Iterative Decoding of Codes Over Complex Numbers for Impulsive Noise Channels

Jürgen Häring and A. J. Han Vinck, *Senior Member, IEEE*

***Abstract*—We discuss the decoding of error-correcting block codes over complex numbers for the transmission over impulsive noise channels. The encoder multiplies a vector of complex information symbols resulting from a modulation scheme, e.g., quadrature amplitude modulation (QAM), with a unitary generator matrix $\boldsymbol{G}$. Choosing the inverse Fourier transform as $\boldsymbol{G}$, the encoding procedure is similar to orthogonal frequency-division multiplex (OFDM) modulation. The maximum *a posteriori* (MAP) receiver is analyzed and a suboptimum decoder based on the turbo decoding principle is derived. Simulation results show the excellent performance of the iterative decoder.**

***Index Terms*—Complex number codes, diversity, fading channel, impulsive noise, orthogonal frequency-division multiplex (OFDM), rotation, turbo decoding.**

I. Introduction

In this paper, we discuss the decoding of block codes over complex numbers. The input to the encoder is a vector of complex numbers generated by a quadrature amplitude modulation (QAM) or phase-shift keying (PSK) modulation scheme. The encoder multiplies this information vector with a unitary generator matrix $\boldsymbol{G}$, i.e., the codewords are simply rotated versions of the information vectors, and the Euclidean distance between two information vectors is the same as between their respective codewords. Hence, on the additive white Gaussian noise (AWGN) channel, these codes achieve no coding gain. However, on various other channels, a gain can be achieved. A well-known example is the fading channel with perfect channel state information (CSI). Here, for a fixed Euclidean distance d between two codewords c, c', the pairwise error probability (PEP) can be modified by changing the distances $d_k = |c'_k - c_k|$ between the code symbols while keeping $d^2 = \sum_{k=0}^{n-1} d_k^2$ constant, see [2]. Such a modification of the "distance profile" of two codewords is introduced by the encoding operation described above. Hence, adopting a carefully chosen generator matrix $\boldsymbol{G}$ can lead to a coding gain. In literature, this coding gain is also referred to as *diversity gain*.

Since we consider a linear transform defined over the complex numbers as an encoding operation, the resulting codes are denoted by *complex number (CN) codes*. CN codes might be used "stand alone." We can also use CN codes as inner codes in a product encoding scheme where the outer code is optimized to increase the Euclidean distance between the codewords. This might be a good approach to design codes, providing both a good distance profile and a large Euclidean distance between the codewords.

In the literature, the principle of CN codes have already been applied in 1963 for the transmission over impulsive noise channels, see [14]. The concept uses a properly chosen linear transform at the sender side and its inverse at the receiver side. The transmitted data passes both transforms and is therefore unaffected, whereas the impulsive noise passes the receiver's transform only. The energy of single impulses is therefore dispersed (smeared) over a portion of information symbols, see [1]. In this way, the error floor that is typical for uncoded transmission over impulsive noise channels, see [16], is (partially) eliminated. The construction of CN codes in [14] was motivated by a convolutional type of CN codes, the so-called "smearing filters," that were originally developed for continuous systems, see [1], [6].

The principle of CN codes is also applied in a different area of communication theory. The orthogonal frequency-division multiplex (OFDM) modulation scheme, see [3], can be interpreted as a special CN code where the generator matrix $\boldsymbol{G}$ is chosen to be the inverse Fourier transform. The classical OFDM receiver is designed for the AWGN channel and, therefore, simply multiplies the received vector with $\boldsymbol{G}^{-1}$. On impulsive noise channels, this decoder disperses the energy of single impulses over several consecutive symbols, similar to the smearing-filters approach. The typical argumentation is that this makes the transmission scheme robust against impulsive noise, see [3]. However, in [7], [8], [10], it has been shown that this "decoding" approach is highly suboptimal in terms of the achievable decoding error rates since the rich structure of impulsive noise is not exploited in the decoding process.

Independently, the idea of CN codes has been developed for reliable transmission over the slowly Rayleigh-fading channel with perfect CSI. The initial idea of constructing two-dimensional rotations for QAM signal constellations in [4] was soon generalized to higher dimensions. The construction of rotational transforms, i.e., CN codes, yielding a good performance is extensively discussed in [5], [13] and references therein. References [13], [15], and [18] address the decoding problem. The decoding algorithm presented in [18] is based on lattice theory and suited for codeword lengths $n \leq 32$. The algorithms derived in [13] and [15] apply the idea of decision feedback equalization (DFE). They typically require a larger n. Similar to the impulsive noise channel, also a convolutional type of CN codes has

Manuscript received April 26, 2001; revised January 30, 2003.The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Washington, DC, June 2001.

The authors are with the Institute for Experimental Mathematics, University of Essen, 45326 Essen, Germany (e-mail: j.haering@web.de; vinck@exp-math.uni-essen.de).

Communicated by R. Urbanke, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2003.810636

been developed. A comprehensive study including code construction, performance analysis, and decoding issues is given in [19].

To understand why CN codes can be applied to both, the Rayleigh-fading channel with perfect CSI and impulsive noise channels, we first compare both channels. Impulsive noise channels can be modeled as AWGN channels with randomly varying noise variance. The noise variance is assumed to be unknown to the receiver, see [12]. For the fading channel with perfect CSI, basically the same model is used. However, in contrast to the impulsive noise channel, the receiver is perfectly informed about the noise variance. The decoding algorithms presented in [13], [15], and [18] use this information and therefore cannot be applied to the impulsive noise channel. Hence, in this paper, a new decoding algorithm for a channel with both impulsive and AWGN is derived. The decoder is based on the turbo principle.

It is interesting to compare our new decoder and the DFE-based decoders presented in [13], [15] in more detail. Both concepts apply the idea of iterative decoding. As indicated earlier, the DFE decoders are provided with perfect CSI. As a consequence, the additive channel noise is Gaussian distributed (conditioned on the known channel state), and therefore the optimum minimum mean-square error (MMSE) estimators in the forward and the backward branches of the DFE decoder become linear. In contrast, on the impulsive noise channel, *linear* MMSE estimators cannot mitigate the influence of the channel noise, see [10], [16]. Hence, the DFE principle developed in [13], [15] cannot be applied to impulsive noise channels.

The paper is organized as follows. We will first define the encoding procedure (Section I-A), the channel model (Section I-B), and the optimum maximum *a posteriori* (MAP) probability decoder (Section I-C). This is followed by the definition of our new, iterative decoding algorithm (Section II). In Section III, we discuss some simulation results and conclude with Section IV.

A. Encoding

Let $\mathcal{X}$, $\mathcal{X} \subset \mathbb{C}$, denote the discrete input alphabet with cardinality $|\mathcal{X}|$, where $\mathbb{C}$ denotes the complex numbers. Let $S = (S_0, \dots, S_{n-1})$, $S_k \in \mathcal{X}$, denote the input to the CN encoder. We call the components S_k, $k = 0, \dots, n-1$, the *information symbols*. Assuming that every $S_k \in \mathcal{X}$ is transmitted equally likely, we describe the S_k as independent random variables with probability distribution

$$P(S_k) = \frac{1}{|\mathcal{X}|} \tag{1}$$

and assume

$$E\{S_k\} = 0$$
$$E\{\mathrm{Re}[S_k]^2\} = E\{\mathrm{Im}[S_k]^2\} = \frac{1}{2}\,\sigma_x^2$$

and

$$E\{\mathrm{Re}[S_k]\mathrm{Im}[S_k]\} = 0$$

where $E\{\cdot\}$ denotes the expectation operator.

