit of the $(d = 2)$-constraint RLL codeword of length $n - 6$. Thus, there are $N(n-5,2)/2 + N(n-6,2)/2$ codewords. Since $N(n-6,2) \leq N(n-5,2)$, there are $N(n-6,2)/2$ classes of such.

Subset-4. Each class type in subset-4 consists of two codewords starting with 000 or 111 and ending with 011 or 10. For $n > 5$, the codewords of length n ending with 10 can be constructed by appending two bits to the beginning and three bits to the last bit of the $(d = 2)$-constraint RLL codeword of length $n - 5$. For $n > 6$, the codewords of length n ending with 011 can be constructed by appending two bits to the beginning and four bits to the last bit of the $(d = 2)$-constraint RLL codeword of length $n - 6$. Thus, there are $N(n-5,2)/2 + N(n-6,2)/2$ codewords, and since $N(n-6,2) \leq N(n-5,2)$, $N(n-6,2)/2$

classes of such.

Table 4.13 illustrates the number of codeword classes calculated with the help of a computer search. We also add the lower bound (LB) given in Eq. (4.3), the source word length $m = \lfloor \log_2 N_c(n,2) \rfloor$ and the upper bound on the source word length, i.e., $\lfloor nC_{d=2} \rfloor$.

Table 4.13: Code size, $N_c(n,d)$, versus n for $d = 2$-constraint.

n	LB	$N_c(n,2)$	m	$\lfloor nC_{d=2} \rfloor$
4	2	4	2	2
5	5	6	2	2
6	8	8	2	3
7	11	12	3	3
8	17	19	4	4
9	25	28	4	4
10	36	41	5	5
11	53	60	5	6
12	78	88	6	6
13	114	129	7	7
14	167	189	7	7
15	245	277	8	8
16	359	406	8	8
17	526	595	9	9
18	771	872	9	9

Table 4.13 reveals that the code rate $R_r = 7/13$ is highly attractive as it has a code rate efficiency of $E(13,2) = 0.9764$. The next higher code rates are 8/15 and 9/17 with code rate efficiencies of 0.9671 and 0.9599, respectively.

4.1.3 One-bit Look-Ahead RLL Code Construction For $d = 3$

Tables 4.14 and 4.15 illustrate the begin and end types for the ($d = 3$)-constraint. The code construction procedure as described in Subsection 4.1.1 can be applied to find the principal state-sets and the cover lists, see Table 4.16. Since, there is a large number of potential codeword class types with respect to the cover lists and principal state-sets, Table 4.17 considers only the codeword class types that consist of one or two codewords.

Table 4.14: The begin types for $d = 3$.

Begin type	Initial part of the codeword
$z1$	01
$z2$	001
$z3$	0001
$z4$	0000
$o1$	10
$o2$	110
$o3$	1110
$o4$	1111

Table 4.15: The end types for $d = 3$.

End type	Final part of the codeword
$z1$	10
$z2$	100
$z3$	1000
$z4$	0000
$o1$	01
$o2$	011
$o3$	0111
$o4$	1111

Table 4.16: Principal state-sets and cover list.

Cover lists	Principal State-sets
$z4$	$z4$
$o4$	$o4$
$z1$ & $o2$	$z1$ & $o3$
$z1$ & $o3$	
$z2$ & $o1$	$z2$ & $o2$
$z2$ & $o2$	
$z2$ & $o3$	$z2$ & $o3$
$z3$ & $o1$	$z3$ & $o1$
$z3$ & $o2$	$z3$ & $o2$
$z3$ & $o3$	$z3$ & $o3$

The Total Number of Codeword Classes For $d = 3$

Due to the tedious calculations, Eq. (4.1) can be only generalized to $d = 3$ by considering the classes that are formed by up to two codewords, see Table 4.17. The number of codeword classes for $d = 3$ can be lower bounded as

$$N_c(n, 3) \geq N(n-5, 3) + N(n-8, 3) + (N(n-7, 3) + N(n-9, 3))/2, \qquad (4.4)$$

where $N(n < 0, 3) = 0$, $N(0, 3) = 2$ and see Eq. (3.2) and Table 3.1 for $N(n > 0, 3)$. The proof of Eq. (4.4) is similar to the proof of Equations (4.1) and (4.3).

Table 4.18 illustrates the number of codeword classes that is calculated with the help of a computer search. We also add the lower bound (LB) given in Eq. (4.4), the source word length $m = \lfloor \log_2 N_c(n, 3) \rfloor$ and the upper bound on the source word length, i.e., $\lfloor nC_{d=3} \rfloor$. Table 4.18 for $d = 3$ reveals that the rates 8/18, 7/16 and 3/7 are suitable candidate rates for the creation of high rate codes. The code of rate 8/18 has a code rate efficiency of 0.9558.

Table 4.17: Code class types for $(d = 3)$-constraint.

Index	Code class types
1	$\begin{pmatrix} (\mathbf{z4\text{-}z4}) \\ (\mathbf{z4\text{-}o4}) \\ (\mathbf{o4\text{-}z4}) \\ (\mathbf{o4\text{-}o4}) \end{pmatrix}$
2	$\begin{pmatrix} (\mathbf{z1\text{-}z4}) \\ (\mathbf{z1\text{-}o4}) \end{pmatrix}$ or $\begin{pmatrix} (\mathbf{o2\text{-}z4}) \\ (\mathbf{o2\text{-}o4}) \\ (\mathbf{o3\text{-}z4}) \\ (\mathbf{o3\text{-}o4}) \end{pmatrix}$
3	$\begin{pmatrix} (\mathbf{z2\text{-}z4}) \\ (\mathbf{z2\text{-}o4}) \\ (\mathbf{z3\text{-}z4}) \\ (\mathbf{z3\text{-}o4}) \end{pmatrix}$ or $\begin{pmatrix} (\mathbf{o1\text{-}z4}) \\ (\mathbf{o1\text{-}o4}) \\ (\mathbf{o2\text{-}z4}) \\ (\mathbf{o2\text{-}o4}) \\ (\mathbf{o3\text{-}z4}) \\ (\mathbf{o3\text{-}o4}) \end{pmatrix}$
4	$\begin{pmatrix} (\mathbf{z4\text{-}z1}) \\ (\mathbf{o4\text{-}z1}) \end{pmatrix}$ or $\begin{pmatrix} (\mathbf{z4\text{-}o3}) \\ (\mathbf{o4\text{-}o3}) \end{pmatrix}$
5	$\begin{pmatrix} (\mathbf{z4\text{-}z2}) \\ (\mathbf{o4\text{-}z2}) \end{pmatrix}$ or $\begin{pmatrix} (\mathbf{z4\text{-}o2}) \\ (\mathbf{o4\text{-}o2}) \end{pmatrix}$
6	$\begin{pmatrix} (\mathbf{z4\text{-}z3}) \\ (\mathbf{o4\text{-}z3}) \end{pmatrix}$ or $\begin{pmatrix} (\mathbf{z4\text{-}o1}) \\ (\mathbf{o4\text{-}o1}) \end{pmatrix}$

4.1.4 Discussion

Fig. 4.1 shows the code rate efficiency $E(n, d)$ of block-decodable look-ahead RLL codes of length n as a function of $N_c(n, d)$. The code efficiency was calculated by the total number of available classes, i.e., no truncation to a power of two. The efficiency approaches to unity with increasing codeword length n.

Table 4.18: Code size, $N_c(n,d)$, versus n for $d=3$-constraint.

n	LB	$N_c(n,3)$	m	$\lfloor nC_{d=3} \rfloor$
4	-	2	1	1
5	2	4	2	2
6	2	5	2	3
7	5	8	3	3
8	9	10	3	3
9	13	13	3	4
10	18	19	4	4
11	26	29	4	5
12	36	42	5	5
13	49	55	5	6
14	67	75	6	6
15	93	107	6	6
16	129	151	7	7
17	178	212	7	7
18	245	289	8	8

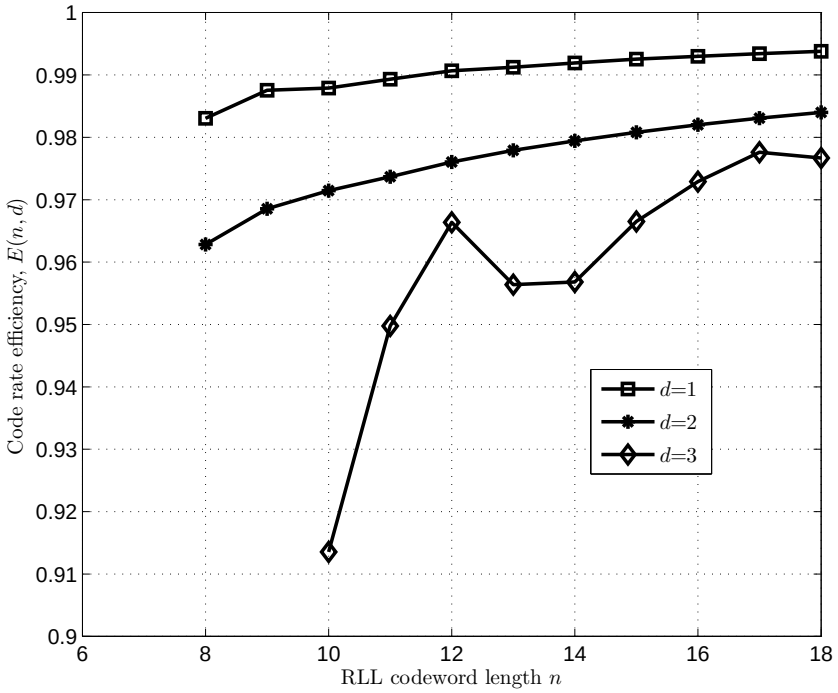

Figure 4.1: Code rate efficiency for $d_{\min}=1$.

4.2 Block-Decodable RLL Codes with $d_{\min} > 1$

In this section, we describe the construction of block-decodable look-ahead RLL codes with $d_{\min} > 1$. Several new codes, with specified d-constraint and $d_{\min}$ are presented. The designed codes enable soft information to be available at the RLL decoder. Hence, a soft decoding can be used to improve the error rate performance of the system. Codes with $d_{\min} = 2$ are obtained from a simple construction which partitions the admissible RLL codewords into two subsets having even and odd parities, respectively. For $d_{\min} > 2$, two constructions are described and compared. The first construction is based on Gilbert Type Code Construction. The second construction is an extension of the first one, where the look-ahead code classes are considered for the code search. Note that, the topic of combined error correction and RLL coding has been studied in several papers, see [29–33].

4.2.1 One-bit Look-Ahead RLL Codes with $d_{\min} = 2$

Given the number of input bits, m, and an RLL constraint d, a rate m/n code with $d_{\min} = 2$ can be constructed as follows. The admissible d-constraint RLL codewords are partitioned into two subsets B_e and B_o, where both satisfy the constraint and have even and odd parities, respectively. Thus, for two different codewords in B_e or B_o, the Hamming distance is greater than or equal to two, i.e., $d_{\min}(B_e) = 2$ and $d_{\min}(B_o) = 2$. The total number of codeword classes in a code with $d_{\min} = 2$ is equal to $N(n, d, d_{\min} = 2) = \max(N_e, N_o)$, where N_e and N_o are the numbers of classes in even and odd subsets, respectively. Thus, the code rate of the proposed code is

$$R_r = \frac{\lfloor N(n, d, d_{\min} = 2) \rfloor}{n}. \tag{4.5}$$

Lower bound on the number of classes

A lower-bound can be calculated by considering the classes that are formed by only one codeword. Thus, the number of codeword classes for $d_{\min} = 2$ can be lower bounded as

$$N(n, 2, d_{\min} = 2) \geq N(n - 2d, d)/2, \tag{4.6}$$

where $N(n < 0, d) = 0$, $N(0, d) = 2$ and see Eq. (3.2) and Table 3.1 for $N(n > 0, d)$. The proof of Eq. (4.6) is as follows. Each class type that consists of one codeword starts and ends with $(d + 1)$ zeros and ones. For $n > 2d$, there are $N(n - 2, d)$ codewords and $N(n - 2, d)$ classes of such. After partitioning into even and odd subsets, one subset must have $\geq N(n - 2d, d)/2$ classes.

Tables 4.19, 4.20 and 4.21 tabulate the code size $N(n, d, d_{\min} = 2)$ as a function of n for $d = 1$, $d = 2$ and $d = 3$, respectively. The lower bound (LB) given in Eq. (4.6) and the source word length calculated as $m = \lfloor \log_2 N(n, d, d_{\min} = 2) \rfloor$ are also depicted in the table. For $d = 1$, see Table 4.19, the rates 6/11, 7/12, 8/14, 9/15, 10/16, 11/18 and 12/19 are suitable candidates for $d_{\min} = 2$. Efficiency-wise speaking, the code of rate 12/19 looks more attractive with rate efficiency $E(19, 1, d_{\min} = 2) = 0.9098$, but the code is too complex for an implementation. For $d = 2$, see Table 4.19, it is possible to construct codes with rates 6/13, 7/15, 8/17, 9/19, 10/21. The rate efficiency of the rate 6/13 code is 0.8396, which reveals that we can still have good rate efficiencies with less complex codes (only 64 classes). For $d = 3$, see Table 4.21, a rate 8/20 code can be constructed with a rate efficiency of

0.8602. The code rate efficiency, $E(n, d, d_{\min} = 2) = \lfloor \log_2 N(n, d, d_{\min} = 2) \rfloor / C_d$, of various codes versus codeword length is plotted in Fig. 4.2. Examination of the curves shows that construction of one-bit look-ahead RLL codes with an efficiency above 0.85 is possible with a realizable codeword length.

Table 4.19: Code size $N(n, d = 1, d_{\min} = 2)$.

n	LB	$N(n, d = 1, d_{\min} = 2)$	m
5	3	4	2
6	5	10	3
7	8	11	3
8	13	25	4
9	21	32	5
10	34	62	5
11	55	88	6
12	89	159	7
13	144	236	7
14	233	410	8
15	377	629	9
16	610	1062	10
17	987	1664	10
18	1597	2764	11
19	2584	4383	12
20	4181	7209	12

Table 4.20: Code size $N(n, d = 2, d_{\min} = 2)$.

n	LB	$N(n, d = 2, d_{\min} = 2)$	m
6	2	4	2
7	3	6	2
8	4	10	3
9	6	13	3
10	9	22	4
11	13	29	4
12	19	46	5
13	28	64	6
14	41	96	6
15	60	138	7
16	88	203	7
17	129	296	8
18	189	438	8
19	277	637	9
20	406	940	9
21	595	1371	10
22	872	2015	10

Table 4.21: Code size $N(n, d = 3, d_{\min} = 2)$.

n	LB	$N(n, d = 2, d_{\min} = 2)$	m
10	4	11	3
11	5	13	3
12	7	23	4
13	10	26	4
14	14	42	5
15	19	50	5
16	26	80	6
17	36	100	6
18	50	152	7
19	69	191	7
20	95	284	8
21	131	370	8
22	181	544	9

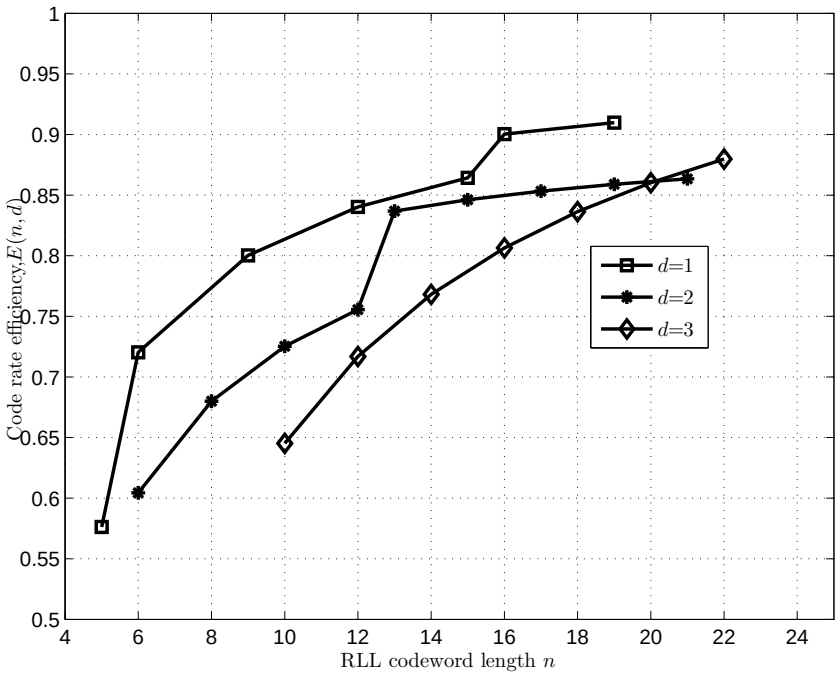

Figure 4.2: Code rate efficiency for $d_{\min} = 2$.