Each vector S is encoded by a CN block code $\mathcal{C}$ with codewords $s = (s_0, \dots, s_{n-1})$ and code symbols $s_i \in \mathbb{C}$. Here and in the following we use capital letters to denote vectors or symbols in the domain of the information sequence (information domain) and lower case letters in the domain of the codewords (codeword domain). The encoding operation is defined by

$$s = \boldsymbol{G}S. \tag{2}$$

The $n \times n$ generator matrix $\boldsymbol{G}$ is unitary, i.e., the Euclidean distance $d(S^0, S^1)$ between two information vectors S^0, S^1 is the same as between their respective codewords s^0, s^1

$$\begin{aligned} d(s^0, s^1) &= \sqrt{(s^0 - s^1)^*(s^0 - s^1)} \\ &= \sqrt{(S^0 - S^1)^* \boldsymbol{G}^* \boldsymbol{G} (S^0 - S^1)} = d(S^0, S^1) \end{aligned}$$

where the asterisk denotes complex conjugate. Note that the encoding operation generally increases the alphabet size.

Finding generator matrices with good error correcting properties is a nontrivial problem. For the MAP decoder it has been shown in [10] that matrices suited for the Rayleigh-fading channel with perfect CSI also perform well on impulsive noise channels. Therefore, references [5], [13] can be used to find good generator matrices. The decoding algorithm derived in this paper imposes two additional restrictions for choosing $\boldsymbol{G}$: first, for the decoder's derivation it is assumed that each code symbol $s_i \in \mathbb{C}$ is calculated by summing over a large number of information symbols, see Approximations II.1 and II.2 introduced later. Loosely speaking, this means that the generator matrix must be nonsparse and n sufficiently large. Second, since matrix–vector multiplications with $\boldsymbol{G}$ and $\boldsymbol{G}^*$ determine both the encoding and decoding complexity, the applicability of fast transform algorithms should be ensured. For these reasons, the Fourier- and the Walsh–Hadamard matrix are considered in this paper. For channels with real input alphabets, e.g., the discrete cosine transform might be applied.

B. Impulsive Noise Channel Model

The sender transmits the code symbols s_k over a memoryless additive impulsive noise channel. The received symbols are given by

$$r_k = s_k + i_k + g_k \tag{3}$$

where the g_k are independent and identically distributed (i.i.d.) complex Gaussian random variables with variance σ_g^2 and probability density function (pdf)

$$p_{g_k}(x) = \frac{1}{2\pi\sigma_g^2} \exp\left(-\frac{xx^*}{2\sigma_g^2}\right).$$

The impulsive noise symbols i_k are also i.i.d. with variance σ_i^2. Their pdf is given by Middleton's Class A noise model

$$p_{i_k}(x) = e^{-A}\,\delta(x) + \sum_{m=1}^{\infty} \frac{e^{-A}A^m}{m!} \frac{1}{2\pi\sigma_m^2} \exp\left(-\frac{xx^*}{2\sigma_m^2}\right) \tag{4}$$

with the Dirac distribution $\delta(\cdot)$ and

$$\sigma_m^2 = \sigma_i^2 m/A \tag{5}$$

(see [16]). The parameter A is called the impulsive index. For small A, say $A = 0.1$, the noise is highly structured since only $(1 - e^{-A}) \approx$ 9.5% of the samples are hit by an impulse. For $A \to \infty$, the pdf becomes Gaussian. Furthermore, the parameter $T = \sigma_g^2/\sigma_i^2$ is defined as the ratio between the variance σ_g^2 of the Gaussian noise g and the variance σ_i^2 of impulsive noise component i.

C. MAP Decoding of CN Codes

At the receiver, the problem of decoding the received vector $r = (r_0, \ldots, r_{n-1})$ arises. The optimum MAP decoder minimizes the probability of a decoding error, see [20]. It, therefore, chooses the codeword $c \in \mathcal{C}$ that maximizes the *a posteriori* probability, i.e.,

$$c = \arg\max_{c \in \mathcal{C}} P(s = c|r) = \arg\max_{c \in \mathcal{C}} p(r|s = c) \tag{6}$$

where we assumed that every codeword is transmitted with equal probability $P(s = c) = 1/|\mathcal{C}|$. Since the noise is additive, we obtain

$$c = \arg\max_{c \in \mathcal{C}} \prod_{k=0}^{n-1} p_{i_k+g_k}(r_k - c_k) \tag{7}$$

where $p_{i_k+g_k}(\cdot)$ is the pdf of the channel noise $i_k + g_k$. Unfortunately, an efficient representation of the code $\mathcal{C}$, e.g., in form of a trellis with a small number of states, is not known. Hence, (7) can only be solved by brute-force search through all $|\mathcal{C}| = |\mathcal{X}|^n$ codewords. MAP decoding is therefore infeasible already for small n.

It is interesting to note that on the AWGN channel, MAP decoding can be efficiently realized based on the vector $R = \boldsymbol{G}^{-1}r$. We consider this vector for the impulsive noise channel

$$\begin{aligned} R &:= \boldsymbol{G}^{-1}r = S + \boldsymbol{G}^{-1}i + \boldsymbol{G}^{-1}g \\ &:= S + I + G. \end{aligned} \tag{8}$$

The MAP decoder now chooses the information vector

$$U = \arg\max_{U \in \mathcal{X}^n} p(R|S = U) = \arg\max_{U \in \mathcal{X}^n} p_{I+G}(R - U) \tag{9}$$

equivalent to (6). Since the components I_k of $I = \boldsymbol{G}^{-1}i$ are statistically dependent, the pdf $p_{I+G}(\cdot)$ of the sum $I + G$ cannot be written as a product similar to (7). Therefore, solving (9) requires the evaluation of $p(R|S = U)$ for all $|\mathcal{X}|^n$ possible information vectors U. This results again in a decoding complexity of $\mathcal{O}(|\mathcal{X}|^n)$. On the one hand, the statistical dependencies among the I_k make decoding a hard problem, on the other hand, they are the basis for the error-correcting properties of CN codes.

II. Iterative Decoding Algorithm

In this section, we introduce a new, suboptimum decoding algorithm for CN codes. The block diagram of the algorithm is depicted in Fig. 1. Two information-exchanging estimators, one in the codeword and one in the information domain, are applied to obtain an estimate for the impulsive noise i. Both estimators use only partial statistical information from their inputs which makes a low-complexity realization possible. The result from the iterative scheme is used to increase the reliability of the decision in the final decoding step. The decoded vector U is an estimate of the transmitted information vector S.

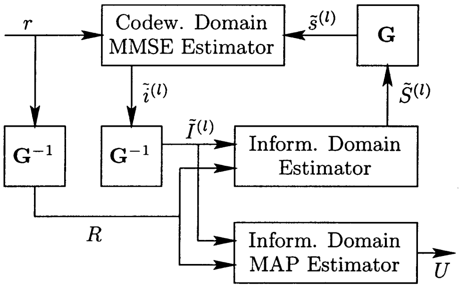

Fig. 1. Block diagram of the iterative decoding algorithm with U denoting the final result of the decoding operation.

In the following, we will first introduce a linear model that is used to approximate all random vectors occurring in the algorithm (Section II-A). In (Section II-B), we will then discuss the influence of the unitary transforms $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$ on various signals and introduce two important approximations. Based on them, the low-complexity codeword- and the information-domain estimators are designed in (Section II-D) and (Section II-E). In (Section II-G), we finally give an exact definition of the algorithm depicted in Fig. 1.

A. Linear Signal Model

To develop the estimators used in the algorithm, see Fig. 1, knowledge about the statistical properties of their random input vectors is required. Since an exact description is too difficult, we *approximate* the inputs by the simple linear model introduced in the following.

First, we consider the codeword domain. Using the independent random vectors s, i, and g, any arbitrary random vector y of length n can be described by

$$y = \tilde{y} + e = \alpha_s s + \alpha_i i + \alpha_g g + e \tag{10}$$

where $e = y - \tilde{y}$ is the error term of the linear model $\tilde{y}$. We define the average squared error of this representation by

$$\phi = \frac{1}{n} \sum_{k=0}^{n-1} |y_k - \alpha_s s_k - \alpha_i i_k - \alpha_g g_k|^2 \tag{11}$$

where ϕ is a random variable depending on the actual realizations of s_k, i_k, g_k. The α-coefficients are chosen to minimize the average error

$$E\{\phi\} = \int_{-\infty}^{\infty} \phi p(y, s, i, g)\, dy\, ds\, di\, dg \tag{12}$$

where $E\{\cdot\}$ denotes the expectation operator. Calculating this linear MMSE estimate analytically is difficult. Hence, based on measured data, we use the linear regression method (linear least squares estimation) to evaluate the α-coefficients and the variance $\sigma_e^2 := \frac{1}{2} E\{|e_k|^2\}$, see, e.g., [11]. It is well known that the linear least squares estimate is unbiased, i.e., for the error term in (10), $E\{e_k\} = 0$ holds. In the following, we will additionally *assume* that e is statistically independent of the linear model $\tilde{y}$.

The linear signal model is now applied to describe the output $\tilde{\imath}^{(l)}$ of the codeword domain estimator

$$\tilde{\imath}^{(l)} = \alpha_s^{(l)} s + \alpha_i^{(l)} i + \alpha_g^{(l)} g + e^{(l)} \tag{13}$$

(see Fig. 1) with the scalar coefficients $\alpha^{(l)}$ and the error vector $e^{(l)}$. The superscript (l) denotes the iteration step in the algorithm. Using (13), it is straightforward to calculate the transform $\tilde{I}^{(l)} = \boldsymbol{G}^{-1}\tilde{\imath}^{(l)}$ as done in the algorithm (see Fig. 1)

$$\tilde{I}^{(l)} = \alpha_s^{(l)} S + \alpha_i^{(l)} I + \alpha_g^{(l)} G + E^{(l)} \tag{14}$$

where the notation $E^{(l)} = \boldsymbol{G}^{-1}\tilde{e}^{(l)}$, $S = \boldsymbol{G}^{-1}s$, $I = \boldsymbol{G}^{-1}i$, and $G = \boldsymbol{G}^{-1}g$ is used. Obviously, the scalar coefficients are invariant to the transform and only the "basis" vectors of the linear expansion changed, i.e., in the information domain all vectors are represented as a linear combination of S, I, and G. Hence, the output $\tilde{S}^{(l)}$ of the information domain estimator is modeled by

$$\tilde{S}^{(l)} = \beta_S^{(l)} S + \beta_I^{(l)} I + \beta_G^{(l)} G + D^{(l)} \tag{15}$$

with the scalar coefficients $\beta^{(l)}$ and the error term $D^{(l)}$. With the notation $d^{(l)} = \boldsymbol{G}\tilde{D}^{(l)}$ the transformation $\tilde{s}^{(l)} = \boldsymbol{G}\tilde{S}^{(l)}$ is given by

$$\tilde{s}^{(l)} = \beta_s^{(l)} s + \beta_i^{(l)} i + \beta_g^{(l)} g + d^{(l)}. \tag{16}$$

In the next section, it will be shown that for deriving the decoding algorithm, it is sufficient to know the variances of the error terms $e^{(l)}$ and $D^{(l)}$. With this, the advantage of the linear signal model is obvious: all random vectors are sufficiently described by only four parameters, i.e., three scalar coefficients and the variance of the error term. Since these parameters are invariant to the multiplication with $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$, respectively, they also describe the transformed random vectors.

B. Signal Transformation

The decoding algorithm employs two estimators, see Fig. 1. Their complexity will be high if the components of their input vectors are statistically dependent, see e.g., Section I-C, where MAP decoding in the information domain is discussed. In our case, the inputs r, $\tilde{s}^{(l)}$, R, and $\tilde{I}^{(l)}$ of the estimators are all vectors with statistically dependent components as shown by the linear signal representations in (3), (8), (14), and (16): in every representation at least one vector with statistically dependent components is used, e.g., $I = \boldsymbol{G}^{-1}i$ in (14). In all cases, the statistical dependencies are introduced by the linear transforms $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$, respectively.

To make a low-complexity estimator realization possible, we approximate all "basis" vectors and error terms in the linear signal models and, therefore, also the inputs to the estimators as vectors of i.i.d. components. Since the linear transform introduces the dependencies, the following approximation is introduced.

Approximation II.1: Let $x = \{x_0, \ldots x_{n-1}\}$ be a vector of i.i.d. complex random variables with $E\{x_k\} = 0$ and $E\{x_k x_k^*\} = 2\sigma_x^2$. Then, the components y_k of the vector $y = \boldsymbol{G}x$ are approximated as i.i.d. complex random variables with $E\{y_k\} = 0$ and $E\{y_k y_k^*\} = 2\sigma_x^2$. The same is assumed for the transform $y = \boldsymbol{G}^{-1}x$. This approximation is motivated by Lemma V.1 given in the Appendix. Here, with some additional assumptions about the x_i, it is shown that for $n \to \infty$ the y_i are i.i.d. Gaussian random variables.

TABLE I
EXACT AND APPROXIMATED MARGINAL PDFS FOR THE VECTORS USED IN THE LINEAR SIGNAL MODEL

	codeword domain	information domain
s_k	$\mathcal{N}(0,\sigma_s^2)$	$P(S_k)$, see eq. (1)
i_k	$p(i_k)$, see eq. (4)	$\mathcal{N}(0,\sigma_i^2)$
g_k	$\mathcal{N}(0,\sigma_g^2)$	$\mathcal{N}(0,\sigma_g^2)$
d_k	$\mathcal{N}(0,\sigma_{d^{(l)}}^2)$	pdf unknown
e_k	pdf unknown	$\mathcal{N}(0,\sigma_{e^{(l)}}^2)$

If we apply the approximation, the marginal pdf $p_{y_k}(\cdot)$ of the vector components y_k has to be determined. The pdf is usually obtained by integrating over all y_j, $j \neq k$, in the joint pdf $p(y_0, \ldots, y_{n-1})$. In general, this calculation is a difficult task. Therefore, another approximation is introduced.

Approximation II.2: Let $x = \{x_0, \ldots x_{n-1}\}$ be a vector of i.i.d. complex random variables with $E\{x_k\} = 0$, $E\{\mathrm{Re}[x_k]^2\} = E\{\mathrm{Im}[x_k]^2\} = \sigma_x^2$, and $E\{\mathrm{Re}[x_k]\mathrm{Im}[x_k]\} = 0$. Then, the components y_i, $i = 0, 1, \ldots, n-1$ of the vectors $y = \boldsymbol{G}x$ and $y = \boldsymbol{G}^{-1}x$ are approximated as complex Gaussian random variables with $E\{y_k\} = 0$, $E\{\mathrm{Re}[y_k]^2\} = E\{\mathrm{Im}[y_k]^2\} = \sigma_x^2$, and $E\{\mathrm{Re}[y_k]\mathrm{Im}[y_k]\} = 0$, respectively. Let $\mathcal{N}(0, \sigma_x^2)$ denote this pdf. In the limiting case when $n \to \infty$, the approximation is exact since the statement follows from the central limit theorem, see Lemma V.1 given in the Appendix. For finite n, the accuracy of the approximation strongly depends on the size of n and the shape of the pdf of x_i.

We now apply both approximations and then resume the results in Table I.

1) Consider the transform $s = \boldsymbol{G}S$ (encoding operation). The components S_k of the information vector S are i.i.d. random variables. Their probability distribution is given by (1). Since S fulfills all assumptions to apply the approximations, we approximate s as a vector of i.i.d. random variables s_k with pdf $\mathcal{N}(0, \sigma_s^2)$.