4.2.2 One-bit Look-Ahead RLL Codes with $d_{\min} > 2$

We discuss two construction methods of look-ahead RLL codes with $d_{\min} > 2$. Construction 1 is based on the Gilbert type code construction. The principle of Gilbert type code construction can be summarized as follows.

1. Select a codeword from the set of all admissible d-constraint RLL codes of length n.
2. Remove all codewords at distance $< d_{\min}$ from the set.
3. Subsequently go to the first step and continue until the complete set of codewords has been found by purging the original set of all d-constraint RLL codes of length n.

At every iteration, one codeword is chosen randomly in the remaining subset. Since, it is desirable to remove less number of codewords at distance $< d_{\min}$ from the set, Construction 1 can be obtained by extending the algorithm.

Construction 1 for $d_{\min} > 2$

A block-decodable look-ahead RLL code with a desirable $d_{\min}$ can be formed as follows.

1. Select the codeword which has the largest subset after removing all codewords at distance $< d_{\min}$ from the main set. In this case, we calculate for each codeword the size of the remaining subset. Then, we select the codeword with the largest subset.
2. Remove all codewords at distance $< d_{\min}$ from the set.
3. Subsequently go to the first step and continue until the complete set of codewords has been found by purging the original set of all d-constraint RLL codes of length n.

At the last step, the total number of available classes $N(n, d, d_{\min})$ should be verified by calculating the number of codeword classes of each type in the remaining subset. In that way, the blocks of a look-ahead RLL code can be characterized by the minimum distance. Table 4.22 tabulates the code size $N(n, d, d_{\min} = 3)$ as a function of n and d. As we may observe, for $d = 1$, it is possible to construct a rate 9/18 code with $d_{\min} = 3$ with a rate efficiency of 0.72. The rate 8/21 for $d = 2$ and the rate 6/20 for $d = 3$ are the suitable candidate rates for $d_{\min} = 3$.
Construction 1 has the drawback that not all the selected codewords can form classes. In principle, some codewords may not contribute to the number of codeword classes. The codeword selection process can be improved by considering the codeword class types.

Table 4.22: Code size $N(n, d, d_{\min} = 3)$ based on Construction 1.

n	$N(n, d = 1, d_{\min} = 3)$	$N(n, d = 2, d_{\min} = 3)$	$N(n, d = 3, d_{\min} = 3)$
8	8	4	2
9	14	8	4
10	17	7	4
11	29	12	7
12	39	16	8
13	70	24	10
14	94	35	15
15	158	44	21
16	223	62	26
17	354	89	32
18	529	126	47
19	845	171	61
20	1232	238	82
21	-	329	105

Construction 2 for $d_{\min} > 2$

A block-decodable look-ahead RLL code with a desirable $d_{\min}$ can be formed as follows.

1. The admissible d-constraint RLL codewords are partitioned into subsets depending on the number of codewords in their class types. We collect the class types formed by one codeword in the first subset, class types formed by two codewords in the second subset and so on.

2. Select a codeword in the first subset which yields the largest number of codewords after removing all codewords in the main set at distance $< d_{\min}$.

3. Remove all codewords at distance $< d_{\min}$ from the main set.

4. Subsequently go to the second step and continue until the complete set of codewords in the first subset has been purged. Here, since each codeword in the first subset defines a class type, we try to select as many codewords in the first subset as possible.

5. Go to the next subset and select the codewords which can together form a class type and yields the largest number of codewords after removing all codewords in the main set at distance $< d_{\min}$ to the selected codewords.

6. Remove all codewords at distance $< d_{\min}$ from the main set.

7. Subsequently go to the fifth step and continue until the complete set of codewords in all subsets has been purged.

The total number of available classes $N(n, d, d_{\min})$ can be found by adding up the number of codeword classes of each type. Table 4.23 lists the code size $N(n, d, d_{\min} = 3)$ for $d = 1, 2$ and 3. For comparison purposes, Fig. 4.3 compares the code rate efficiencies calculated by Construction 1 and Construction 2 for $d_{\min} = 3$. It may be noticed from Table 4.23 that there is an improvement in the code efficiencies obtained by Construction 2. When $d = 3$, a large number of codeword class types has to be considered for the code construction. As a result, Construction 2 yields a clear improvement for $d = 3$.

Table 4.23: Code size $N(n, d, d_{\min} = 3)$ based on Construction 2.

n	$N(n, d = 1, d_{\min} = 3)$	$N(n, d = 2, d_{\min} = 3)$	$N(n, d = 3, d_{\min} = 3)$
8	9	5	4
9	14	8	4
10	21	10	4
11	32	13	8
12	47	20	11
13	70	27	13
14	107	35	18
15	173	51	24
16	245	69	30
17	385	93	41
18	574	134	54
19	902	184	71
20	1335	248	87
21	-	359	119
22	-	496	158

Table 4.24 shows $N(n, d, d_{\min} = 4)$ as a function of codeword length n for $d = 2$ and 3. The results are obtained using Construction 2. The table reveals the feasibility of the construction of rates 5/16, 6/18 and 7/20 $d = 2$-constraint RLL block codes. For $d = 3$ and

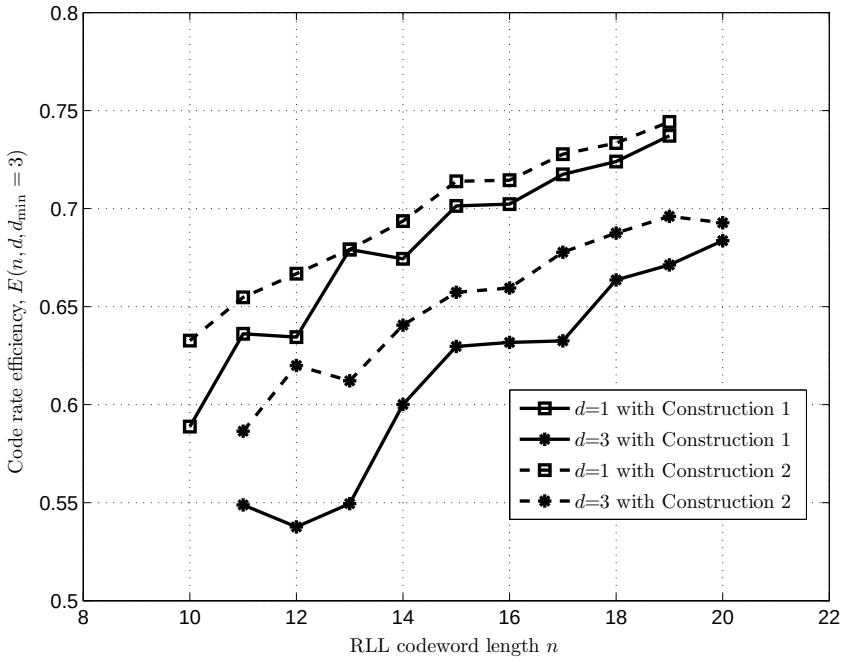

Figure 4.3: Code rate efficiencies for $d_{\min} = 3$.

$d_{\min} = 4$, we can construct codes with rates 5/18 and 6/20. Obviously, as $d_{\min}$ increases, the feasible rate of the code decreases. Since the code rates are less attractive, some values of d and $d_{\min}$, such as $d = 1$ and $d_{\min} = 4$, are not considered. Table 4.24 shows that a rate 6/24 ($d = 3$)-constraint RLL code can be constructed for $d_{\min} = 5$.

4.3 Conclusion

The look-ahead RLL code construction algorithm as introduced in [20] was discussed and generalized for $d = 2$ and $d = 3$. In addition, we presented many novel block-decodable RLL codes having a minimum Hamming distance larger than one. The constructed codes have several advantages in terms of the achievable code rate, the minimum distance and simple encoding and decoding.

Table 4.24: Code sizes $N(n, d, d_{\min} = 4)$ and $N(n, d, d_{\min} = 4)$ based on Construction 2.

n	$N(n, d = 2, d_{\min} = 4)$	$N(n, d = 3, d_{\min} = 4)$	$N(n, d = 3, d_{\min} = 5)$
12	14	10	4
13	17	10	6
14	23	12	6
15	30	14	9
16	47	24	13
17	54	26	15
18	81	35	17
19	115	38	22
20	157	68	29
21	198	75	40
22	287	104	48
23	—	120	60
24	—	190	77

Chapter 5

OFDM Modulation for Impulsive Noise Error Correction

OFDM is a form of multi-carrier modulation scheme used in many data transmission systems such as digital video broadcasting (DVB), digital audio broadcasting (DAB) and wireless local area networks (WLANs). It is also a strong candidate as a modulation scheme in high data rate (up to 1 Mbps) narrowband (9-500 kHz) PLC modems, see [45] and [46]. In narrowband PLC, the reliability of transmission is strongly influenced by the non-Gaussian impulsive noise. In OFDM, the IDFT is used for modulating a block of n information symbols on n subcarriers [47]. The time duration of an OFDM symbol is n times larger than that of a single-carrier system. This longer duration of OFDM symbol provide an advantage that the impulsive noise energy is spread among the n subcarriers due to the DFT operation. This spreading causes less interference over all n subcarriers [49]. However, when the impulsive noise energy exceeds a certain threshold, a significant performance loss can occur due to the higher level of interference at each subcarrier [48]. It was shown in [15] that impulsive noise leads to an enormous loss in the capacity, as well as in the error rate performance.

In this chapter, we study the similarity between the IDFT and the RS encoder. If the DFT of the transmitted OFDM symbol contains some amount of consecutive zeros appended to the information, the OFDM modulator can be seen as a RS encoder [34]. In practice, not all subcarriers are used to carry information. Some subcarriers are set to known data (pilot symbols) or zeros for different purposes, such as DC cancelation, synchronization or channel-estimation. It will be shown that this redundancy can be used to correct the impulsive noise errors.

The relationship between the DFT and RS codes was examined by Wolf [35] to correct the impulsive noise errors. He showed that when a frequency domain sequence containing two consecutive zeros is modulated by the IDFT, one impulsive noise error can be corrected by examining the frequency components that should be zero. An algebraic scheme based on BCH decoding is applied to estimate the position and the amplitude of the impulsive noise error. However, the scheme was only presented for a single impulsive error correction, and it was not generalized for multiple errors.

Kumaresan [54] investigated RS and BCH codes in real and complex fields to correct impulsive noise errors in the presence of additional minor errors. He proposed several strategies

using least squares techniques and singular value decomposition to decode the location and the corresponding amplitudes of the impulsive noise errors.

The similarity between the IDFT and RS encoder was further investigated by Redinbo in [55]. He presented an error correcting procedure for real number codes which also have consecutive zeros in the frequency domain. The procedure is divided into two parts. The first part is the large activity detection to locate the position of impulsive noise errors, and the second part is the minimum mean-squared error estimator coupled with activity detection.

The proposed procedures in [54] and [55] consider that the DFT of the transmitted sequence contains a large number of consecutive zeros. In practice, however, only a limited number of consecutive zeros is available. In [56], the impulsive noise error correction procedure based on the similarity between the DFT and RS codes is generalized to the case when the redundant data (zeros or known data) are scattered among the DFT of the transmitted sequence. The main drawback is that the correction capacity is conditioned by the position of the redundant data (2 impulsive noise errors can be corrected using five redundant data).

In this chapter, we propose to separate the error correcting procedure into two parts, threshold detector followed by an erasure decoding. The first part determines if large amplitudes are present in the received vector, declaring their locations as erasures, while the second part decodes the identified erasures. The threshold detector determines locations of the large amplitudes by applying Bayes hypothesis.

The paper is organized as follows. After a brief introduction to the OFDM encoding procedure, the similarity between the IDFT and RS encoder is explained in Section 5.3. The proposed decoder is discussed in Section 5.4. Finally, Section 5.5 concludes the chapter.

5.1 OFDM Modulator

We assume in general that the input vector $\mathbf{u} \in \mathbf{U}^m$ to the OFDM modulator consists of $m \leq n$ information symbols with a symbol energy E_s and a zero vector of length $n-m$ appended to the information, i.e., $\mathbf{u} = (u_1, ..., u_m, 0, ..., 0)^T$. The symbol alphabet $\mathbf{U}$ is a set of $|\mathbf{U}|$ discrete points in the complex plane depending on the modulation scheme adopted. As an example, for 4-ary quadrature amplitude modulation (4-QAM), $\mathbf{U} = \pm(1+j); \pm(1-j)$ is obtained with $j := \sqrt{-1}$. The transformed symbol vector $\mathbf{c} \in \mathbf{C}$ defined by the linear mapping

$$\mathbf{c} = \mathbf{V}^{-1}\mathbf{u}, \tag{5.1}$$

where

$$\mathbf{V}^{-1} = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{(n-1)} & \cdots & \alpha^{(n-1)(n-1)} \end{pmatrix}, \ \alpha = e^{\frac{j2\pi}{n}}. \tag{5.2}$$

The IDFT or DFT matrix is a Vandermonde unitary matrix [50], where α is a root of unity, i.e., $|\alpha^{k(l-1)}| = 1$. Unitary means that $\mathbf{V}\mathbf{V}^{-1} = \mathbf{I}$ holds, where $\mathbf{I}$ is the identity matrix, and

$\mathbf{V}$ is the DFT matrix. The transmitted OFDM samples c_k's are complex variables with

$$\mathrm{E}\{\mathrm{Re}\{c_k\}^2\} = \mathrm{E}\{\mathrm{Im}\{c_k\}^2\} = \frac{1}{2}\sigma_{c_k}^2 = \frac{m}{n}\frac{E_s}{2}; \quad k = 1, ..., n, \tag{5.3}$$

and Re{} and Im{} denote the real and imaginary part of the complex signal, respectively. We assume that every transformed OFDM symbol vector $\mathbf{c} \in \mathbf{C}$ is transmitted equally likely with the probability $P(\mathbf{c}) = 1/|\mathbf{C}|$, where $|\mathbf{C}|$ is the cardinality of $\mathbf{C}$. The vector $\mathbf{c}$ is transmitted over the two-state impulsive noise channel model, see Section 2.2.1 for the channel definition. The received vector $\mathbf{r} = (r_1, ..., r_n)^T$ can be expressed as $\mathbf{r} = \mathbf{c} + \mathbf{w}$, where $\mathbf{w} = (w_1, ..., w_n)^T$ is the channel noise vector, which consists of the complex noise samples.

5.2 The ML Decoding Rule

The ML decoder selects the most likely transmitted vector $\mathbf{c}$, which maximizes the conditional PDF $p(\mathbf{r}|\mathbf{c})$. If the channel state vector $\mathbf{s}$ is known at the receiver, the ML decoding rule can be simplified to

$$\max_{\mathbf{c}} p(\mathbf{r}|\mathbf{c}, \mathbf{s}) = \max_{\mathbf{c}} \sum_{k=1}^{n} \frac{|r_k - c_k|^2}{\sigma_{s_k}^2}, \tag{5.4}$$

see Eq. 2.10. The above equation can be solved by searching through all possible codewords $\mathbf{c} \in \mathbf{C}$. However, this results in a very high decoding complexity depending on the cardinality of $\mathbf{C}$. For complexity reasons, ML decoding rule for OFDM transmission is not considered in this thesis.

5.3 OFDM as an Error Correcting Code

If the positions of zeros (or known data) at the input vector $\mathbf{u}$ are consecutive in a circular way, there is a similarity between the RS encoder and IDFT. In the following, the similarity is further explained by deriving some properties based on the Vandermonde matrix structure.

Property 5.1. Any submatrix of $\mathbf{V}^{-1}$, which contains m consecutive columns (rows), has rank m.

Property 5.1 easily follows from the fact that any submatrix containing m consecutive columns also has the form of a Vandermonde matrix. Since all m columns and rows are independent, the resulting matrix has rank m.

Definition 5.1. The RS encoder performs the inner product of an information vector $\mathbf{u}$ of length m with a $n \times m$ matrix $\mathbf{V}_{\mathrm{sub}}^{-1}$. The resulting vector $\mathbf{c}$ is a code vector. We can represent $\mathbf{V}_{sub}^{-1}$ as the submatrix of $\mathbf{V}^{-1}$ containing its first m columns, where α is taken as a primitive element of the Galois field [58]. It can be followed from Property 5.1 that no linear combination of columns can give a code vector with m zeros. Hence, the minimum nonzero weight of any code vector is $n - m + 1$. Since RS codes are linear codes, the minimum weight of the code is identical to the minimum Hamming distance.

The IDFT matrix $\mathbf{V}^{-1}$ can be seen as a RS encoder by defining RS code over the complex number field with $\alpha = e^{\frac{j2\pi}{n}}$.

Property 5.2. There exists an $(n-m) \times n$ matrix of rank $(n-m)$, denoted by $\mathbf{H}^H$, the Hermitian transpose of the parity-check matrix $\mathbf{H}$, which satisfies

$$\mathbf{H}^H \mathbf{V}_{\text{sub}}^{-1} = \mathbf{0}. \tag{5.5}$$

For the IDFT matrix $\mathbf{V}^{-1}$, the matrix $\mathbf{H}^H$ can be written as

$$\mathbf{H}^H = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & \alpha^{(n-m)} & \cdots & \alpha^{(n-1)(n-m)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^2 & \cdots & \alpha^{2(n-1)} \\ 1 & \alpha & \cdots & \alpha^{(n-1)} \end{pmatrix}, \ \alpha = e^{\frac{j2\pi}{n}}. \tag{5.6}$$

In Eq. (5.6), we again have the structure of the Vandermonde matrix. Thus, $\mathbf{H}^H$ has rank $(n-m)$.

At the input of the OFDM modulator, the last $n-m$ positions are set to zero. Hence, the syndrome vector $\mathbf{S}_{\text{syn}}$ can be defined as

$$\mathbf{S}_{\text{syn}} = \mathbf{H}^H \mathbf{r}, \tag{5.7}$$

where $\mathbf{r} = (r_1, ..., r_n)^T$ is the received vector after transmission over the impulsive noise channel. Let us assume no background noise to simplify the explanation of the decoding. Then, the received vector can be written as $\mathbf{r} = \mathbf{c} + \mathbf{i}$, where $\mathbf{i} = (i_1, ..., i_n)^T$ is the impulsive noise vector. Using Property 5.2 and from Equations (5.7) and (5.1), the syndrome vector $\mathbf{S}_{\text{syn}}$ directly follows

$$\begin{aligned} \mathbf{S}_{\text{syn}} &= \mathbf{H}^H \mathbf{V}^{-1} \mathbf{u} + \mathbf{H}^H \mathbf{i}, & (5.8) \\ &= \mathbf{H}^H \mathbf{i}. & (5.9) \end{aligned}$$

In Eq. (5.8), the vector $\mathbf{S}_{\text{syn}}$ depends only on the impulsive noise components, not on the transmitted OFDM samples, see Fig. 5.1.

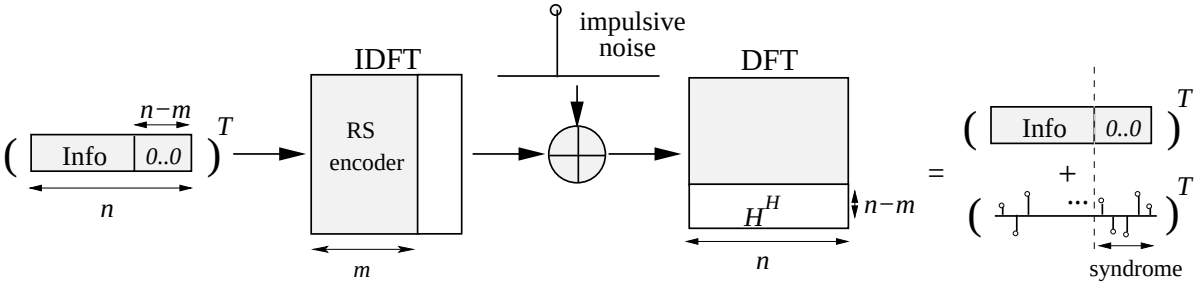

Figure 5.1: Illustration of the OFDM encoding and syndrome forming.

Property 5.3. Any $\leq (n-m)$ erasures in the vector $\mathbf{c}$ can always be corrected.

Suppose that we write the information vector $\mathbf{u}$ as a degree $m-1$ polynomial $u(x)$. Then, from the theory of Lagrange interpolation [57], we know that at least m points of evaluation are needed to specify a polynomial of degree $m-1$. This implies that in the case of $\leq (n-m)$ erasures, the information vector $\mathbf{u}$ can be reconstructed from the $\geq (n-m)$ values of $\mathbf{c}$ with the knowledge of their locations.

Property 5.4. Any $\leq (n-m)/2$ errors in the vector $\mathbf{c}$ can always be corrected.

This property follows from Definition 5.1 that a code with a minimum Hamming distance $n-m+1$ can correct $\leq (n-m)/2$ errors [58]. However, the decoding problem is the estimation of the positions and the values of the errors from the syndrome values. Therefore, for $\leq (n-m)/2$ errors, the algorithms for decoding RS codes can be adapted, see [35,54–56].

Example 5.1. Let us consider 4-OFDM transmission with two zeros appended to the information. After applying IDFT transformation, the impulsive noise vector $\mathbf{i}$ is added to the transformed vector $\mathbf{c}$, see Fig. 5.2. Suppose the vector $\mathbf{i}$ has only one non-zero component at the second position. Due to Property 5.4, we expect to correct $(4-2)/2 = 1$ error.

At the receiver side, given the received vector $\mathbf{r}$, we calculate the syndrome vector as

$$\mathbf{S}_{\text{syn}} = \begin{pmatrix} i_2\alpha^2/2 \\ i_2\alpha/2 \end{pmatrix}, \tag{5.10}$$

where $\alpha = e^{\frac{j2\pi}{n}}$. Using the algebraic scheme introduced by Wolf [35], the location of errors can be calculated by the equation

$$\text{location} = j\frac{n}{2\pi}\ln\frac{S_{\text{syn}}(2)}{S_{\text{syn}}(1)} + 1 = j\frac{4}{2\pi}\ln\frac{i_2\alpha/2}{i_2\alpha^2/2} + 1 = 2 \tag{5.11}$$

and the complex amplitude of error can be given as

$$\text{amplitude} = \sqrt{n}\frac{S_{\text{syn}}(2)^2}{S_{\text{syn}}(1)} = 2\frac{(i_2\alpha/2)^2}{(i_2\alpha^2/2)} = i_2. \tag{5.12}$$

The algebraic scheme presented by Wolf was not generalized for multiple errors. In that case, well-known decoding algorithms for RS codes can be adapted for the complex field [35, 54–56]. The algorithms are based on the polynomial representation of RS codes, and the general steps can be summarized as follows.

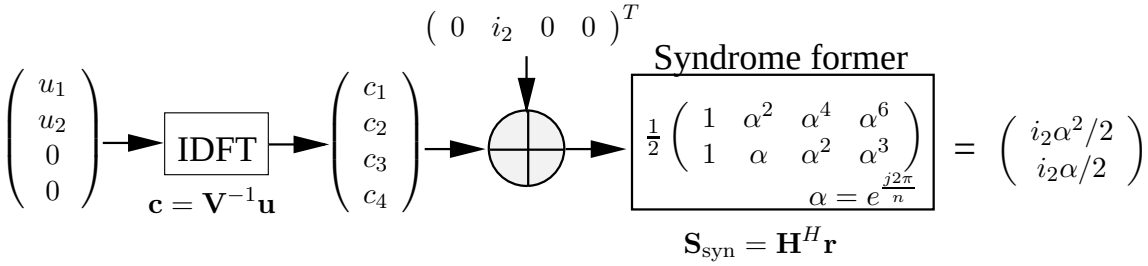

Figure 5.2: Example of the OFDM encoding and syndrome forming.

- Computation of the syndrome.
- Determination of an error locator polynomial, whose roots provide an indication of where the errors are.
- Finding the roots of the error locator polynomial.
- Calculating the error values.

In practice, after filtering operation, only a small part of consecutive known data or zeros can be used for the error correction. Furthermore, the transmitted vector $\mathbf{c}$ is also corrupted by the background noise. Hence, the performance of the error decoding procedure is quite limited for the impulsive noise channels.

5.4 OFDM as an Erasure Correcting Code

In this section, we propose to separate the error correcting procedure into two parts, threshold detector followed by an erasure decoding. The first part determines if large amplitudes are present in the received vector $\mathbf{r}$, declaring their locations as erasures, while the second part decodes the identified erasures. The threshold detector determines locations of the large amplitudes by applying Bayes hypothesis. To simplify the explanation of the proposed algorithm, let us first assume an impulsive noise channel with perfect channel state information (CSI), i.e., the location of impulse noise errors is known at the receiver.

5.4.1 Channel with Perfect CSI

After the transmission over two-state impulsive noise model, the received vector $\mathbf{r} = (r_1, ..., r_n)^T$ can be expressed as

$$\mathbf{r} = \mathbf{c} + \mathbf{i} + \mathbf{g}, \tag{5.13}$$

where $\mathbf{g}$ is the Gaussian background noise vector. The corresponding syndrome vector $\mathbf{S}_{\text{syn}}$ directly follows

$$\mathbf{S}_{\text{syn}} = \mathbf{H}^H\mathbf{i} + \mathbf{H}^H\mathbf{g}. \tag{5.14}$$

From $\mathbf{S}_{\text{syn}}$, the whole vector $\mathbf{i}$ has to be reconstructed and subtracted from the vector $\mathbf{r}$ to correct the impulsive noise errors. Suppose that the impulsive noise vector $\mathbf{i} = (i_1, ..., i_n)^T$, has $t \leq (m - n)$ non-zero samples (positions where the channel state $s_k > 0$). Then, the factor $\mathbf{H}^H\mathbf{i}$ can be rewritten as

$$\mathbf{H}^H\mathbf{i} = \mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}}, \tag{5.15}$$

where $\mathbf{i}_{sub}$ consists of t non-zero samples of $\mathbf{i}$, and the $(n-m) \times t$ matrix $\mathbf{H}_{\text{sub}}$ is constructed by selecting t columns of $\mathbf{H}^H$ that are at the position of impulsive noise errors, see Fig. 5.3. Using Eq. (5.15), Eq. (5.14) can be rewritten as

$$\mathbf{S}_{\text{syn}} = \mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}} + \mathbf{H}^H\mathbf{g}. \tag{5.16}$$

Since all entries of $\mathbf{H}^H$ are on the normalized unit circle with $|\alpha^{k(l-1)}| = 1$, the sum term $\mathbf{H}^H\mathbf{g}$ in Eq. (5.16) adds n background Gaussian random variables with variance σ_g^2/n. Hence, the sum term also yields a Gaussian random variable with variance σ_g^2. Therefore, Eq (5.16) can be simplified as

$$\mathbf{S}_{\text{syn}} = \mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}} + \mathbf{g}. \tag{5.17}$$

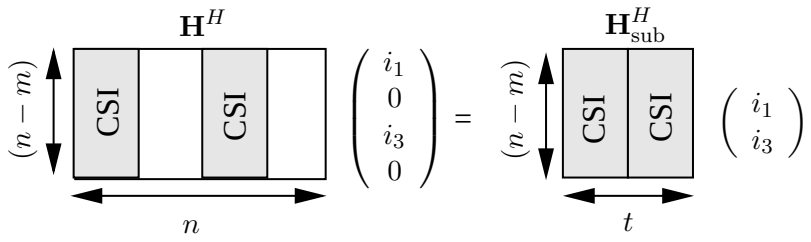

Figure 5.3: Illustration of the factor $\mathbf{H}^H_{\text{sub}}\mathbf{i}_{\text{sub}}$.

The decoding problem is now reduced to the estimation of the vector $\mathbf{i}_{\text{sub}}$ in the presence of the Gaussian background noise vector $\mathbf{g}$.

If $t = (n-m)$, i.e., the number of impulses is equal to the number of zeros, then $\mathbf{H}^H_{\text{sub}}$ is a square matrix of rank $(n-m)$, see Property 5.1. Hence, $\mathbf{H}_{\text{sub}}$ is invertible, and there is only one solution for the inverse $\mathbf{H}^{-1}_{\text{sub}}$. We can then obtain the estimate $\widetilde{\mathbf{i}}_{\text{sub}}$ of the vector $\mathbf{i}_{\text{sub}}$ as

$$\begin{aligned} \widetilde{\mathbf{i}}_{\text{sub}} &= \mathbf{H}^{-1}_{\text{sub}}\mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}} + \mathbf{H}^{-1}_{\text{sub}}\mathbf{g}, \\ &= \mathbf{i}_{\text{sub}} + \mathbf{H}^{-1}_{\text{sub}}\mathbf{g}. \end{aligned} \tag{5.18}$$

If $t < (n-m)$, there is not a unique solution for Eq. (5.17). In that case, a least-squares solution can be applied.

Least-Squares Estimation

The least-squares estimator f_{LSE} outputs the best fit between the syndrome values $\mathbf{S}_{\text{syn}}$ and the values $\mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}}$ in its least-squares sense, i.e.,

$$f_{\text{LSE}} = (\mathbf{S}_{\text{syn}} - \mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}})^H(\mathbf{S}_{\text{syn}} - \mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}}). \tag{5.19}$$

Expanding the multiplication in Eq. (5.19), we find for the least-squares

$$f_{\text{LSE}} = \mathbf{S}^H_{\text{syn}}\mathbf{S}_{\text{syn}} - \mathbf{i}^H_{\text{sub}}\mathbf{H}^H_{\text{sub}}\mathbf{S}_{\text{syn}} - \mathbf{S}^H_{\text{syn}}\mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}} + \mathbf{i}^H_{\text{sub}}\mathbf{H}^H_{\text{sub}}\mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}}. \tag{5.20}$$

Considering a scalar computation, we differentiate with respect to i_{sub}, and set the result equal to zero, obtaining

$$\begin{aligned} -S^T_{\text{syn}}H_{\text{sub}} + i^T H^T_{\text{sub}}H_{\text{sub}} &= 0, \\ i^T H^T_{\text{sub}}H_{\text{sub}} &= S^T_{\text{syn}}H_{\text{sub}}, \\ i &= (H^T_{\text{sub}}H_{\text{sub}})^{-1}H^T_{\text{sub}}S_{\text{syn}}. \end{aligned} \tag{5.21}$$

After writing the scalars in matrix form, a least square solution to Eq. (5.17) is given by

$$\begin{aligned} \widetilde{\mathbf{i}}_{\text{sub}} &= (\mathbf{H}^H_{\text{sub}}\mathbf{H}_{\text{sub}})^{-1}\mathbf{H}^H_{\text{sub}}\mathbf{S}_{\text{syn}}, \\ &= (\mathbf{H}^H_{\text{sub}}\mathbf{H}_{\text{sub}})^{-1}\mathbf{H}^H_{\text{sub}}\mathbf{H}_{\text{sub}}\mathbf{i}_{\text{sub}} + (\mathbf{H}^H_{\text{sub}}\mathbf{H}_{\text{sub}})^{-1}\mathbf{H}^H_{\text{sub}}\mathbf{g}, \\ &= \mathbf{i}_{\text{sub}} + (\mathbf{H}^H_{\text{sub}}\mathbf{H}_{\text{sub}})^{-1}\mathbf{H}^H_{\text{sub}}\mathbf{g}. \end{aligned} \tag{5.22}$$

The inverse $(\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}})^{-1}\mathbf{H}_{\text{sub}}^H$ is known as Moore-Penrose pseudoinverse of $\mathbf{H}_{\text{sub}}$ [59].

The estimated impulsive noise vector $\widetilde{\mathbf{i}}$ contains the values of the computed vector $\widetilde{\mathbf{i}}_{\text{sub}}$ at the positions of impulsive noise errors, otherwise it contains zero. The vector $\widetilde{\mathbf{i}}$ is then subtracted from $\mathbf{r}$. After that conventional OFDM receiver can be applied, that is, the resulting vector $\mathbf{r} - \widetilde{\mathbf{i}}$ is transformed into the frequency domain by means of DFT, and the ML detector maps each frequency samples on a signal domain in the QAM-constellation, see Fig. 5.4.