2) The impulsive noise i is a vector of i.i.d. random variables distributed according to (4). Since i fulfills the assumptions to apply the approximations, we model $I = \boldsymbol{G}^{-1}i$ as a vector of i.i.d. random variables I_k with pdf $\mathcal{N}(0, \sigma_i^2)$.

3) Since g is a vector of i.i.d. complex Gaussian random variables, the components of $G = \boldsymbol{G}^{-1}g$ are also i.i.d. complex Gaussian distributed. This means that the Gaussian distribution is invariant to the multiplication with $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$ and no approximation is needed, see, e.g., [11].

4) The pdf of the error term $e^{(l)}$ in (3) is not known and it is difficult to calculate from the formulas of the estimators developed in the next section. As we will see in Section II-D, $e^{(l)}$ can be approximated as a vector of i.i.d. random variables. For symmetry reasons, all assumptions of the approximations hold, and we model $E^{(l)} = \boldsymbol{G}^{-1}e^{(l)}$ as a vector of i.i.d. random variables with pdf $\mathcal{N}(0, \sigma_{e^{(l)}}^2)$.

5) The pdf of the error term $D^{(l)}$ in (15) is also not known. Using the same arguments as before, $d^{(l)} = \boldsymbol{G}D^{(l)}$ is approximated as a vector of i.i.d. random variables with pdf $\mathcal{N}(0, \sigma^2_{D^{(l)}})$.

C. Estimation Criteria

To design the estimators used within the algorithm, see Fig. 1, a cost function or estimation criterion has to be chosen for each estimator. In order to minimize the probability of a decoding error, the estimator in the final decoding step is based on the MAP criterion. For both other estimators, the MAP and the MMSE criteria are considered. The MMSE estimator minimizes the error $\epsilon_{\text{MSE}} = E\{|y - \tilde{y}|^2\}$, where $\tilde{y}$ denotes the estimate obtained for the actual value y. In the information domain, both the MAP and the MMSE estimator are derived and their performance will be compared. In the codeword domain, the mixed-type random variable i_k has to be estimated, i.e., i_k has a discrete and a continuous part, see (4). Hence, using the MAP estimator is not reasonable, and we, therefore, only consider MMSE estimation.

D. Codeword Domain MMSE Estimation

The codeword domain estimator uses the input vectors

$$r = s + i + g \tag{17}$$

$$\tilde{s}^{\,(l-1)} = \beta_s^{(l-1)} s + \beta_i^{(l-1)} i + \beta_g^{(l-1)} g + d^{(l-1)} \tag{18}$$

where r is the received vector, and $\tilde{s}^{\,(l-1)}$ is an estimate for s provided by the information domain estimator in the preceding iteration, see (16). To simplify the notation, the superscript (l) will be omitted in the following.

According to the approximations summarized in Table I, the highly structured impulsive noise i_k is the only non-Gaussian signal component in the codeword domain and can, therefore, be well distinguished from s_k, g_k, and d_k. As a consequence, the codeword domain estimator is applied to estimate the impulsive noise vector i. According to Approximation II-1, the inputs r and $\tilde{s}$ to the codeword domain estimator are both modeled as vectors of i.i.d. random variables. Hence, the MMSE estimate for each i_k is a function of only two complex numbers, i.e., r_k and $\tilde{s}_k$. It is given by the conditional expected value

$$\tilde{i}_k = E\{i_k | r_k, \tilde{s}_k\} = \int_{-\infty}^{\infty} x p\left(i_k = x | r_k, \tilde{s}_k\right) \, dx$$

(see [11], [17]). Note that the above integral is two-dimensional since i_k is complex. With (17), (18), and Table I, straightforward evaluation of the integral yields

$$\tilde{i}_k = \frac{\sum\limits_{m=0}^{\infty} \frac{A^m}{m!} \frac{b_{\text{Re}}(r, \tilde{s}) + j b_{\text{Im}}(r, \tilde{s})}{2(\sigma_r^2 \sigma_{\tilde{s}}^2 - \text{cov}(r, \tilde{s}))} \tilde{p}_{\text{Re}}(r, \tilde{s}) \tilde{p}_{\text{Im}}(r, \tilde{s})}{\sum\limits_{m=0}^{\infty} \frac{A^m}{m!} \tilde{p}_{\text{Re}}(r, \tilde{s}) \tilde{p}_{\text{Im}}(r, \tilde{s})} \tag{19}$$

with variances and covariance

$$\sigma_r^2 = \sigma_s^2 + \sigma_m^2 + \sigma_g^2$$

$$\sigma_{\tilde{s}}^2 = \beta_s^2 \sigma_s^2 + \beta_i^2 \sigma_m^2 + \beta_g^2 \sigma_g^2 + \sigma_D^2$$

$$\text{cov}(r, \tilde{s}) = \beta_s \sigma_s^2 + \beta_i \sigma_m^2 + \beta_g \sigma_g^2$$

respectively. The functions $b(r, \tilde{s})$ and $\tilde{p}(r, \tilde{s})$ are given by

$$b(r, \tilde{s}) = 2\sigma_m^2 \left[\left(\sigma_{\tilde{s}}^2 - \beta_i \text{cov}(r, \tilde{s}) \right) r + \left(\beta_i \sigma_r^2 - \text{cov}(r, \tilde{s}) \right) \tilde{s} \right]$$

$$\tilde{p}(r, \tilde{s}) = \frac{\exp\left(-\frac{1}{2} \frac{\sigma_{\tilde{s}}^2 r^2 - 2\text{cov}(r, \tilde{s}) r \tilde{s} + \sigma_r^2 \tilde{s}^2}{\sigma_r^2 \sigma_{\tilde{s}}^2 - \text{cov}(r, \tilde{s})^2} \right)}{\sqrt{\sigma_r^2 \sigma_{\tilde{s}}^2 - \text{cov}(r, \tilde{s})^2}}.$$

In (19), the abbreviations

$$\tilde{p}_{\text{Re}}(r, \tilde{s}) := \tilde{p}(\text{Re}\{r\}\text{Re}\{\tilde{s}\})$$

and

$$\tilde{p}_{\text{Im}}(r, \tilde{s}) := \tilde{p}(\text{Im}\{r\}\text{Im}\{\tilde{s}\})$$

are used, where the operators $\text{Re}\{\cdot\}$ and $\text{Im}\{\cdot\}$ evaluate the real and imaginary parts of a complex number, respectively. The same abbreviation is used for $b_{\text{Re}}(r, \tilde{s})$ and $b_{\text{Im}}(r, \tilde{s})$. Note that in (19), the values σ_r^2, $\sigma_{\tilde{s}}^2$, and the covariance are different for every sum term since σ_m^2 changes according to (5). Moreover, the sum terms rapidly decrease as m increases due to the "m!"-term in the denominator. Hence, in a practical situation, it is sufficient to compute only the first few sum terms.

To estimate the whole vector $\tilde{i}$, (19) has to be calculated once for every component $\tilde{i}_k$. The complexity of this procedure grows linear with the codeword length n.

The result of the estimation procedure is now expressed in terms of the linear signal model, see (13). Here, the error term $e^{(l)}$ is modeled as a vector of i.i.d. random variables since all estimates $\tilde{i}_k$ have been calculated independently. We used this i.i.d. model already to justify the approximation of the pdf of $E_k^{(l)}$ given in Table I.

Finally, it should be noted that the calculation of $r - \tilde{i}$ gives the MMSE estimate for $s + g$. Since the information domain estimator uses R as an input, see Fig. 1, the MMSE estimator for $s + g$ and i in the codeword domain lead to equivalent algorithms, respectively.