As it can be seen from Eq. (5.22), an estimation of the vector $\mathbf{i}_{\text{sub}}$ can be obtained by multiplying the Moore-Penrose pseudoinverse of $\mathbf{H}_{\text{sub}}$ with the syndrome vector $\mathbf{S}_{\text{syn}}^H$. However, the power of the least-squares estimation noise $(\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}})^{-1}\mathbf{H}_{\text{sub}}^H \mathbf{g}$ may be more than the power of the original background noise $\mathbf{g}$. The covariance of the estimation noise can be calculated as

$$\begin{aligned}
\Phi_{syn} &= \mathrm{E}\{(\widetilde{\mathbf{i}}_{\text{sub}} - \mathbf{i}_{\text{sub}})(\widetilde{\mathbf{i}}_{\text{sub}} - \mathbf{i}_{\text{sub}})^H\}, \\
&= \mathrm{E}\{(\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}})^{-1}\mathbf{H}_{\text{sub}}^H \mathbf{g}\mathbf{g}^H \mathbf{H}_{\text{sub}}(\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}})^{-1}\}, \\
&= \mathbf{H}_{\text{sub}}^{-1}(\mathbf{H}_{\text{sub}}^H)^{-1}\mathbf{H}_{\text{sub}}^H N_0 \mathbf{H}_{\text{sub}}\mathbf{H}_{\text{sub}}^{-1}(\mathbf{H}_{\text{sub}}^H)^{-1}, \\
&= N_0(\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}})^{-1}, \qquad (5.23)
\end{aligned}$$

where $\mathrm{E}\{\mathbf{g}\mathbf{g}^H\} = N_0$. Clearly, the estimation noise is correlated and enhanced with the amplification factor $(\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}})^{-1}$. Small eigenvalues of $\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}}$ will lead to large estimation error. Since $\mathbf{H}_{\text{sub}}$ has dimensions $(n-m) \times t$, an increase in $(n-m)$ leads to a better performance by reducing the amplification factor. However, it can be shown for $t = (n-m) \to \infty$ that, the noise amplification factor tends to infinity [59]. Furthermore, dependency on the location of impulsive noise errors can be also observed in the amplification factor.

Minimum Mean-Squared Error Estimation

As we discussed before, the least-squares estimation enhances the background noise. Another approach is to apply the mean-squared error estimation (MMSE), which minimizes the mean square error, i.e.,

$$f_{MMSE} = \mathbf{E}\{(\widetilde{\mathbf{i}}_{\text{sub}} - \mathbf{i}_{\text{sub}})^H (\widetilde{\mathbf{i}}_{\text{sub}} - \mathbf{i}_{\text{sub}})\}. \qquad (5.24)$$

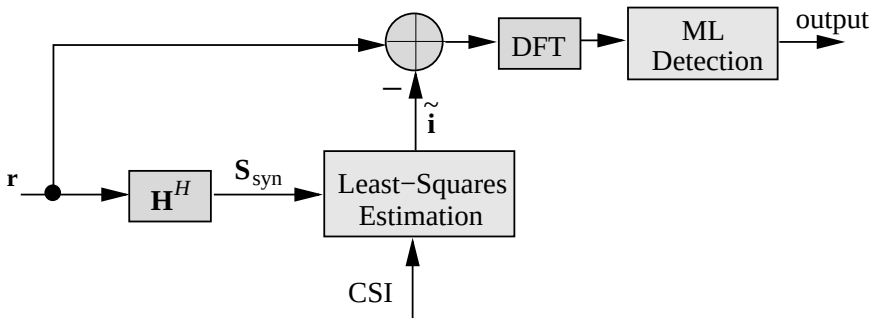

Figure 5.4: Block diagram of the receiver with perfect CSI

After solving the above equation, the MMSE estimate of $\mathbf{i}_{\text{sub}}$ can be written as

$$(\mathbf{H}_{\text{sub}}^H \mathbf{H}_{\text{sub}} + \frac{\sigma_g^2}{\sigma_i^2}\mathbf{I}_t)^{-1}\mathbf{H}_{\text{sub}}^H \mathbf{S}_{\text{syn}}, \tag{5.25}$$

where $\mathbf{I}_t$ is the $t \times t$ identity matrix. Unlike the least-squares estimator, the syndrome vector is multiplied by a matrix that is a function of σ_g^2/σ_i^2. However, since $\sigma_i^2 \gg \sigma_g^2$, the MMSE estimation converges to the least-squares estimation.

Nulling

If the number of impulses is larger than the number of zeros, i.e., $t > (n-m)$, the observation of $(n-m)$ zero frequency components in the syndrome vector $\mathbf{S}_{\text{syn}}$ is not sufficient to estimate the values of impulsive noise errors. Therefore, we propose to use the procedure below.

1. The $t-(n-m)$ components of $\mathbf{r}$ that are at the positions of impulsive noise errors, are nulled, that is, we set it to zero. Here, we select the $t-(n-m)$ components that have the smallest magnitudes.

2. The resulting vector $\widetilde{\mathbf{r}}$ is used to calculate the syndrome vector $\mathbf{S}_{\text{syn}} = H^H\widetilde{\mathbf{r}}$. Here, the additional noise introduced by setting $t-(n-m)$ components of $\mathbf{r}$ is spread over the frequency domain, and it is treated as background noise in the algorithm.

3. The $(n-m) \times (n-m)$ matrix $\mathbf{H}_{\text{sub}}$ is constructed by selecting the $(n-m)$ columns of $\mathbf{H}^H$ that are at the position of non-zero impulsive noise errors.

4. By calculating the Moore-Penrose pseudoinverse of $\mathbf{H}_{\text{sub}}$, a least squares solution is obtained, see Eq. (5.22).

5. The estimated impulsive noise vector $\widetilde{\mathbf{i}}$ contains the values of $\widetilde{\mathbf{i}}_{\text{sub}}$ at the positions of non-zero impulsive noise errors, otherwise it contains zero.

6. The vector $\widetilde{\mathbf{i}}$ is then subtracted from $\widetilde{\mathbf{r}}$. After that conventional OFDM receiver can be applied.

5.4.2 Channel without CSI

The block diagram of the receiver without CSI is depicted in Fig. 5.5. After the transmission over two-state impulsive noise model, the received vector $\mathbf{r}$ is processed by the threshold detector to estimate the channel state vector $\mathbf{s}$. For $k = 1...n$, the detector must decide every time instant k whether there is an impulsive noise or not, that is,

$$s_k = \begin{cases} 1 & \text{for } |r_k| \leq Th, \\ 0 & \text{for } |r_k| > Th, \end{cases} \tag{5.26}$$

where Th represents the threshold. For the calculation of the threshold Th, two hypotheses can be considered.

$$H_0 : r_k = c_k + g_k, \quad 1 \leq k \leq n, \tag{5.27}$$

$$H_1 : r_k = c_k + i_k + g_k, \quad 1 \leq k \leq n, \tag{5.28}$$

i.e., the received sample r_k may consist of background noise or it may consist of the sum of background noise and impulsive noise.

Approximation 5.1. For $n \to \infty$, due to the central limit theorem [7], the transmitted samples c_k's of the vector $\mathbf{c} = \mathbf{V}^{-1}\mathbf{u}$ are marginally complex Gaussian distributed random variables with zero mean, variance σ_c^2 and statistically independent real- and imaginary parts.

Using Approximation 5.1, the transmitted OFDM samples c_k's are approximated as a complex Gaussian distribution variables with zero mean, variance σ_c^2 and statistically independent real- and imaginary parts. Hence, the received sample r_k under hypothesis H_0 is approximated as a complex Gaussian distributed random variable with zero mean and variance $\sigma_{H_0}^2 = \sigma_c^2 + \sigma_g^2$, and the received sample r_k under hypothesis H_1 is also approximated as a complex Gaussian distributed random variable with zero mean and variance $\sigma_{H_1}^2 = \sigma_c^2 + \sigma_i^2 + \sigma_g^2$.

Assuming that the real- and imaginary parts of r_k's are independent, the PDF of $|r_k|$ follows a Rayleigh distribution [7]. Therefore,

$$p(|r_k| \mid H_0) = \frac{|r_k|}{\sigma_{H_0}^2} e^{-|r_k|^2/2\sigma_{H_0}^2} \tag{5.29}$$

and

$$p(|r_k| \mid H_1) = \frac{|r_k|}{\sigma_{H_1}^2} e^{-|r_k|^2/2\sigma_{H_1}^2}. \tag{5.30}$$

Applying the likelihood ratio test of the Bayes criterion [60], we decide for hypothesis H_1 if

$$\frac{p(|r_k| \mid H_1)}{p(|r_k| \mid H_0)} > \frac{P(H_0)}{P(H_1)}, \tag{5.31}$$

where $P(H_0)$ is the probability that the impulsive noise is not present, and $P(H_1)$ is the probability that the impulsive noise is present. Since we consider two-state impulsive channel model, the probabilities can be written as $P(H_0) = P(s_k = 0) = (1 - A)$ and $P(H_1) = P(s_k = 1) = A$, see Subsection 2.2.1.

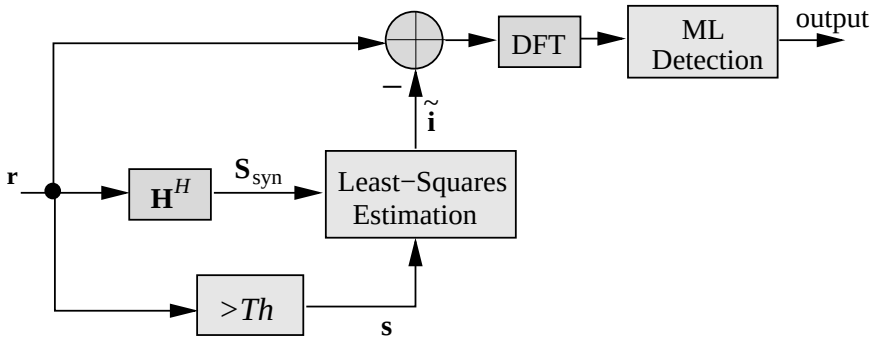

Figure 5.5: Block diagram of the receiver without CSI

After canceling common terms and taking logarithm, we decide for hypothesis H_1 if

$$\ln \frac{\sigma_{H_0}^2}{\sigma_{H_1}^2} + |r_k|^2 \frac{(\sigma_{H_1}^2 - \sigma_{H_0}^2)}{2\sigma_{H_0}^2 \sigma_{H_1}^2} > \ln \frac{(1-A)}{A}. \tag{5.32}$$

The quantity on the left is monotonically increasing. Hence, an optimal threshold can be calculated by setting

$$\ln \frac{\sigma_{H_0}^2}{\sigma_{H_1}^2} + |Th|^2 \frac{(\sigma_{H_1}^2 - \sigma_{H_0}^2)}{2\sigma_{H_0}^2 \sigma_{H_1}^2} = \ln \frac{(1-A)}{A}, \tag{5.33}$$

so that the threshold Th is obtained as

$$Th = \sqrt{\frac{2\sigma_{H_0}^2 \sigma_{H_1}^2}{(\sigma_{H_1}^2 - \sigma_{H_0}^2)} \left(\ln \left(\frac{1-A}{A} \right) - \ln \left(\frac{\sigma_{H_0}^2}{\sigma_{H_1}^2} \right) \right)}. \tag{5.34}$$

Figures 5.6, 5.7 and 5.8 show the QAM symbol error probability (SER) as a function of $E_s/2\sigma_g^2$ for the 64-OFDM transmission over the two-state impulsive noise channel model. The SER for the uncoded (no least-squares estimation) 64-OFDM transmission over the AWGN channel is also plotted for comparison.

Fig. 5.6 shows that, for $A = 0.1$, a small gain can be achieved in the error rate performance. This is due to the fact that since 8 zeros are available for the least-squares solution, error probability for the high SBNR is mainly influenced by the probability of having more than 8 impulses in an OFDM symbol. This probability can be halved by assuming that half of the QAM symbols are in error, i.e.,

$$\frac{1}{2} P(\text{number of impulses per OFDM symbol} > 8) = \frac{1}{2}(1 - \sum_{i=0}^{8} \binom{64}{i} A^i (1-A)^{64-i}) = 0.0933, \tag{5.35}$$

where $\binom{64}{i}$ denotes the binomial coefficient, and it is evaluated as $\frac{(64)!}{i!(64-i)!}$. It should also be noted that if the number of impulses is larger than the number of zeros, we are still able to correct some errors due to the nulling, see Subsection 5.4.1. For $A = 0.05$ and $A = 0.01$, large gains in error rate performance can be observed. The halved probability of having more than 8 impulses in an OFDM symbol can be calculated as $\approx 2 \cdot 10^{-3}$ for $A = 0.05$ and $\approx 8 \cdot 10^{-9}$ for A=0.01.

In Fig. 5.7, we vary the ratio $\Delta = \sigma_c^2/\sigma_i^2$ to evaluate the performance of the threshold detector. It can be seen that, for $\Delta = 1$, the impulsive noise can hardly be distinguished, i.e., the estimator losses information about the channel state. The SER for the least-squares estimator with perfect CSI is also added for comparison.

Fig. 5.8 illustrates the role of the number of zeros $(n - m)$ in the SER performance of the decoder. Increasing $(n - m)$ leads to a large gain, whereas, for small $(n - m)$, the performance improvement is quite limited.

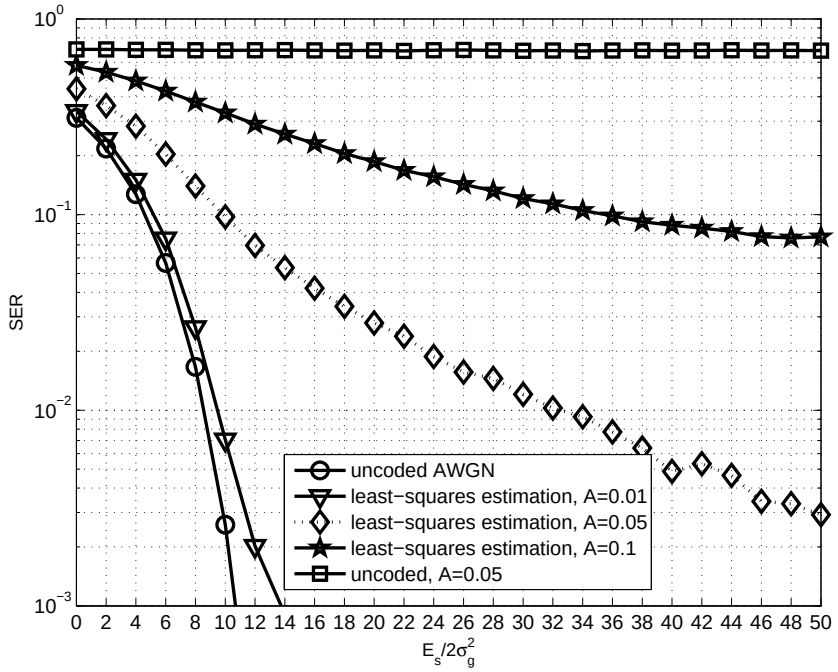

Figure 5.6: SER of the iterative 64-OFDM transmission using 4-QAM over the two-state impulsive noise channel with 8 zeros and $\Delta = 10^{-2}$.

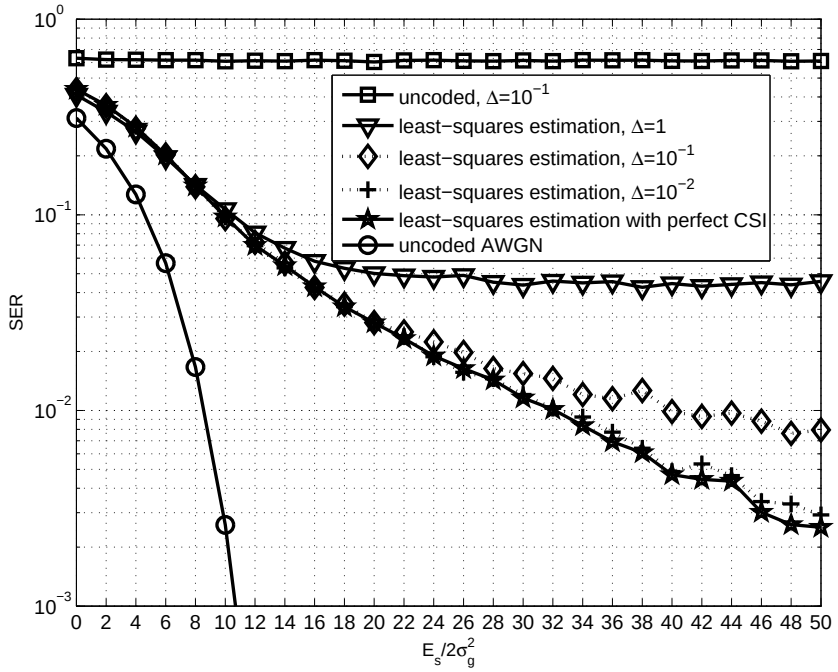

Figure 5.7: SER of the iterative 64-OFDM transmission using 4-QAM over the impulsive noise channel with 8 zeros and A=0.05.