E. Information Domain MMSE Estimation

The information domain estimator uses the inputs

$$R = \boldsymbol{G}^{-1} r = S + I + G \tag{20}$$

$$\tilde{I}^{\,(l)} = \alpha_s^{(l)} S + \alpha_i^{(l)} I + \alpha_g^{(l)} G + E^{(l)} \tag{21}$$

where R is the received vector transformed into the information domain, and $\tilde{I}^{\,(l)}$ is the transformed output of the codeword domain estimator, see (14). To simplify notation, the superscript will be omitted in the following.

Equivalent to the estimator in the codeword domain, the MMSE estimate of each S_k is given by the conditional expectation

$$\tilde{S}_k = E\{S_k | R_k, \tilde{I}_k\} = \sum_{x \in \mathcal{X}} x P\left(S_k = x | R_k, \tilde{I}_k \right). \tag{22}$$

Using (20), (21), and Table I, the integral is solved by a straightforward calculation

$$\tilde{S}_k = \frac{\sum\limits_{x\in\mathcal{X}} x\tilde{p}_{\mathrm{Re}}(R,\, \tilde{I},\, x)\tilde{p}_{\mathrm{Im}}(R,\, \tilde{I},\, x)}{\sum\limits_{x\in\mathcal{X}} \tilde{p}_{\mathrm{Re}}(R,\, \tilde{I},\, x)\tilde{p}_{\mathrm{Im}}(R,\, \tilde{I},\, x)}. \tag{23}$$

With the moments

$$\sigma_R^2 = \sigma_i^2 + \sigma_g^2$$

$$\sigma_{\tilde{I}}^2 = \alpha_i^2\sigma_i^2 + \alpha_g^2\sigma_g^2 + \sigma_e^2$$

$$\mathrm{cov}(R,\, \tilde{I}) = \alpha_i\sigma_i^2 + \alpha_g\sigma_g^2$$

we define the functions

$$\begin{aligned} c(R,\, \tilde{I},\, x) = {} & \sigma_{\tilde{I}}^2(R-x)^2 - 2\mathrm{cov}(R,\, \tilde{I})(R-x), \\ & (\tilde{I} - \alpha_s x) + \sigma_R^2(\tilde{I} - \alpha_s x)^2 \end{aligned} \tag{24}$$

$$\tilde{p}(R,\, \tilde{I},\, x) = \frac{\exp\left(-\frac{1}{2}\,\frac{c(R,\, \tilde{I},\, x)}{\sigma_r^2\sigma_{\tilde{I}}^2 - \mathrm{cov}(R,\, \tilde{I})^2}\right)}{\sqrt{\sigma_r^2\sigma_{\tilde{I}}^2 - \mathrm{cov}(R,\, \tilde{I})^2}}.$$

Again, the subscripts Re and Im indicate that only the real and imaginary parts of the complex arguments are used in the function, respectively.

Similarly to the codeword domain, the estimation has to be carried out n times to obtain all components of the vector $\tilde{S}$. The result $\tilde{S}$ of the estimation procedure is now represented with the linear signal model already given in (15). Because the estimates $\tilde{S}_k$ are calculated from random variables approximated as i.i.d., we also model the components of the vector $\tilde{S}$ and the error term $D^{(l)}$ as i.i.d. This model has already been used to justify the approximation for the pdf of $d_k^{(l)}$ given in Table I.

F. Information Domain MAP Estimation

The MAP estimator chooses the symbol $\tilde{S}_k$ satisfying

$$\tilde{S}_k = \arg\max_{x\in\mathcal{X}} p\left(S_k = x | R_k,\, \tilde{I}_k\right).$$

The distribution $P(S_k | R_k,\, \tilde{I}_k)$ has already been calculated in the derivation of the MMSE information domain estimator, see (22). Using this result, straightforward calculation yields

$$\tilde{S}_{k,\, MAP} = \arg\min_{x\in\mathcal{X}} \left\{ c_{\mathrm{Re}}(R,\, \tilde{I},\, x) + c_{\mathrm{Im}}(R,\, \tilde{I},\, x) \right\} \tag{25}$$

with $c(\cdot)$ given by (24). Note that computing the MAP estimate is much simpler than computing the MMSE estimate.

G. Algorithm

Based on the previous sections, the main idea of the decoding algorithm can now be explained. The approximation of the inputs to the estimators as i.i.d. random variables, see Approximation II.1, is the basis for the design of low-complexity estimators. However, the estimators are suboptimum: the codeword domain estimator neglects the dependencies among the code symbols s_k, while treating the i.i.d. impulsive noise symbols i_k correctly. Conversely, the information domain estimator neglects the dependence among the impulsive noise samples I_k, while treating the information symbols S_k correctly. The idea is that by alternately using both estimators, at least to some extend, the whole statistical information provided by the received vector r about the transmitted codeword is exploited in the decoding process. Approximation II.1 plays the central role in the derivation of the decoding algorithm.

```
Initialization
  set $l = 0,\ \beta_s^{(0)} = \beta_i^{(0)} = \beta_g^{(0)} := 0$
  set $\alpha_s^{(0)} = \alpha_i^{(0)} = \alpha_g^{(0)} := 0$
  calculate $R = \mathbf{G}^{-1}r.$
while $(l <= \eta_{max})$
  l=l+1
  MMSE Codeword Domain Estimation
    $\forall k : \tilde{i}_k^{(l)} = E\left\{i_k | r_k, \tilde{s}_k^{(l-1)}\right\}$, see eq. (19)
  inverse Transform
    $\tilde{I}^{(l)} = \mathbf{G}^{-1}\tilde{i}^{(l)}$
  if $(l < \eta_{max})$
    MMSE/MAP Information Domain Estimation
      if MMSE estimation is used
        $\forall k : \tilde{S}_k^{(l)} = E\left\{S_k | R_k, \tilde{I}_k^{(l)}\right\}$, see eq. (23)
      if MAP estimation is used
        $\forall k : \tilde{S}_k^{(l)} = \operatorname*{argmax}_{x\in\mathcal{X}} p(S_k = x | R_k, \tilde{I}_k^{(l)})$, see eq. (25)
    Transform
      $\tilde{s}^{(l)} = \mathbf{G}\tilde{S}^{(l)}$
  end if
end while
MAP Information Domain Estimation
  $\forall k : U_k = \operatorname*{argmax}_{x\in\mathcal{X}} P(S_k = x | R_k, \tilde{I}_k^{(l)})$, see eq. (25)
```

Fig. 2. The iterative decoding algorithm with decoding result U and $\eta_{\max}$ denoting the number of iterations.

This idea is similar to the idea of turbo decoding: here, instead of jointly decoding a concatenated code, two component-code decoders are used alternately. Each decoder exploits only the parity-check equations of one component code and neglects the dependencies introduced by the other. By alternately using both decoders, at least to some extend, the whole statistical information about the transmitted codeword is exploited. The argument that the dependencies of one component code can be neglected is justified by using an interleaver. In our algorithm, the unitary transform $\boldsymbol{G}$ plays the role of the interleaver, and Approximation II.1 is used to justify that only partial statistical information is used in the estimators.

The exact definition of the decoding algorithm is given in Fig. 2. The input to the decoder is the received vector r. Before using the algorithm, the coefficients $\alpha_s^{(l)}$, $\alpha_i^{(l)}$, $\alpha_g^{(l)}$, $\beta_s^{(l)}$, $\beta_i^{(l)}$, $\beta_g^{(l)}$, as well as the variances of the error terms $D^{(l)}$ and $e^{(l)}$ have to be determined for all iterations $l = 0, \ldots, \eta_{\max}$. This is done by the following procedure: based on the parameters in the initialization part of the decoding algorithm, the vector $i^{(1)}$ is evaluated for randomly selected codewords and random noise. The values of $i^{(1)}$ are then used to measure the parameters $\alpha_s^{(1)}$, $\alpha_i^{(1)}$, $\alpha_g^{(1)}$, and $\sigma_{e^{(1)}}^2$ employing least squares estimation as described in Section II-A. Given the $\alpha^{(1)}$-values, the same method is applied to determine the $\beta^{(1)}$-parameters. This procedure is repeated until the coefficients are known for all iterations.