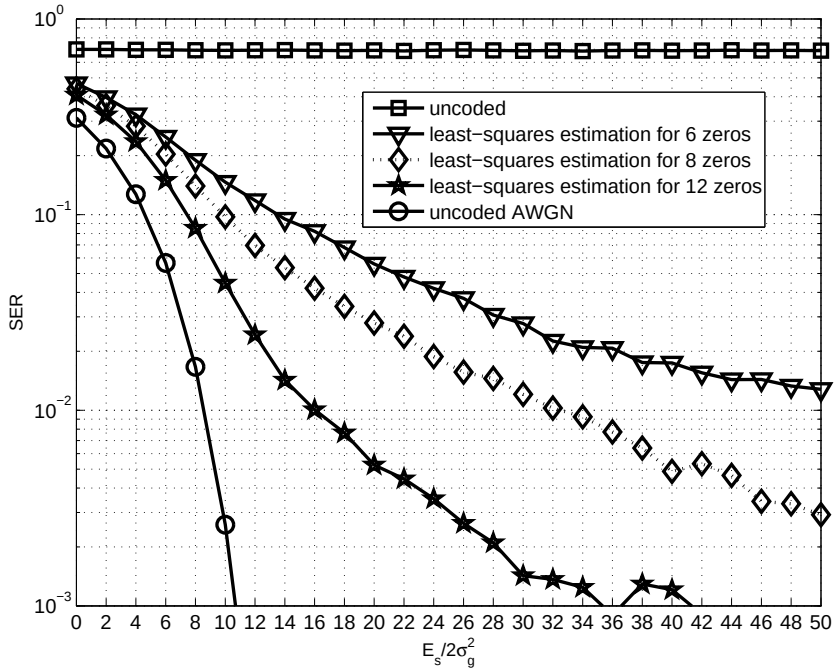

Figure 5.8: SER of the iterative 64-OFDM transmission using 4-QAM over the impulsive noise channel with A=0.05 and $\Delta = 10^{-2}$.

5.5 Discussion

Based on the above results, it can be observed that the SER performance of the decoder is quite poor for the small SBNR. This is mainly due to the amplified background noise in the least-squares estimator. Additionally, the parameter A has a strong influence on the decoder's performance. For $A = 0.1$, a small gain can be observed. Hence, from a practical point of view, applying simple pre-processing techniques, such as clipping and nulling, may lead to a better SER performance with the less complexity in the receiver design, see Fig. 6.7.

Chapter 6

Iterative Impulsive Noise Suppression in OFDM

Simple iterative impulsive noise suppression algorithms can improve the performance by exploiting the noise structure in the time and frequency domain, see [15] and [51]. In these algorithms, the number of subcarriers, n, plays an important role. For large $n > 256$, the convergence speed is fast, whereas, for $n \leq 256$, the algorithms converge slowly or even not at all. This is mainly due to the Gaussian approximation used in the algorithms for the distribution of the transformed random variables after DFT or IDFT. As n increases, the approximation becomes more accurate.

The analysis in the previous chapter showed that the error rate performance of the least-squares estimator receiver is limited. Therefore, the iterative algorithm as given in [15] is investigated in this chapter. We first show that using a clipping and nulling technique at the input of the algorithm significantly improves the error rate performance, and it enables a fast convergence for $n > 64$. Furthermore, we extend the iterative algorithm with a novel low complexity syndrome decoder to improve the error rate performance for $n \leq 64$. The decoder applies a successive impulsive noise error decoding instead of decoding all errors jointly. This approach has the advantage that the non-consecutive zeros or pilot tones can also be used in the decoding. Simulation results are provided to show the improvement in error rate.

This paper is organized as follows. In section 6.1, the impulsive noise spreading in OFDM is explained. It is followed by the iterative impulsive noise suppression algorithm, which is discussed in Section 6.2. The clipping and nulling technique is introduced in Section 6.3. Section 6.4 presents the extension of the iterative algorithm by a simple syndrome decoder. Finally, a conclusion is given in Section 6.5.

6.1 OFDM and Impulsive Noise Spreading

We assume in general that the input vector $\mathbf{u} \in \mathbf{U}^m$ to the OFDM modulator consists of $m \leq n$ information symbols with a symbol energy E_s and a zero vector of length $n - m$ appended to the information, i.e., $\mathbf{u} = (u_1, ..., u_m, 0, ..., 0)^T$. The symbol alphabet $\mathbf{U}$ is a set of $|\mathbf{U}|$ discrete points in the complex plane depending on the modulation scheme adopted.

The transformed symbol vector $\mathbf{c} \in \mathbf{C}^n$ defined by the linear mapping

$$\mathbf{c} = \mathbf{V}^{-1}\mathbf{u}, \tag{6.1}$$

where

$$\mathbf{V}^{-1} = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{(n-1)} & \cdots & \alpha^{(n-1)(n-1)} \end{pmatrix}, \ \alpha = e^{\frac{j2\pi}{n}}. \tag{6.2}$$

The OFDM symbol vector $\mathbf{c}$ is transmitted over the two-state impulsive noise channel model, see Section 2.2.1 for the channel definition. The received signal r_k of the received vector $\mathbf{r} = (r_1, ..., r_n)^T$ can be expressed as $r_k = c_k + w_k$, where w_k is the complex channel noise sample at time instant k that consists of the impulsive noise and the background noise samples. At the receiver, the received vector $\mathbf{r}$ is multiplied with the DFT matrix $\mathbf{V}$ yielding $\mathbf{R} = (R_1, ..., R_n) = \mathbf{V}\mathbf{r}$. Hence, the R_k's are given by

$$R_k = \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \alpha^{k(l-1)} r_l = u_k + \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \alpha^{k(l-1)} w_l, \tag{6.3}$$

where $u_k = 0$ for $k > m$.

Approximation 6.1. For $n \to \infty$, due to the central limit theorem [7], the sum term $\frac{1}{\sqrt{n}} \sum_{l=1}^{n} \alpha^{k(l-1)} w_l$ is a Gaussian distributed random variable with zero mean and variance σ_w^2, where σ_w^2 is the average noise variance of the channel noise, see Eq. (2.12).

This indicates that, for large n, the noise after DFT can be approximated to be AWGN. Moreover, the impulsive noise which is added to the transmitted signal is multiplied with the DFT matrix only; therefore, each impulse is spread in time so that its effect in the frequency domain may be considerable reduced. In general, the power of the impulsive noise is reduced by a factor of n. For $n = 1024$, the power is reduced by more than 30 dB, compared to the original signal power, see Fig. 6.1.

Since the noise components after DFT are approximated as Gaussian random variables, the conventional type of OFDM demodulator designed for the AWGN channel can be applied. This means that a ML detector makes an estimation of the information symbol u_k in each time instant $k = 1, ..., m$ based on the component R_k. However, the drawback of this detection is the reduced reliability of information, since the energy concentrated in one impulse is distributed over all information symbols. Hence, independently applying a detector to each component of $\mathbf{R}$ neglects the statistical dependency between the noise components after DFT, and results in a huge performance degradation. It was also shown in [16] that spreading can lead to an enormous loss in the capacity.

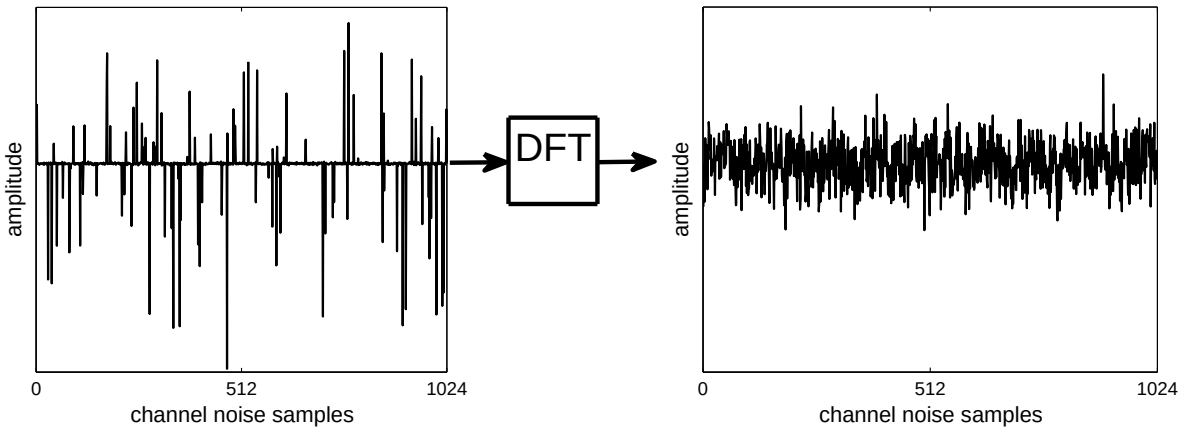

Figure 6.1: Impulsive noise spreading after DFT.

6.2 Iterative Noise Suppression Algorithm

An iterative impulsive noise suppression algorithm was proposed in [15] and [51] to increase the performance of the OFDM transmission scheme by exploiting the dependency of the noise in the frequency domain (after DFT). In this section, we review the algorithm as given in [15]. The block diagram of the receiver is depicted in Fig. 6.2. The algorithm performs the following steps.

1. After transmission over the impulsive noise channel, the received vector $\mathbf{r}$ is multiplied by the DFT matrix $\mathbf{V}$, yielding the transformed received vector $\mathbf{R}^{(0)} = \mathbf{V}\mathbf{r}$. The vector $\mathbf{R}^{(0)}$ represents the input of the ML detector at iteration zero.

2. The ML detector maps the first m elements of $\mathbf{R}^{(0)}$ on a signal point in the QAM constellation, and the last $n-m$ elements are set to zero. The estimated input vector $\mathbf{u}^{(l)}$, $l \geq 0$, represents the result of the lth iteration in the iterative algorithm.

3. $\mathbf{u}^{(l)}$ is then converted back into the time domain $\mathbf{c}^{(l)}$ and subtracted from the received vector $\mathbf{r}$. The result $\mathbf{w}^{(l)} = \mathbf{r} - \mathbf{c}^{(l)}$ denotes the estimated channel noise at lth iteration.

4. $\mathbf{w}^{(l)}$ is processed by the threshold detector to estimate the complex amplitudes at the positions of the impulsive noise, i.e.,

$$\widetilde{w}_k^{(l)} = \begin{cases} 0 & \text{for } |w_k^{(l)}| \leq Thr, \\ w_k^{(l)} & \text{for } |w_k^{(l)}| > Thr, \end{cases} \tag{6.4}$$

   where Thr represents the threshold defined in [15], and $k = 1, ..., n$.

5. The vector $\widetilde{\mathbf{w}}^{(l)}$ is subtracted from $\mathbf{r}$. The result is then converted back into the time domain, yielding $\mathbf{R}^{(l)} = \mathbf{V}(r - \widetilde{\mathbf{w}}^{(l)})$. The iterative process continues from Step 2 using $\mathbf{R}^{(l)}$.

If $\mathbf{u}^{(l+1)} = \mathbf{u}^{(l)}$ for the first time, the iteration stops.

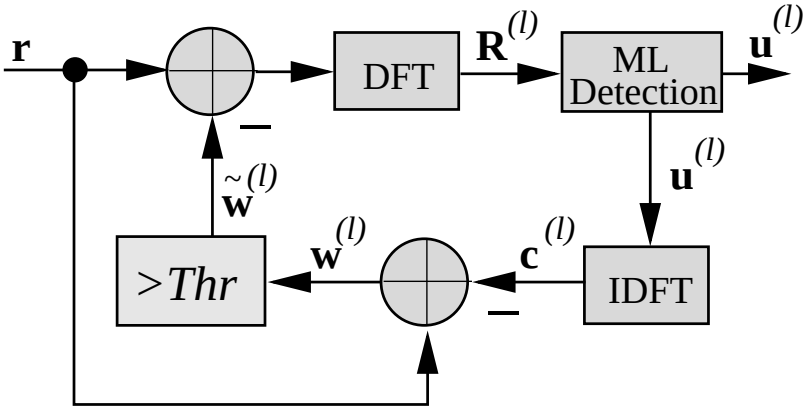

Figure 6.2: Block diagram of the Haering's iterative receiver.

The threshold Thr is calculated under the assumption that the vector $\mathbf{u} - \mathbf{u}^{(l)}$ (the contribution of the wrong decisions made by the ML detection) is Gaussian distributed after the IDFT operation. Thus, the threshold detector given in Eq. (6.4) decides whether the amplitudes $|w_k^{(l)}|$ belong to the impulsive peaks or to the underlying Gaussian noise. In Fig. 6.3, the calculated threshold is shown versus SNR, where the SNR is calculated as $\sigma_c^2/2\sigma_w^2$, see Eq. (2.17).

Fig 6.4 shows the SER as a function of the SBNR, $E_s/2\sigma_g^2$, for the 1024-OFDM transmission using 4-QAM over the two-state impulsive noise channel. For simplicity, we assume that $n = m$, i.e., no zeros are appended to the information. The SER for the uncoded transmission over the AWGN channel is also plotted for comparison. We observe that the algorithm converges within five iterations.

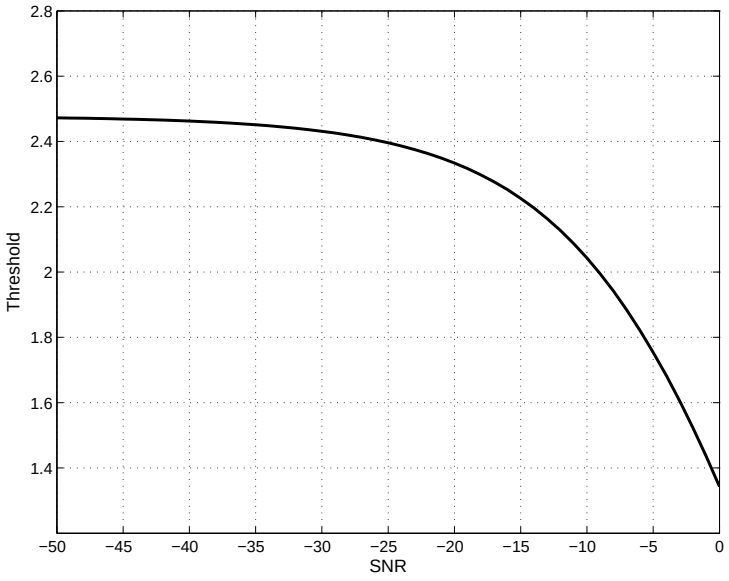

Figure 6.3: Calculated threshold values versus SNR.

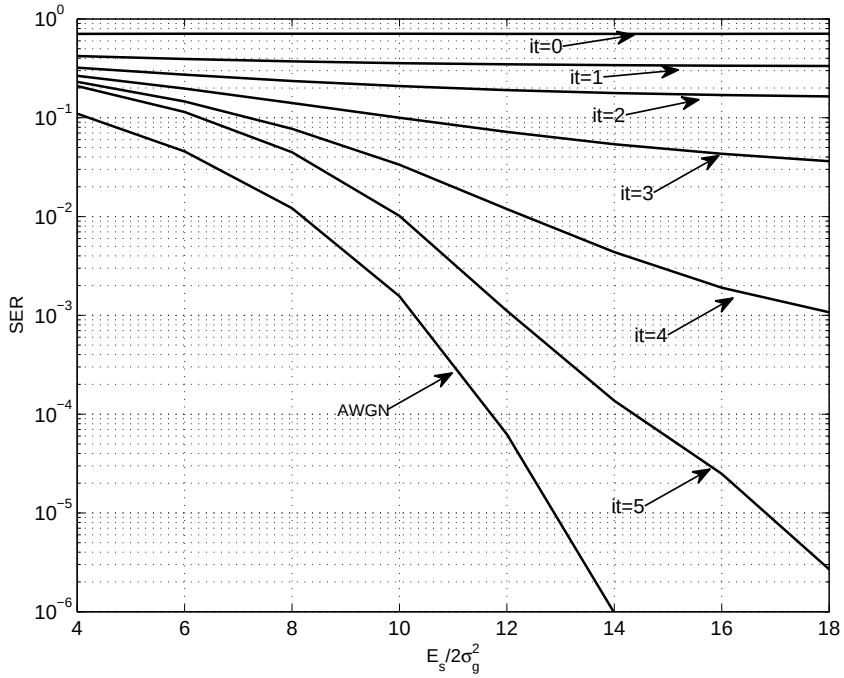

Figure 6.4: SER of the iterative 1024-OFDM transmission using 4-QAM over the two-state impulsive noise channel with $A = 0.1$ and $\Delta = \sigma_c^2/\sigma_i^2 = 10^{-2}$.

In the previous section, it was approximated for large n that the multiplication with the DFT matrix converts the impulsive noise channel into an AWGN channel, see Approximation 6.1. However, for small n, the performance should be evaluated in a different way. Let us consider the sum term $\frac{1}{\sqrt{n}} \sum_{l=1}^{n} \alpha^{k(l-1)} w_l$ in Eq. 6.3. Since all entries in $\mathbf{V}$ are on the normalized unit circle with a radius $|\alpha^{k(l-1)}| = 1$, the sum term adds n class A random variables with variance σ_w^2/n and parameters A and Δ. It was proved in [16] that the addition of the two class A random variables with parameters A, Δ and variance σ^2 is also a class A variable with parameters $2A$, Δ and variance $2\sigma^2$. Hence, the sum term $\frac{1}{\sqrt{n}} \sum_{l=1}^{n} \alpha^{k(l-1)} w_l$ yields a class A random variable with parameters $A_{sum} = nA$, $\Delta_{sum} = \Delta$ and the total variance $n\sigma_w^2/n = \sigma_w^2$.

Since, for small n, the noise components after DFT has a Class A distribution, the error probability achieved by the OFDM transmission over the AWCN channel is the same as the error probability achieved by the single carrier transmission over the AWCN channel with parameters A_{sum}, Δ_{sum} and variance σ_w^2. In general, for $n \leq 256$, the impulse energy after DFT is not spread evenly over all the information symbols, and consequently, the reliability of the first noise estimation $\mathbf{w}^{(0)}$ is low. In this case, the algorithm's convergence behavior is also poor, see Fig. 6.8. To maximize the SNR, impulsive noise mitigation techniques can be applied before the DFT.

6.3 Iterative Noise Suppression Algorithm with Pre-processing Techniques

The block diagram of the modified iterative receiver is depicted in Fig. 6.5. We consider two simple pre-processing techniques at the input of the iterative algorithm to increase the reliability of the first noise estimate $\mathbf{w}^{(0)}$ by reducing the impulsive noise energy.

a) Clipping [52]: Due to its simplicity, clipping technique is often used in practical applications. It can be applied to the received vector $\mathbf{r}$ to reduce the power of the impulsive noise by limiting the maximum signal value. For the received signal r_k, the decision regions are given as

$$\widetilde{r}_k = \begin{cases} r_k & \text{for } |r_k| \le T_{clip} \\ T_{clip} e^{j \arg r_k} & \text{for } |r_k| > T_{clip}, \end{cases}$$

where $T_{clip} = 2.2\sigma_c^2$ is the clipping threshold, see [52] and Fig. 6.6.

b) Clipping and nulling [53]: Since the mean value of the transmitted signal is zero, additional to the clipping, the amplitudes at the positions that are most likely hit by an impulse are set to zero. For the received signal r_k, the decision regions are given as

$$\widetilde{r}_k = \begin{cases} r_k & \text{for } |r_k| \le T_{clip} \\ T_{clip} e^{j \arg r_k} & \text{for } T_{clip} < |r_k| \le 1.4 T_{clip} \\ 0 & \text{for } |r_k| > 1.4 T_{clip} \end{cases} \tag{6.5}$$

where $T_{clip} = 2.2\sigma_c^2$ is the clipping threshold and $1.4T_{clip} \approx 3\sigma_c^2$ is the nulling threshold. It should be noted that the probability that the OFDM samples c_k's are above the nulling threshold $3\sigma_c^2$ can be calculated as $2 \cdot 10^{-3}$.

The pre-processed vector $\widetilde{\mathbf{r}} = (\widetilde{r}_1, ..., \widetilde{r}_n)^T$ is only used for the first estimation of the input vector $\mathbf{c}^{(0)}$, see Section 6.2. The input for the lth iteration is given by $\mathbf{r} - \widetilde{\mathbf{w}}^{(l)}$.

Fig. 6.6 shows the SER as a function of the clipping threshold for the 64-OFDM transmission using 4-QAM over the two-state impulsive noise channel. For all curves, clipping and

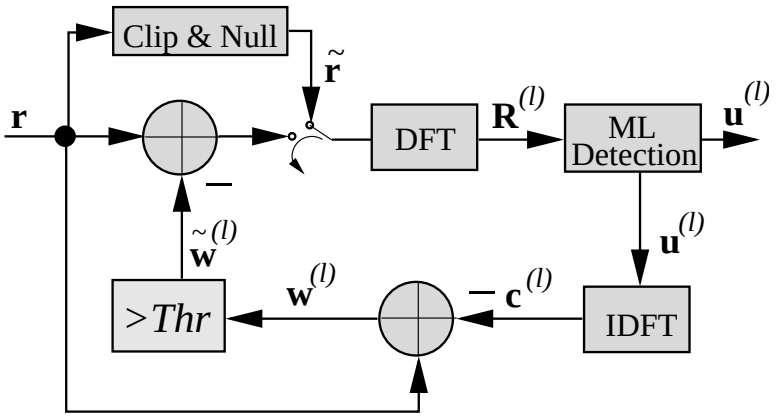

Figure 6.5: Block diagram of the iterative receiver with clipping and nulling.

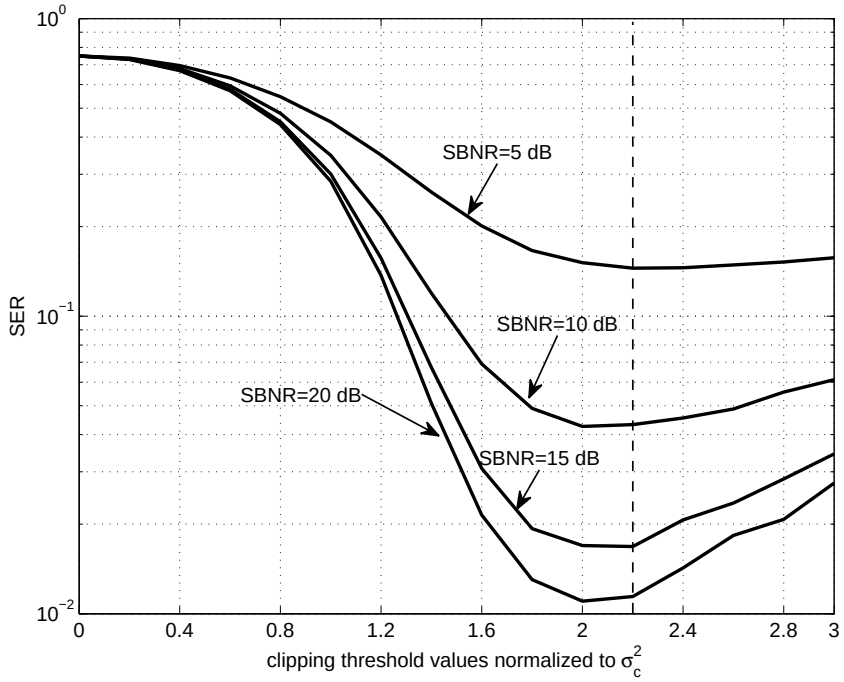

Figure 6.6: SER as a function of clipping threshold T_{clip} for the 64-OFDM transmission over the two-state impulsive noise channel with $A = 0.1$ and $\Delta = \sigma_c^2/\sigma_i^2 = 10^{-1}$.

nulling technique is applied followed by the conventional OFDM receiver, i.e., $l = 0$. It can be observed that the SER reaches the minimum value around $2.2\sigma_c^2$.

Fig. 6.7 shows the SER as a function of $E_s/2\sigma_g^2$ for the 64-OFDM transmission using 4-QAM over the impulsive noise channel model. The SER for the uncoded transmission over the AWGN channel is also plotted for comparison. For all curves, we consider no iteration, i.e., $l = 0$. We observe that using a clipping and nulling technique leads to a large improvement in the performance.

Fig. 6.8 shows the SER as a function of $E_s/2\sigma_g^2$ for the 128-OFDM transmission using the iterative receiver. The performance results illustrate that SER decreases significantly using a low complexity clipping and nulling technique at the input of the iterative algorithm. The algorithm converges within two iterations, whereas no convergence is observed in the iterative algorithm described in Section 6.2.

Fig. 6.9 shows the SER as a function of $E_s/2\sigma_g^2$ for the 256-OFDM transmission using the iterative receiver. It can be observed that after only two iterations, the performance of the uncoded transmission over the AWGN channel is almost achieved.

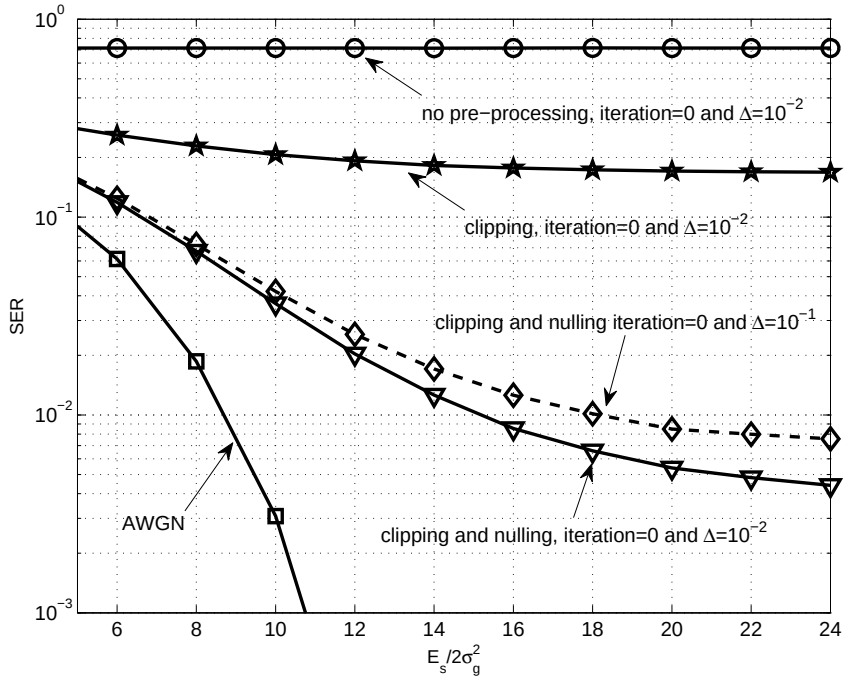

Figure 6.7: SER of the 64-OFDM transmission over the impulsive noise channel with A=0.1 and $\Delta = \sigma_c^2/\sigma_i^2$.

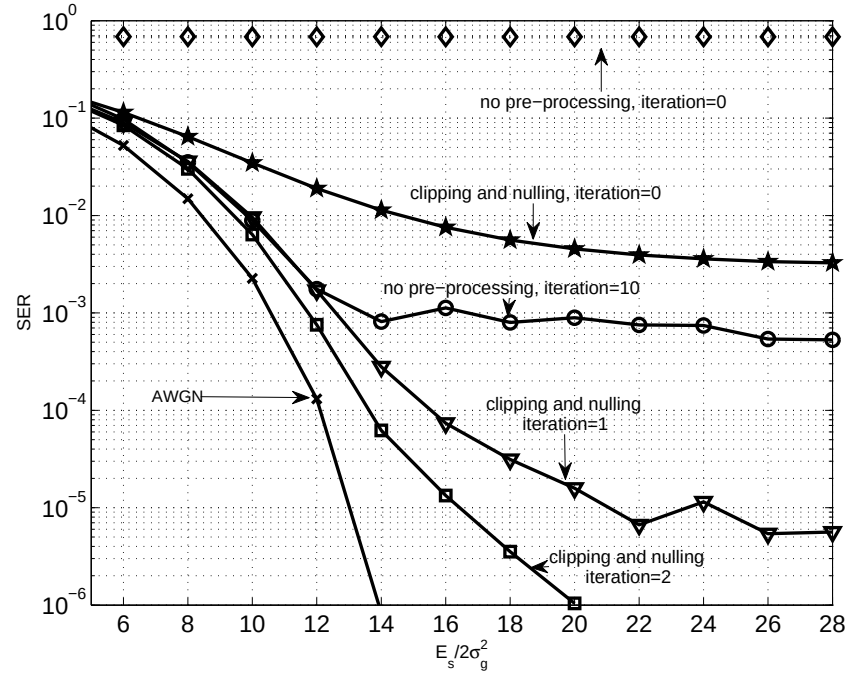

Figure 6.8: SER of the iterative 128-OFDM transmission using 4-QAM over the impulsive noise channel with A=0.1 and $\Delta = 10^{-2}$.

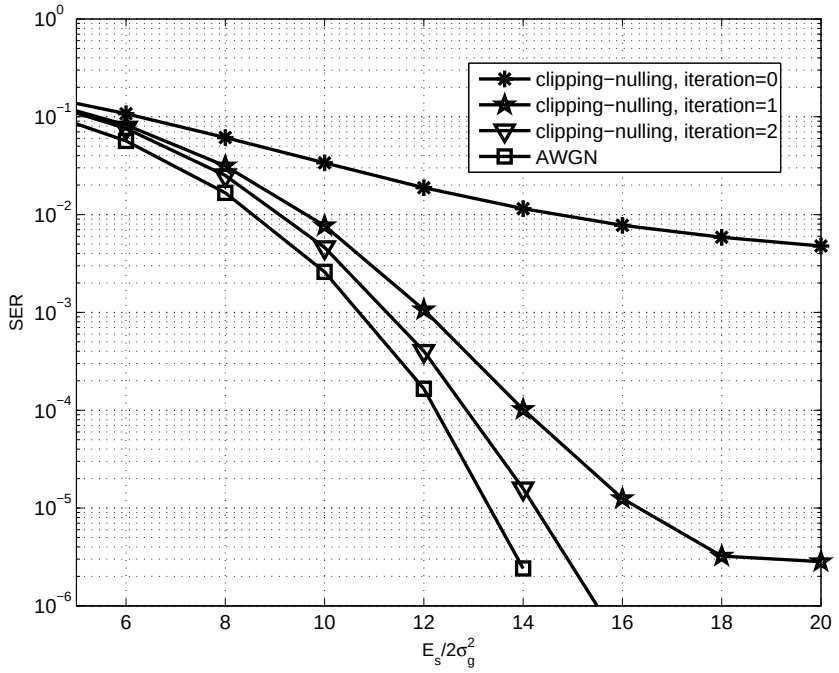

Figure 6.9: SER of the iterative 256-OFDM transmission using 4-QAM over the impulsive noise channel with A=0.1 and $\Delta = 10^{-2}$.

6.4 Iterative Receiver with 'Clipping and Nulling' and Syndrome Decoding

In this section, a syndrome decoder is included in the iterative receiver, located at the position as given in Fig. 6.10. At each iteration $l \geq 0$, $\mathbf{w}^{(l)}$ is processed by a modified threshold detector. The complex amplitudes after the threshold are defined as

$$\widetilde{w}_k^{(l)} = \begin{cases} w_k^{(l)} & \text{for} \quad |w_k^{(l)}| > Thr \quad \text{or} \quad |r_k| > T_{clip}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.6}$$

One approach to a lower complexity design of the syndrome decoder is to use a successive decoding strategy instead of decoding all noise components jointly. The syndrome decoder decodes the channel noise components one by one in descending order of magnitude of the obtained vector, $\widetilde{\mathbf{w}}^{(l)}$, see Fig. 6.11. First, the algorithm decodes the strongest noise component in $\widetilde{\mathbf{w}}^{(l)}$. Then, after removing the decoded noise component, the algorithm detects the next strongest noise component, and it continues until all components in $\widetilde{\mathbf{w}}^{(l)}$ are decoded. Decoding of each channel noise component essentially involves four steps.

1) *Masking:* The component of $\widetilde{\mathbf{w}}^{(l)}$ with the largest magnitude is masked, that is, we set it to zero. The resulting vector, $\widetilde{\mathbf{w}}_{unmasked}$, is then subtracted from $\mathbf{r}$. In this step, we want to cancel the channel noise in $\mathbf{r}$, except for the noise at the position of the masked component.