The decoding algorithm includes a special case. Choosing $\eta_{\max} = -1$, the algorithm is equivalent to the "smearing filters" decoder described in the Introduction. Then, the U_k are determined based only on the components R_k of the vector $R = \boldsymbol{G}^{-1}r$. Due to Approximations II.1 and II.2, the R_k are approximately Gaussian distributed with variance $\sigma^2 = \sigma_i^2 + \sigma_g^2$ and mean S_k. Hence, as the codeword length n increases, the performance of this decoder approaches the performance of uncoded transmission over an AWGN channel with variance σ^2. When $\boldsymbol{G}$ is taken as the inverse Fourier matrix $\boldsymbol{F}^{-1}$, this decoder is even more equivalent to the conventional OFDM demodulator applied in an impulsive noise environment. For more detailed discussion, see [10].

Finally, the decoder's complexity is analyzed. It is mainly determined by the matrix–vector products $\tilde{I}^{(l)} = \boldsymbol{G}^{-1}\tilde{i}^{(l)}$ and $\tilde{s}^{(l)} = \boldsymbol{G}\tilde{S}^{(l)}$. For general, nonsparse generator matrices, the decoding complexity is therefore given by $(2\eta_{\max} + 3)n^2$. If a generator matrix $\boldsymbol{G}$ is applied for which fast transform algorithms such as the fast Fourier transform (FFT) with complexity order $n\log_2(n)$ are available, the decoder's complexity reduces to $(2\eta_{\max} + 3)n\log_2(n)$.

III. Simulation Results and Discussion

In this section, simulation results for the decoding symbol-error rate (SER) obtained by our algorithm are presented. The SER is defined as the probability $P(S_k \neq U_k)$ that $S_k \in \mathcal{X}$ is transmitted but $U_k \neq S_k$, $U_k \in \mathcal{X}$, is erroneously decoded. In Figs. 3–5, the SER is plotted for different numbers of iterations as a function of the signal-to-noise ratio (SNR) SNR $:= E\{|S_k|^2\}/(2\sigma^2)$, where $S_k \in \mathcal{X}$ and $\sigma^2 = \sigma_i^2 + \sigma_g^2$ holds. Additionally, the SER achieved by uncoded transmission, a lower bound denoted by "Gauss bound," and the performance of uncoded transmission over the AWGN channel with noise variance

$$\sigma^2_{\text{AWGN}} = 1/\sum_{m=0}^{\infty}(e^{-A}A^m)/(m!\sigma_m^2)$$

("AWGN uncoded") is plotted, where σ_m^2 is defined by (5). The bound "Gauss bound" is derived by assuming that only the Gaussian noise g is present on the channel, i.e., the impulsive noise part is set to zero, see [10]. The comparison with uncoded transmission over the AWGN channel with variance σ^2_{AWGN} is motivated by [9], [10]. Here, it is shown that the PEP of a CN code, i.e., the probability of a decoding error in the binary decision between any two codewords of the code, is lower-bounded by the PEP achieved on the AWGN channel with noise variance σ^2_{AWGN}.

For all simulations, 4-QAM modulation has been applied to generate the information symbols S_k, and $\boldsymbol{G} := \boldsymbol{F}^{-1}$ has been chosen, where $\boldsymbol{F}$ is the Fourier matrix. For the Walsh–Hadamard matrix (almost) the same performance was observed.

Fig. 3 depicts simulation results for $n = 1024$ employing MMSE estimation in the information domain. It can be observed that large coding gains are obtained already after only one iteration. After two iterations, the "AWGN-uncoded" curve and the lower bound are almost achieved. This basically means that the errors introduced by the impulsive noise are (almost) completely corrected. As an example, Table II shows some selected α- and β-parameters used in the simulation. The parameters have been evaluated as described in Section II-G using 10^5 vectors for the least squares estimation method. This value was found to be sufficient to obtain reliable parameter estimates for SERs greater than 10^{-5}. However, instead of using the expected value of ϕ as specified by (12), we use $\sigma^2_{d^{(l)}} = \frac{1}{2}\max\{\phi\}$, where $\sigma^2_{d^{(l)}}$ is the variance of the error term d_k, see Table I, and $\max\{\phi\}$ is the maximum average error of the linear least squares estimation method. When using the expected value of ϕ, the algorithm is not able to correct all errors in the high-SNR domain and an error floor remains. We explain this by the fact that the pdf of the error term $D_k^{(l)}$, see (15), is very spiky if the SER becomes small (i.e., $n\cdot\text{SER} \approx 1$) during the iterations. As a consequence, Approximation II.2 used to describe the pdf of $d_k^{(l)}$ becomes inaccurate. Using the maximum ϕ instead of the expected value "covers" this inaccuracy. The variance $\sigma^2_{d^{(l)}}$ might also be interpreted as a parameter that determines how much the information domain estimate influences the codeword domain estimation process.

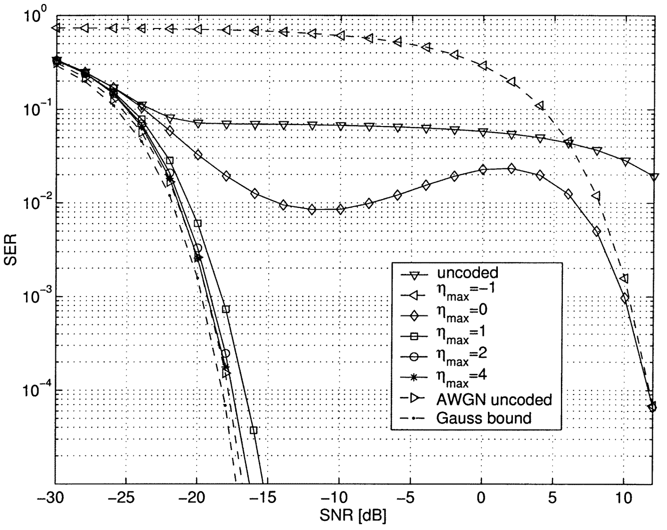

Fig. 3. Simulation results obtained with MMSE estimation, $n = 1024$, $A = 0.1$, $T = 10^{-3}$.

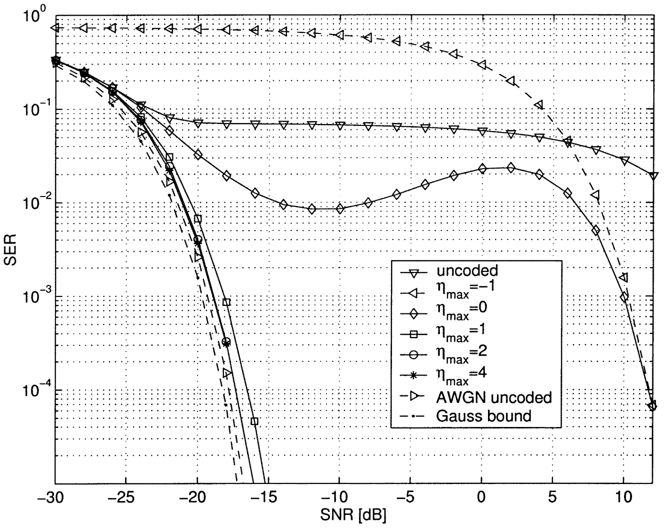

Fig. 4. Simulation results obtained with MAP estimation, $n = 1024$, $A = 0.1$, $T = 10^{-3}$.