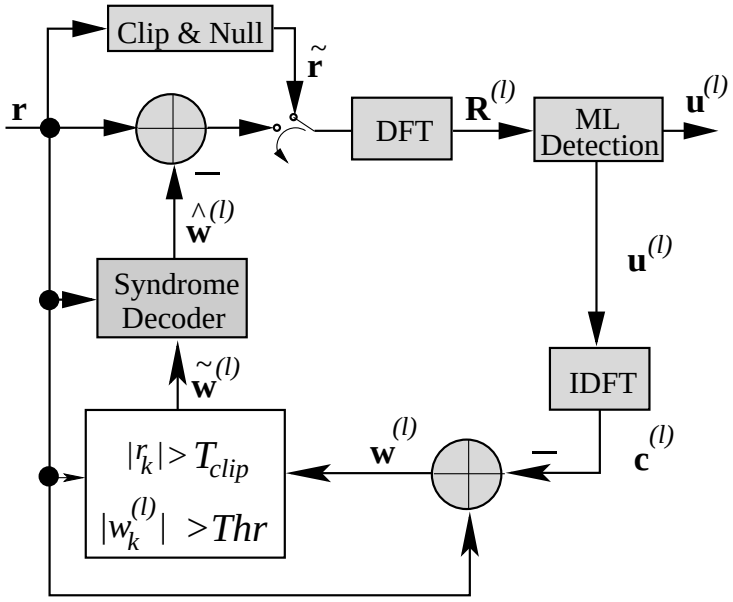

Figure 6.10: Block diagram of the proposed iterative receiver

2) *Syndrome former:* The last $n - m$ consecutive positions at the input of the OFDM modulator are set to zero. Hence, the syndrome vector $\mathbf{S}_{syn}$ can be calculated by an $(n-m)\times n$ matrix $\mathbf{H}^H$, the Hermitian transpose of the parity-check matrix $\mathbf{H}$, where $\mathbf{H}^H\mathbf{V}^{-1}\mathbf{u} = \mathbf{0}$. The vector $\mathbf{S}_{syn}$ depends only on the noise components, not on the transmitted OFDM symbol. Thus,

$$\mathbf{S}_{syn} = \mathbf{H}^H(\mathbf{r} - \widetilde{\mathbf{w}}_{unmasked}) = \mathbf{H}^H(\mathbf{w} - \widetilde{\mathbf{w}}_{unmasked}). \tag{6.7}$$

3) *Decoding of the noise at the masked position:* We want to estimate the channel noise component, w_{masked}, at the position of the masked component of $\widetilde{\mathbf{w}}^{(l)}$. Given that position, the syndrome vector $\mathbf{S}_{syn}$ can be rewritten as

$$\mathbf{S}_{syn} = \mathbf{H}_{sub} w_{masked} + \mathbf{H}^H\mathbf{e}, \tag{6.8}$$

where $\mathbf{H}_{sub}$ is the $(n-m)\times 1$ sub-matrix of $\mathbf{H}^H$ containing its column at the position of the masked component. The contribution $\mathbf{H}^H\mathbf{e}$ follows from $(\mathbf{w} - \widetilde{\mathbf{w}}_{unmasked})$ at the positions of the unmasked components. By applying the Moore-Penrose pseudo-inverse of $\mathbf{H}_{sub}$, a least square solution to Eq. (6.8) is given by

$$\begin{aligned} \widehat{w}_{masked} &= (\mathbf{H}_{sub}^H\mathbf{H}_{sub})^{-1}\mathbf{H}_{sub}^H\mathbf{S}_{syn}, \\ &= w_{masked} + (\mathbf{H}_{sub}^H\mathbf{H}_{sub})^{-1}\mathbf{H}_{sub}^H\mathbf{H}^H\mathbf{e}. \end{aligned} \tag{6.9}$$

$$\tag{6.10}$$

4) *Removing the decoded noise component:* The computed $\widehat{w}_{masked}$ is added to $\widetilde{\mathbf{w}}^{(l)}_{unmasked}$ at the position of the masked component. Then, in the result, the complex amplitude at the position of the next largest magnitude of $\widetilde{\mathbf{w}}^{(l)}$ is masked. It can be observed that the estimation error in (6.9), $(\widehat{w}_{masked} - w_{masked})$, propagates to the decoding of the next

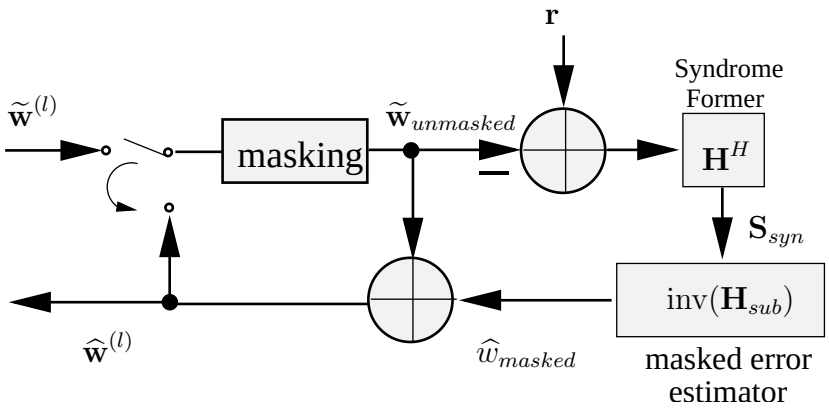

Figure 6.11: Block diagram of the syndrome decoder

component. The covariance of the estimation error can be calculated as

$$\begin{aligned}\Phi_{syn} &= \mathrm{E}\{(\widehat{w}_{masked} - w_{masked})(\widehat{w}_{masked} - w_{masked})^H\} \\ &= N_e(\mathbf{H}_{sub}^H\mathbf{H}_{sub})^{-1}, \\ &= \frac{n}{n-m}N_e, \end{aligned} \tag{6.11}$$

where $N_e\mathbf{I}_n = \mathrm{E}\{ee^H\}$, N_e denotes the power of e and I_n is the identity matrix of size $n \times n$. In Eq. (6.11), small eigenvalues of $\mathbf{H}_{sub}^H\mathbf{H}_{sub}$ will lead to a large estimation error. Since $\mathbf{H}_{sub}$ has dimensions $(n-m) \times 1$, the amplification factor, $(\mathbf{H}_{sub}^H\mathbf{H}_{sub})^{-1}$, can be calculated as $n/(n-m)$. Hence, an increase in $n-m$ leads to a better error rate performance of the receiver. It can also be observed that successive decoding of the noise components has the advantage that the non-consecutive zeros or pilot tones can also be used in the decoding. Since the matrix $\mathbf{H}_{sub}$ has rank 1, there is no condition on the position of zeros or known data. Instead of decoding the noise components one by one, the alternative way is to decode up to $n-m$ noise components jointly. However, in this case, there are two main disadvantages.

- A large amplification factor can be expected, see Subsection 5.4.1.
- The non-consecutive zeros or pilot tones cannot be used, see Property 5.2.

At each iteration $l \geq 0$, the algorithm continues until all positions in $\widetilde{\mathbf{w}}^{(l)}$ are estimated. The output vector $\widehat{\mathbf{w}}^{(l)}$ contains the complex noise amplitudes of the lth iteration that are successively decoded by the syndrome decoding. The iteration process continues by subtracting the vector $\widehat{\mathbf{w}}^{(l)}$ from $\mathbf{r}$. The complexity of the syndrome decoding depends on the number of masked positions at each iteration. On the average, An impulses occur during the transmission of one OFDM symbol, where A denotes the probability of having an impulsive noise sample. For small σ_g^2 , $A = 0.1$ and $n = 32$, it corresponds to a successive decoding of about 3 impulses on average.

Fig. 6.12, Fig. 6.13 and Fig. 6.14 show the SER as a function of $E_s/2\sigma_g^2$ for the transmission over the two-state impulsive noise channel model. We consider four zeros in 32-OFDM and eight in 64-OFDM, respectively. We observe in Fig. 6.12 that using syndrome decoding at the iterative receiver results in a lower SER beyond 12 dB. A large gain can be obtained after only two iterations. Since more iterations improve the results slightly, they are not shown in the figure. For $n = 64$, the performance gain due to the syndrome decoding is significant with decreasing σ_g^2, see Fig. 6.13 and Fig. 6.14. For $\Delta = 10^{-2}$, the SER drops to 10^{-6} at 27 dB and for $\Delta = 10^{-3}$ at 18 dB, respectively.

Based on the above results and numerous other simulations employing different values for n, m, A and Δ, the following characteristics can be identified.

- The codeword n plays an important role for the algorithm. For $n > 64$, the syndrome decoder is not needed, since iterative noise suppression algorithm with clipping and nulling technique converges after two iterations. For $n \leq 64$, the syndrome decoder can be applied to decrease the SER for an increased complexity in the receiver design.

- Increasing $(n - m)$ leads to a better SER performance since the amplification factor in (6.11) decreases. Thus, an additional gain can also be achieved by adding zeros to the information for the decoding of impulsive noise errors only.

- A determines the number of impulsive noise errors in an OFDM block. Hence, decreasing A increases the convergence speed, whereas increasing A has the opposite effect.

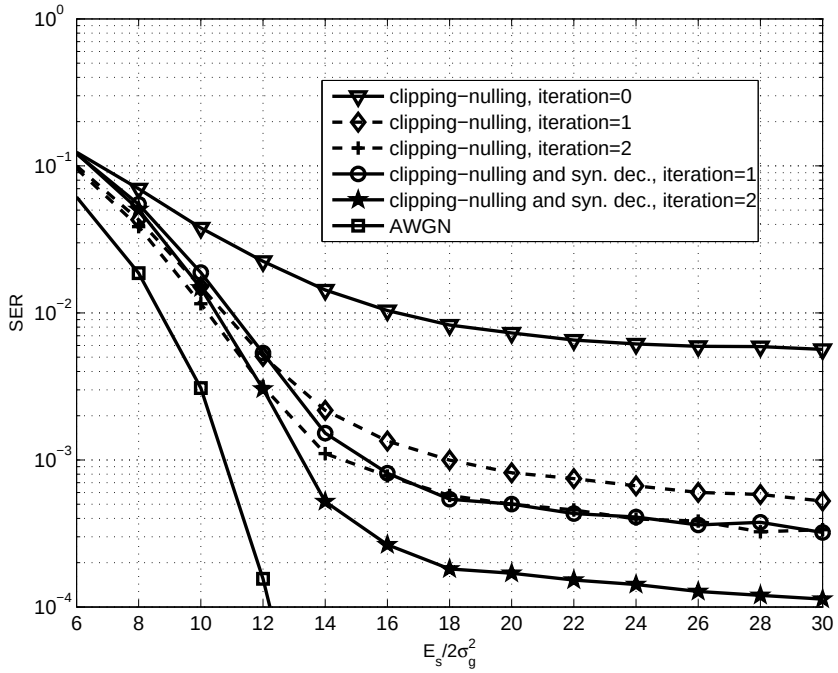

Figure 6.12: SER of the iterative 32-OFDM transmission using 4-QAM over the impulsive noise channel with A=0.1 and $\Delta = 10^{-2}$.

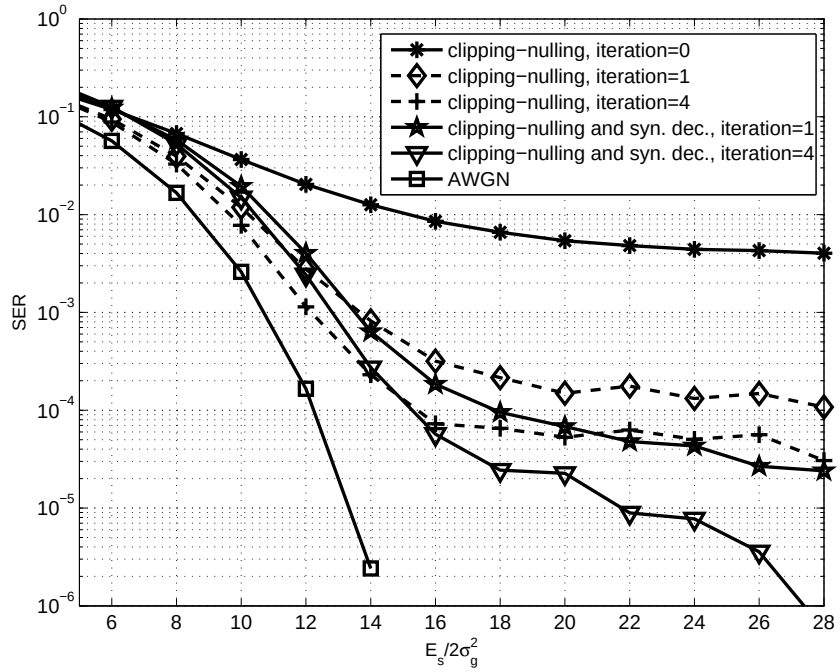

Figure 6.13: SER of the iterative 64-OFDM transmission using 4-QAM over the impulsive noise channel with A=0.1 and $\Delta = 10^{-2}$.

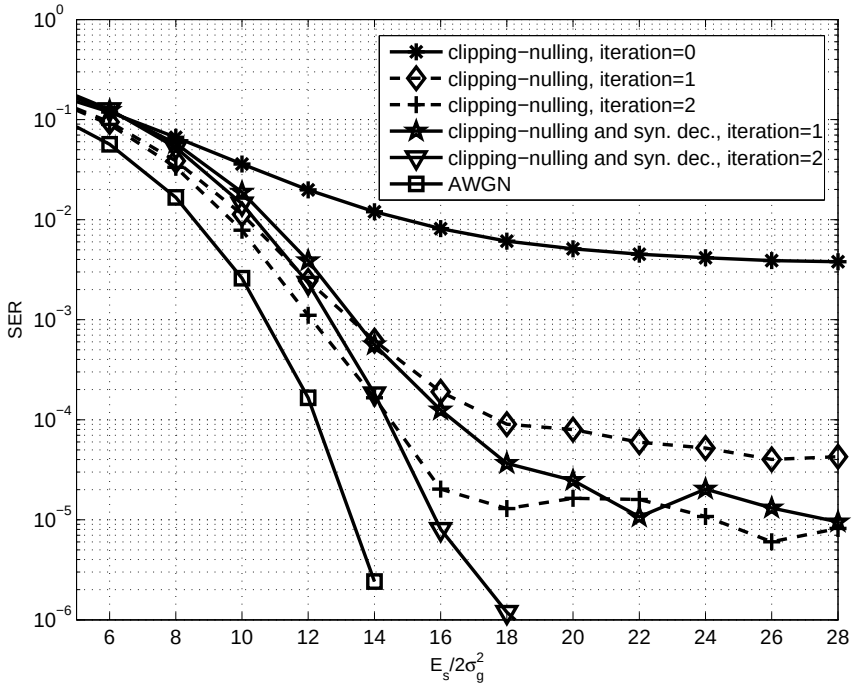

Figure 6.14: SER of the iterative 64-OFDM transmission using 4-QAM over the impulsive noise channel with A=0.1 and $\Delta = 10^{-3}$.

- The parameter $\Delta = \sigma_c^2/\sigma_i^2$ has an influence on the iterative receiver's performance. As σ_i^2 increases, the impulsive noise magnitudes can be clearly distinguished by the threshold detector in the frequency domain. Hence, the SER of the iterative receiver also decreases, see Fig. 6.13 and Fig. 6.14.

- Finally, the non-consecutive zeros or pilot tones can be used in the syndrome decoding.

6.5 Conclusion

The performance of the iterative impulsive noise suppression algorithm depends on the number of subcarriers in OFDM. In this chapter, we first show that applying the clipping and nulling technique at the front of the iterative algorithm decreases the SER significantly, and it leads to a fast convergence for $n > 64$. Furthermore, we demonstrate a novel syndrome decoder which uses the zeros or pilot tones that are transmitted for synchronization or other purposes. Instead of decoding all noise components jointly, a successive decoding is applied to achieve a low SER.

Chapter 7

Summary

7.1 Summary of the thesis

This thesis can be divided into two parts. In the first part, Chapters 3 and 4, a combined coding and modulation scheme based on the single carrier modulation has been proposed. Accordingly, it has been shown that high coding gains can be achieved at the same information rate without leading to an essential change in the power spectral density (PSD). In the second part of the thesis, Chapters 5 and 6, OFDM modulation has been considered. If the DFT of the transmitted OFDM symbol contains a small number of zeros or known data, there is a similarity between the IDFT and RS encoder. It has been shown that the similarity can be used to correct the impulsive noise errors. In this section, the main results obtained in the different chapters are summarized.

After a short introduction, the channel models have been introduced in the second chapter. The Middleton's AWCN model has been considered as a statistical model for the impulsive noise channel. The model covers both impulsive and background noise, and it can be interpreted as an infinite number of parallel AWGN channels with increasing variances. It was also shown that the AWCN channel model can be simplified to a two-state impulsive noise channel model by considering only two channels.

In the third chapter, it has been proposed to combine the design of coding and modulation by controlling the minimum channel symbol duration (the minimum duration in which the channel symbol stays constant) of a set of block waveforms defined in a constant time. An RLL encoder has been introduced between the channel encoder and the constant envelope modulator. In RLL coding, the data sequences are encoded into a restricted set of sequences. Runlength-limited simply means that the number of symbols in the intervals, where the signal stays constant, are bounded from below by the parameter d. The parameter d can be used to control the minimum channel symbol duration and the power spectrum characteristics of the block waveforms. The key idea of the proposed RLL coded modulation scheme is to map the uncoded set of block waveforms into the set of RLL coded block waveforms such that the minimum Euclidean distance in the set is increased. Hence, a soft decision decoding can be applied to obtain high coding gains at the same information rate without leading to an essential change in the PSD. The concatenation of a RS code with the RLL code has been studied. In this case, the number of bits per RS symbol is chosen as the length of the input block of the RLL encoding. It has been shown for BPSK

and 2-FSK modulations that the same amount of information can be transmitted within the same minimum channel symbol duration with coding gains of 4 and 5 dB at BER of 10^{-5} compared with the uncoded BPSK and 2-FSK modulations, respectively.