For the result depicted in Fig. 4, we used the MAP estimator in the information domain. It can be observed that the performance is slightly worse than for the MMSE estimator. However, in practice, the MAP detector might be preferred due to its much simpler structure.

TABLE II
PARAMETERS APPLIED FOR THE SIMULATION DEPICTED IN FIG. 3 (INFORMATION DOMAIN MMSE ESTIMATION, $n = 1024, A = 0.1, T = 10^{-3}$)

SNR	l	$\alpha_s^{(l)}$ $[10^{-2}]$	$\alpha_i^{(l)}$ $[10^{0}]$	$\alpha_g^{(l)}$ $[10^{-2}]$	$\sigma^2_{e^{(l)}}$ $[10^{-2}]$	$\beta_s^{(l)}$ $[10^{-1}]$	$\beta_i^{(l)}$ $[10^{-5}]$	$\beta_g^{(l)}$ $[10^{-2}]$	$\sigma^2_{d^{(l)}}$ $[10^{-3}]$	SER
$-24dB$	1	14.414	1.000	4.048	11.090	8.423	35.602	41.288	120.41	$1.044 \cdot 10^{-1}$
$-24dB$	2	9.758	1.000	3.775	5.597	8.957	28.574	34.629	109.703	$7.836 \cdot 10^{-2}$
$-24dB$	3	7.342	1.000	3.722	4.204	9.121	26.061	31.793	107.313	$6.981 \cdot 10^{-2}$
$-24dB$	4	6.337	1.000	3.705	3.858	9.177	17.614	30.695	105.383	$6.664 \cdot 10^{-2}$
$-24dB$	5	5.914	1.000	3.707	3.775	–	–	–	–	$6.529 \cdot 10^{-2}$
$-22dB$	1	10.029	0.999	18.915	10.177	9.090	25.536	31.415	95.391	$5.887 \cdot 10^{-2}$
$-22dB$	2	4.134	1.000	18.600	3.411	9.637	5.565	18.062	56.233	$2.836 \cdot 10^{-2}$
$-22dB$	3	1.569	1.000	18.592	2.019	9.750	3.303	13.972	49.959	$2.085 \cdot 10^{-2}$
$-22dB$	4	0.924	1.000	18.589	1.794	9.778	3.195	12.835	47.685	$1.879 \cdot 10^{-2}$
$-22dB$	5	0.727	1.000	18.590	1.747	–	–	–	–	$1.826 \cdot 10^{-2}$
$-20dB$	1	10.056	0.999	9.376	9.770	9.496	28.912	23.061	83.831	$3.262 \cdot 10^{-2}$
$-20dB$	2	3.036	1.000	8.733	1.984	9.924	2.063	6.303	20.203	$6.040 \cdot 10^{-3}$
$-20dB$	3	0.749	1.000	8.667	0.976	9.963	0.268	3.719	14.346	$3.296 \cdot 10^{-3}$
$-20dB$	4	0.481	1.000	8.670	0.913	9.970	-0.136	3.206	12.938	$2.739 \cdot 10^{-3}$
$-20dB$	5	0.427	1.000	8.678	0.905	–	–	–	–	$2.563 \cdot 10^{-3}$
$-18dB$	1	9.523	0.998	10.133	9.646	9.705	26.412	16.181	76.372	$1.930 \cdot 10^{-2}$
$-18dB$	2	1.502	1.000	9.640	1.218	9.991	-7.651	1.191	15.843	$7.298 \cdot 10^{-4}$
$-18dB$	3	-0.440	1.000	9.683	0.564	9.997	-8.157	0.509	4.851	$2.464 \cdot 10^{-4}$
$-18dB$	4	-0.502	1.000	9.692	0.555	9.998	-8.222	0.428	5.156	$1.946 \cdot 10^{-4}$
$-18dB$	5	-0.513	1.000	9.695	0.554	–	–	–	–	$1.767 \cdot 10^{-4}$
$-16dB$	1	10.339	0.997	9.622	9.617	9.810	35.297	11.729	87.952	$1.255 \cdot 10^{-2}$
$-16dB$	2	1.697	1.000	8.825	0.802	10.000	-0.002	0.030	8.099	$3.710 \cdot 10^{-5}$
$-16dB$	3	0.058	1.000	8.851	0.346	–	–	–	–	$< 10^{-5}$

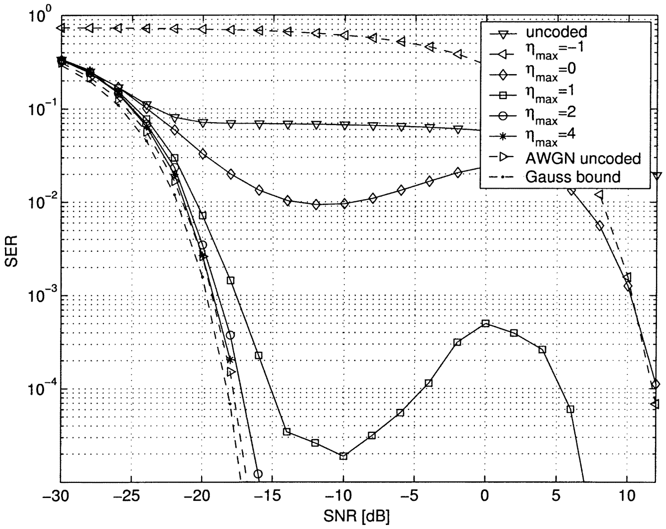

Fig. 5. Simulation results obtained with MMSE estimation, $n = 256, A = 0.1, T = 10^{-3}$.

Finally, Fig. 5 shows the SER achieved for the codeword length $n = 256$. For the first iteration, the convergence speed is significantly reduced compared to the previously discussed cases. However, after four iterations, the same performance as for $n = 1024$ is achieved.

Based on the above results and numerous other simulations employing different values for n, A, and T, the following three characteristics were identified to determine the algorithm's convergence behavior:

1) The codeword length n plays an important role for the convergence of the algorithm. Increasing n leads to a faster convergence, whereas for small n, the algorithm converges slowly or not at all. This can be expected since the derivation of the algorithm is mainly based on Approximations II.1 and II.2, and these approximations become more accurate as n increases. Typically, if the algorithm does not converge for a given parameter set A, T, and n, convergence can be achieved by sufficiently increasing n. As an example, for $A = 0.1$, $T = 10^{-3}$, and $n \geq 128$ the algorithm converges, see Fig. 3, whereas for $n \leq 64$ convergence cannot be achieved.

2) The algorithms convergence speed is almost invariant with respect to the channel parameter T.

3) It is important that at least one estimator, i.e., the codeword-domain or the information-domain estimator, evaluates reliable estimates. If this is not ensured, the algorithm's convergence behavior is poor. This statement is supported by two observations.

First, in Fig. 5 as well as for various other parameter sets, it can be observed that the algorithm converges slower around SNR $\approx$ 0 dB. Here, the noise power and, therefore, also the average height of the impulses i_k becomes smaller. Hence, the codeword domain estimator can barely distinguish the code symbols s_k from the impulsive noise i_k and therefore evaluates unreliable estimates. Additionally, the estimates in the information domain are unreliable since the SNR is still relatively small. Since both estimators give unreliable results, the convergence is slow. In other regions, this problem does not occur since here at least one of both estimators gives good estimates.