As a practical application of the proposed RLL coded modulation scheme, we consider PLC channel, where there is a limitation on the output voltage and bandwidth. FSK modulation is suited for the PLC, where the transmission of only one frequency per time unit leads to a constant envelope signal, and it is an attractive solution for the implementations that are oriented to low data rates. For the RLL coded 2-FSK modulation, noncoherent maximum-likelihood block decoding scheme has been investigated for the AWGN and Middleton's AWCN channel models. It was shown that by estimating the channel state information, performance improvement can be obtained for the heavily disturbed impulsive noise channel. Additionally, receiver performance has been improved by making a noncoherent decision over a block length, so that the combination loss due to the squaring in the envelope detection is avoided. Furthermore, the effect of the narrow band noise on the 2-FSK transmission scheme has been addressed. Simulation results showed that under a strong frequency disturbance, the decision can be made based on a single frequency.

In the third chapter, we have described construction algorithms for RLL codes with one-bit look-ahead encoding techniques having a minimum Hamming distance larger than one. A block-code is said to be look-ahead if the encoding is done as a function of not only the present and past inputs, but also of a finite number of inputs yet to come. The advantage of this type of encoding is that codes with higher rates can be constructed without a need of merging bits during the concatenation of the RLL codewords. At the decoder side, the RLL codewords can be decoded independently, which does not lead to an error propagation between the decoded codewords. We have presented several new high rate RLL codes with the specified d-constraint and the minimum Hamming distance. We also provide efficient bounds on the code sizes and code rates. The constructed codes are used in the RLL coded modulation scheme.

In the forth chapter, OFDM modulation has been considered. Impulsive noise may lead to a significant performance loss in a conventional OFDM receiver designed for the AWGN channel. The reason for this is that the DFT at the OFDM receiver spreads the impulse energy over a portion of information symbols. It has been shown that if the DFT of the transmitted OFDM symbol contains some amount of consecutive zeros or known data appended to the information, the OFDM modulator can be seen as a RS encoder. An impulsive noise error correction algorithm has been proposed in this chapter. First, the number and position of impulsive noise errors are estimated by a threshold-based procedure. Then, the detected impulsive noise errors are corrected by using least-squares estimation. The analysis on the least-squares estimation noise showed that the error rate performance of the least-squares estimator receiver is limited.

In the fifth chapter, an iterative decoder scheme has been proposed. Simple iterative impulsive noise suppression algorithms can improve the performance by exploiting the noise structure in the time and frequency domain. In these algorithms, the number of subcarriers, n, plays an important role. For large $n > 256$, the convergence speed is fast, whereas, for $n \leq 256$, the algorithms converge slowly or even not at all. This is mainly due to the Gaussian approximation used in the algorithms for the distribution of the transformed random

variables after DFT or IDFT. As n increases, the approximation becomes more accurate. In this chapter, it has been shown that applying the clipping and nulling technique at the front of the iterative algorithm decreases the SER significantly, and it leads to a fast convergence for $n > 64$. Furthermore, a novel syndrome decoder has been proposed in the iterative impulsive noise suppression algorithm to improve the error rate performance for $n \leq 64$. The decoder uses the zeros or pilot tones that are transmitted for synchronization or other purposes. Instead of decoding all noise components jointly, a successive decoding has been applied to achieve a low SER. This approach has the advantage that the non-consecutive zeros or pilot tones can also be used in the decoding algorithm.

Bibliography

[1] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, No. 1, pp. 55-67, Jan. 1982.

[2] H. Imai, S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. Inform. Theory*, vol.IT-23, pp. 371-377, May. 1977.

[3] J. B. Anderson, D. P. Taylor, "A bandwidth-efficient class of signal-space codes," *IEEE Trans. Inform. Theory*, vol.IT-24, No. 6, pp. 703-712, Nov. 1978.

[4] J. B. Anderson, A Svensson W. Helstrom, *Coded modulation systems*. Kluwer Academic / Plenum Pub., New York, U.S.A., 2003.

[5] EN 50 065-1, "Signalling on low voltage electrical installations in the frequency range 3kHz to 148.5 kHz; Part 1: General requirements, frequency bands and electromagnetic disurbances." *CENELEC*, Brussels, 1991.

[6] E. Biglieri, "Coding and modulation for a horrible channel," *IEEE Communications Magazine*, vol. 41, no. 5, pp. 92-98, May 2003.

[7] C. W. Helstrom, *Probability and stochastic processes for engineers (2nd edition)*. Macmillan, New York, U.S.A., 1991.

[8] D. Middleton, "Canonical and quasi-canonical probability models of class A interference," *IEEE Trans. Electromagn. Compat.*, vol. 25, No. 2, pp. 76-106, May 1983.

[9] A. D. Spaulding, D. Middleton, "Optimum reception in an impulsive interference environment-part *I*: coherent detection," *IEEE Trans. Commun.*, vol. 25, No. 9, pp. 910-923, Sep. 1977.

[10] M. Zimmerman, K. Dostert, "Analysis and modelling of impulsive noise in broad-band powerline communications," *IEEE Trans. Electromagn. Compat.*, vol. 44, No. 1, pp. 249-258, Feb. 2002.

[11] O. G. Hooijen, *Aspects of residential power line communications*. Shaker Verlag, Aaachen, Germany, 1998.

[12] K. Yamauchi, N. Takahasi, M. Maeda, "Parameter measurement of class A interference on powerline," *Transactions. IICE*, vol. 72, no. 1, pp. 7-9, Jan. 1989.

[13] S. Miyamoto, M. Katayama, N. Morinaga, "Performance analysis of QAM systems under class A impulsive noise environment," *IEEE Trans. Electromagn. Compat.*, vol. 37, no. 2, pp. 260-267, May 1995.

[14] A. J. H. Vinck, "Coded modulation for power line communications," *AEU International Journal of Electronics and Communications*, vol. 54, 2000, pp. 45-49.

[15] J. Haering, A. J. H. Vinck, "OFDM transmission corrupted by impulsive noise," in *Proc. IEEE Int. Symp. Power Line Communications (ISPLC)*, Limerick, Ireland, Apr. 2000, pp. 9-14.

[16] J. Haering, *Error Tolerant Communication Over the Compound Channel*. Shaker Verlag, Aachen, Germany, 2001.

[17] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Techn. J.*, vol. 27, pp. 379-423, July 1948.

[18] D. T. Tang, L. R. Bahl, "Block codes for a class of constrained noiseless channels," *Information and Control*, vol. 17, pp. 436-461, 1970.

[19] B. H. Marcus, R. M. Roth, P. H. Siegel, *Constrained systems and coding for recording channels*. V. S. Pless and W. C. Huffman (Editors), Elsevier Press, Amsterdam, 1998.

[20] H. D. L. Hollmann, "A block-decodable (1,8) runlength-limited rate 8/12 code," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1292-1296, July 1994.

[21] H. D. L. Hollmann, "Bounded-delay-encodable, block-decodable codes for constrained systems," *IEEE Trans. Inform. Theory*, vol. 42, No. 6, pp. 1957-1970, Nov. 1996.

[22] A. Lempel, M. Cohn, "Look-ahead coding for input-restricted channels," *IEEE Trans. Inform. Theory*, vol. IT-28, No. 6, pp. 933-937, Nov. 1982.

[23] P. A. Franaszek, "Construction of bounded delay codes for discrete noiseless channels," *IBM J. Res. Develop.*, vol. 26, pp. 506-514, July 1982.

[24] P. A. Franaszek, "On future-dependent block coding for input-restricted channels," *IBM J. Res. Develop.*, vol. 23, pp. 75-81, Jan. 1979.

[25] K. A. S. Immink, "Block-decodable runlength-limited codes via look-ahead technique," *Philips J. Res. Develop.*, vol. 46, pp. 293-310, Sept. 1991.

[26] K. A. S. Immink, *Codes for mass data storage systems*. Shannon Foundation, Eindhoven, the Netherlands, 2004.

[27] K. A. S. Immink, "A practical method for approaching the channel capacity of constrained channels," *IEEE Trans. Inform. Theory*, vol. IT-43, no. 5, pp. 1389-1399, Sept. 1997.

[28] W. G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM Techn. Discl. Bul.*, vol. 23, pp. 4633-4634, 1981.

[29] H. C. Ferreira, "Lower bounds on the minimum Hamming distance achievable with runlength constrained or dc free block codes and the synthesis of a (16,8), $d_{\min} = 4$ dc free block code," *IEEE Trans. Magn.*, vol. 20, No. 5, pp. 881-883, Sept. 1984.

[30] C. A. French, "Distance preserving run-length limited codes," *IEEE Trans. Magn.*, vol. 25, pp. 4093-4095, Sept. 1989.

[31] M. Blaum, "Combining ECC with modulation: performance comparisons," *IEEE Trans. Inform. Theory*, vol. 37, pp. 945-949, May 1991.

[32] P. Liu, Y. Lin, "A class of (d, k) block codes with single error correcting capability," *IEEE Trans. Magn.*, vol. 33, No. 5, pp. 2758-2760, Sept. 1997.

[33] P. Lee, J. K. Wolf, "A general error-correcting code construction for run-length limited binary channels," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1330-1335, Nov. 1989.

[34] E. Zehavi, J. K. Wolf, "On Runlength Codes," *IEEE Trans. Inform. Theory*, vol. 34, No. 1, pp. 45-54, Jan. 1988.

[35] J. K. Wolf, "Redundancy, the discrete Fourier transform, and impulse noise cancellation," *IEEE Trans. Commun.*, vol.COM-31, no. 3, pp. 458-461, Mar. 1983.

[36] A. Mengi, A. J. H. Vinck, "Coded frequency modulation with RLL sequences for CENELEC band applications," in *Proc. IEEE Int. Symp. Power Line Communications (ISPLC)*, Pisa, Italy, March 2007, pp. 133-137.

[37] A. Mengi, A. J. H. Vinck, "Maximum-likelihood block decoding of noncoherent coded FSK for the CENELEC band," in *Proc. IEEE Int. Symp. Power Line Communications (ISPLC)*, Jeju Island, South Korea, April 2008, pp. 117-122.

[38] A. Mengi, A. J. H. Vinck, "Impulsive noise error correction in 16-OFDM for narrowband power line communication," in *Proc. IEEE Int. Symp. Power Line Communications (ISPLC)*, Dresden, Germany, March 2009, pp. 31-35.

[39] A. Mengi, A. J. H. Vinck, "Successive impulsive noise suppression in OFDM," to be published in *Proc. IEEE Int. Symp. Power Line Communications (ISPLC)*, Rio de Janeiro, Brazil, March 2010.

[40] A. Mengi, A. J. H. Vinck, "Coded modulation with a constraint on the minimum channel symbol duration," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seoul, South Korea, June 2009, pp. 1544-1548.

[41] A. Mengi, A. J. H. Vinck, "On iterative impulsive noise suppression in OFDM," submitted to *IEEE Commun. Letters*.

[42] A. Gallopoulos, C. Heegard, P.H. Siegel, "The power spectrum of run-length-limited codes," *IEEE Trans. Commun.*, vol. 37, No. 9, pp. 906-917, Sept. 1989.

[43] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio and Electroacoustics*, vol. AU-15 , pp. 70-73, June 1967.

[44] J. G. Proakis, D. G. Manolakis, *Digital signal processing: Principles, algorithms, and applications (3rd Edition)*. Prentice Hall, N.J., U.S.A., 1996.

[45] M. Deinzer, M. Stoeger, "Integrated PLC-modem based on OFDM," in *Proc. IEEE Int. Symp. Power Line Communications (ISPLC)*, Lancaster, UK, Apr. 1999, pp. 92-97.

[46] Maxim Integrated Products, "10kHz to 490kHz OFDM-based power line communication modem," in *Abridged Data Sheet*, 2008.

[47] S. B. Weinstein, P. M. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *IEEE Trans. Commun. Techn.*, vol.COM-19, no. 5, pp. 628-634, Oct. 1971.

[48] M. Ghosh, "Analysis of the effect of impulsive noise on multicarrier and single carrier QAM systems," *IEEE Trans. Commun.*, vol. 44, No. 2, pp. 145-147, Feb. 1996.

[49] J. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Commun. Mag.*, pp. 5-14, May 1990.

[50] L. R. Turner, "Inverse of the Vandermonde matrix with applications," *NASA technical note*, D-3547, Aug 1966.

[51] S. V. Zhidkov, "Impulsive noise suppression in OFDM based communication systems," *IEEE Trans. on Consumer Electronics*, vol. 49, no. 4, pp. 944-948, Nov. 2003.

[52] H. A. Suraweera, C. Chai, J. Shentu, J. Amstrong, "Analysis of impulse noise mitigation techniques for digital television systems," in *Proc. 8th Int. OFDM Workshop (InOWo'03)*, Hamburg, Germany, Sep. 2003, pp. 172-176.

[53] S. V. Zhidkov, "Analysis and comparison of several simple impulsive noise mitigation schemes for OFDM receivers," *IEEE Trans. Commun.*, vol. 56, no. 1, pp. 5-9, Jan. 2008.

[54] R. Kumaresan, "Rank reduction techniques and burst error-correction decoding in real/complex fields," in *Proc. Asilomar Conf. on Circuits, Systems and Computers*, Pacific Grove, CA, Nov. 1985, pp. 457-461. pp. 458-461, Mar. 1983.

[55] G. R. Redinbo "Decoding real block codes: activity detection, Wiener estimation," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 609-623, Mar. 2000.

[56] F. Abdelkefi, P. Duhamel, F. Alberge, "Impulsive noise cancellation in multicarrier transmission," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 94-106, Jan. 2005.

[57] E. Kreyszig, *Advanced engineering mathematics (6th edition)*. Wiley, New York, U.S.A., 1988.

[58] M. Bossert, *Channel Coding for Telecommunications (1st edition)*. John Wiley and Sons, Inc., New York, U.S.A., 2000.

[59] A. Ben-Israel, T. N. E. Greville, *Generalized Inverses: Theory and Applications*. Wiley, New York, U.S.A., 1977.

[60] H. L. V. Trees, *Detection, estimation, and modulation theory, part I: Detection, estimation and linear modulation theory*. Wiley, New York, U.S.A., 20001.

[61] M. Schwartz, *Information, transmission, modulation, and noise (4th edition)*. McGraw-Hill Pub. Company, New York, U.S.A., 1990.

List of Abbreviations and Symbols

Abbreviations

$AWCN$	Additive White Class-A Noise
$AWGN$	Additive White Gaussian Noise
BER	Bit Error Rate
$BPSK$	Binary Phase Shift Keying
CSI	Channel State Information
DFT	Discrete Fourier Transform
ECC	Error Control Code
FFT	Fast Fourier Transform
FSK	Frequency Shift Keying
$IDFT$	Inverse Discrete Fourier Transform
i.i.d.	Independent and Identically Distributed
LSE	Least-Squares Estimation
ML	Maximum-Likelihood
$MMSE$	Minimum Mean-Squared Error
$OFDM$	Orthogonal Frequency Division Multiplexing
OOK	On-Off Keying
PEP	Pairwise Error Probability
PDF	Probability Density Function
PLC	Powerline Communications
PSD	Power Spectral Density
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RLL	Runlength-Limited
RS	Reed-Solomon
SER	Symbol Error Rate
$SBNR$	Signal-to-Gaussian Background Noise Ratio
SNR	Signal-to-Noise Ratio

Symbols

k	time instant
n	codeword length
$\mathbf{c} = (c_1, ..., c_n)$	channel input vector
$\mathbf{C}$	set of channel input vectors
$\mathbf{r} = (r_1, ..., r_n)$	channel output vector
$\mathbf{w} = (w_1, ..., w_n)$	channel noise vector
$\mathbf{g} = (g_1, ..., g_n)$	background noise vector
$\mathbf{i} = (i_1, ..., i_n)$	impulsive noise vector
$\mathbf{s} = (s_1, ..., s_n)$	channel state vector
Γ	the ratio between the background noise variance and the impulsive noise variance
Δ	the ratio between the signal variance and the impulsive noise variance
A	the impulsive noise index
$d_{\min}$	minimum Hamming distance
d	constraint on the maximum run of zeros or ones
τ	the duration of one information bit
τ'	the duration of one RLL channel symbol
ϕ	arbitrary phase introduced by the channel
$\mathbf{V}$	DFT matrix
α^{kl}	entries of $\mathbf{V}$
$\mathbf{S}_{\text{syn}}$	syndrome vector
H	parity-check matrix
t	number of non-zero components in $\mathbf{i}$
R_r	the rate of the RLL coding
C_d	the asymptotic information rate of RLL sequences
$N(n, d)$	The number of RLL sequences of length n that satisfy the d-constraint
σ^2	variance of a random variable