Second, it can be observed that the choice of the parameter A has a strong influence on the performance of the decoder. Decreasing A increases the convergence speed, whereas increasing A has the opposite effect. For small A, the impulsive noise becomes more structured or impulsive, and thus, the codeword and the impulsive noise can be clearly distinguished in the codeword domain. Hence, the codeword domain estimator evaluates reliable results. In contrast, when increasing A, the channel noise converges towards the Gaussian distribution. Hence, the codeword and the noise can barely be separated by the codeword-domain estimator, and the convergence speed is reduced. This effect can be observed especially in the region around SNR $\approx$ 0 dB. As an example, for $n = 1024$, $T = 10^{-3}$, and $A = 0.1$, the lower bound is almost achieved within three iterations, see Fig. 3. In contrast, for the same parameters and $A = 0.5$, within five iterations the algorithm achieves significant coding gains only for SNR $\geq$ 4 dB. For SNR $<$ 4 dB, the performance is almost similar to uncoded transmission.

IV. Conclusion

In this paper, we discuss the usage of CN codes for the transmission over an impulsive noise channel. The derivation of the optimum MAP receiver shows that, even for small n, MAP decoding is infeasible in practical applications because of the exponentially growing complexity. We therefore derive a new, suboptimum decoding algorithm applying the idea of turbo decoding. If a generator matrix $\boldsymbol{G}$ is used for which a fast transform algorithm with complexity order $n\log_2(n)$ is known, the decoder's complexity order is determined by $(2\eta_{\max} + 3)n\log_2(n)$, where $\eta_{\max}$ denotes the number of iterations.

The simulation results show that increasing the codeword length n leads to a better performance. We explain this behavior by noting that the accuracy of the approximations used to derive the algorithm increases with increasing n. As an example, for codeword lengths $n \geq 1024$ and parameters $A \leq 0.1$, $T = 10^{-3}$, the lower bound is almost reached after only two iterations. Furthermore, we conclude from the simulation results that using the MMSE estimator in the information domain leads to only slightly better results than using the MAP estimator. Of course, these results are only confirmed by simulations so we cannot make any statements about SERs below $P_{\text{SER}} < 10^{-5}$.

Since the main idea of the iterative decoding algorithm is quite general, we believe that it can be applied to various other channels by adjusting the codeword-domain estimator. As an example, if the transmission over an AWGN channel is corrupted by a memoryless amplifier nonlinearity, the codeword-domain estimator must be designed to estimate this disturbance for each received symbol. The algorithm might also be applied to the whole class of Gaussian mixture noise channels, see [10], [12]; this class includes the impulsive noise channel discussed in this paper as well as the fading channel with perfect CSI.

The close relation between CN codes and the OFDM modulation scheme is very interesting. By applying the iterative decoding scheme introduced in this paper, errors that are typical for OFDM systems, e.g., due to clipping the OFDM symbols, channel nonlinearities, and also impulsive noise, can be corrected without employing "conventional" error-correcting codes designed for the AWGN channel.

Finally, it should be noted that there are many ways to simplify the algorithm by further simplifying the information-domain and codeword-domain estimators. For example, in the information domain, an estimator with only one input $R - \tilde{I}^{(l)}$ might be used. This approach is related to decision feedback theory where an estimate of the channel noise obtained in a previous iteration is fed back and subtracted from the received vector. This significantly reduces the number of parameters needed in the linear signal model to derive the estimator. As discussed in the Introduction, such an DFE-related decoder has to use nonlinear estimators instead of the linear forward and backward matrices applied in the DFE decoder designed for the fading channel with perfect CSI, see [13], [15].

Appendix

Lemma V.1: Let $x = \{x_0, \ldots x_{n-1}\}$ be a vector of i.i.d. complex random variables with

$$E\{x_k\} = 0$$
$$E\{\text{Re}[x_k]^2\} = E\{\text{Im}[x_k]^2\} = \sigma_x^2$$

and

$$E\{\text{Re}[x_k]\text{Im}[x_k]\} = 0.$$

We calculate the vectors $y = \boldsymbol{G}x = \{y_0, y_1, \ldots\}$ and $y = \boldsymbol{G}^{-1}x$, respectively. For $n \to \infty$, the y_k are i.i.d. complex Gaussian random variables with $E\{y_k\} = 0$, $E\{\text{Re}[y_k]^2\} = E\{\text{Im}[y_k]^2\} = \sigma_x^2$, and $E\{\text{Re}[x_k]\text{Im}[x_k]\} = 0$.

Proof of Lemma V.1: The proof of this lemma splits into two parts. In the first step, the central limit theorem is used to show that $\text{Re}[y_k]$ and $\text{Im}[y_k]$ approach a Gaussian distribution. Using that $\boldsymbol{G}$ is a unitary matrix it is shown in the second step by straightforward calculation that

$$E\{y_k\} = 0$$
$$E\{\text{Re}[y_k]^2\} = E\{\text{Im}[y_k]^2\} = \sigma_x^2$$
$$E\{\text{Re}[y_k]\text{Im}[y_l]\} = 0$$

and

$$E\{\text{Re}[y_k]\text{Re}[y_l]\} = E\{\text{Im}[y_k]\text{Im}[y_l]\} = \sigma_x^2\delta_{k,\,l}$$

From the two steps the statement of the lemma follows, see [10].

References

[1] G. Beenker, T. Claasen, and P. van Gerwen, "Design of smearing filters for data transmission systems," *IEEE Trans. Commun.*, vol. COM–33, Sept. 1985.

[2] E. Biglieri, J. Proakis, and S. Shamai (Shitz), "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2619–2692, Oct. 1998.

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

[4] K. Boull and J.-C. Belfiore, "Modulation scheme designed for the Rayleigh fading channel," in *Proc. CISS 1992*, Princeton, NJ, Mar. 1992, pp. 288–293.

[5] J. Boutros and E. Viterbo, "Signal space diversity: A power- and bandwidth-efficient diversity technique for the Rayleigh fading channel," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1453–1467, July 1998.

[6] J. S. Engel, "Digital transmission in the presence of impulsive noise," *Bell Syst. Tech. J.*, pp. 1699–1743, Oct. 1965.

[7] J. Häring and A. J. Han Vinck, "OFDM transmission corrupted by impulsive noise," in *Proc. Int. Symp. Powerline Communications (ISPLC)*, Limerick, Ireland, Apr. 2000, pp. 9–14.

[8] ——, "Performance bounds for optimum and suboptimum reception under class A impulsive noise," *IEEE Trans. Commun.*, vol. 50, pp. 1130–1136, July 2002.

[9] J. Häring and A. J. Han Vinck, "Coding and signal space diversity for fading and impulsive noise channels with side information," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Lausanne, Swizerland, June 2002, p. 470.

[10] J. Häring, *Error Tolerant Communication Over the Compound Channel*. Aachen, Germany: Shaker-Verlag, 2002.

[11] C. W. Helstrom, *Probability and Stochastic Processes for Engineers*. New York: Macmillan, 1991.

[12] S. A. Kassam, *Signal Detection in Non-Gaussian Noise*. Berlin, Germany: Springer-Verlag, 1988.

[13] C. Lamy and J. Boutros, “On random rotations diversity and minimum MSE decoding of lattices,” *IEEE Trans. Inform. Theory*, vol. 46, pp. 1584–1589, July 2000.

[14] G. R. Lang, “Rotational transformation of signals,” *IEEE Trans. Inform. Theory*, vol. IT–9, pp. 191–198, July 1963.

[15] M. Reinhard and J. Lindner, “Transformation of a Rayleigh fading channel into a set of parallel AWGN channels and its advantage for coded transmission,” *Electron. Lett.*, vol. 31, pp. 2154–2155, Dec. 1995.

[16] A. D. Spaulding and D. Middleton, “Optimum reception in an impulsive interference environment—Part I: Coherent detection,” *IEEE Trans. Commun.*, vol. COM–25, pp. 910–923, Sept. 1977.

[17] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1968.

[18] E. Viterbo and J. Boutros, “An universal lattice code decoder for fading channels,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 1639–1642, July 1999.

[19] G. W. Wornell, “Spread-response precoding for communication over fading channels,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 448–501, Mar. 1996.

[20] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: Wiley, 1965.