Wireless Pers Commun
DOI 10.1007/s11277-007-9389-6

Coding Schemes for Crisscross Error Patterns

Simon Plass · Gerd Richter · A. J. Han Vinck

Abstract This paper addresses two coding schemes which can handle emerging errors with crisscross patterns. First, a code with maximum rank distance, so-called Rank-Codes, is described and a modified Berlekamp–Massey algorithm is provided. Secondly, a Permutation Code based coding scheme for crisscross error patterns is presented. The influence of different types of noise are also discussed.

Keywords Rank Codes · Permutation Code · Crisscross errors

1 Introduction

In a number of applications, the following error protection problem occurs: The information symbols have to be stored in $(N \times n)$ arrays. Some of these symbols are transmitted erroneously in such a way that all corrupted symbols are confined to a specified number of rows or columns (or both). We refer to such errors as crisscross errors. These crisscross errors can be found for example in memory chip arrays [1] or in magnetic tape recording [2]. Figure 1 shows a crisscross error pattern that is limited to two columns and three rows.

Since the Hamming metric is not appropriate for these error patterns, Delsarte [3] introduced the rank of a matrix as a metric for error correction purpose. Gabidulin [4] and also Roth [5] introduced codes with maximum rank distance (Rank-Codes) that are capable of

S. Plass (✉)
Institute of Communications and Navigation, German Aerospace Center (DLR), 82234 Wessling, Germany
e-mail: simon.plass@dlr.de

G. Richter
Department of TAIT, Ulm University, 89081 Ulm, Germany
e-mail: gerd.richter@uni-ulm.de

A. J. Han Vinck
Institute for Experimental Mathematics, University of Essen, 45326 Essen, Germany
e-mail: vinck@iem.uni-due.de

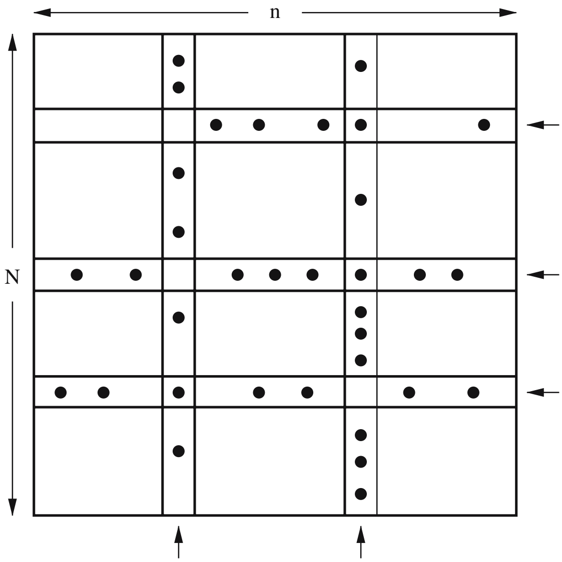

Fig. 1 Crisscross error pattern

correcting a specified number of corrupted rows and columns. Rank-Codes cannot only correct erroneous rows and columns, they can even correct a certain number of rank errors. The number of rank errors is defined as the rank of the error array.

Furthermore, it is also possible to define a Permutation Code in which each codeword contains different integers as symbols. This code can be applied to the crisscross error problem. Other applications are also given in this article.

This article will describe the Rank-Codes and will introduce a modified Berlekamp–Massey algorithm for Rank-Codes as an efficient decoding procedure for decoding rank errors. A Permutation Code for crisscross patterns, its applications, and the effects of different types of noise are also addressed.

2 Rank-Codes

In this section, we describe some fundamentals of Rank-Codes that were introduced by Gabidulin in 1985 [4]. Later, a decoding scheme based on a modified Berlekamp–Massey algorithm is introduced.

2.1 Fundamentals of Rank-Codes

Let $\mathbf{x}$ be a codeword of length n with elements from $GF(q^N)$, where q is a power of a prime. Let us consider a bijective mapping

$$\mathcal{A} : GF(q^N)^n \to \mathbf{A}_N^n,$$

which maps the codeword $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ to an $(N \times n)$ array. In the following, we consider only codewords of length $n \leq N$.

Definition 1 (*Rank Metric over $GF(q)$*) The rank of $\mathbf{x}$ over q is defined as $r(\mathbf{x}|q) = r(\mathbf{A}|q)$. The rank function $r(\mathbf{A}|q)$ is equal to the maximum number of linearly independent rows or columns of $\mathbf{A}$ over $GF(q)$.

It is well known that the rank function defines a norm. Indeed, $r(\mathbf{x}|q) \geq 0, \quad r(\mathbf{x}|q) = 0 \Longleftrightarrow \mathbf{x} = 0$. In addition, $r(\mathbf{x}+\mathbf{y}|q) \leq r(\mathbf{x}|q) + r(\mathbf{y}|q)$. Furthermore, $r(a\mathbf{x}|q) = |a|r(\mathbf{x}|q)$ is also fulfilled, if we set $|a| = 0$ for $a = 0$ and $|a| = 1$ for $a \neq 0$.

Definition 2 (*Rank Distance*) Let $\mathbf{x}$ and $\mathbf{y}$ be two codewords of length n with elements from $GF(q^N)$. The rank distance is defined as $\text{dist}_r(\mathbf{x}, \mathbf{y}) = r(\mathbf{x} - \mathbf{y}|q)$.

Similar to the minimum Hamming distance, we can determine the minimum rank distance of a code $\mathcal{C}$.

Definition 3 (*Minimum rank distance*) For a code $\mathcal{C}$ the minimum rank distance is given by

$$d_r := \min\{\text{dist}_r(\mathbf{x}, \mathbf{y})|\mathbf{x} \in \mathcal{C}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}\},$$

or when the code is linear by

$$d_r := \min\{r(\mathbf{x}|q)|\mathbf{x} \in \mathcal{C}, \mathbf{x} \neq 0\}.$$

Let $\mathcal{C}(n, k, d_r)$ be a code of dimension k, length n, and minimum rank distance d_r. It is shown in [4] that there also exists a Singleton-style bound for the rank distance. Theorem 1 shows, how the minimum rank distance d_r is bounded by the minimum Hamming distance d_h and by the Singleton bound.

Theorem 1 (Singleton-style bound) *For every linear code $\mathcal{C}(n, k, d_r) \subset GF(q^N)^n$ d_r is upper bounded by*

$$d_r \leq d_h \leq n - k + 1.$$

Definition 4 (*MRD code*) A linear (n, k, d_r) code $\mathcal{C}$ is called maximum rank distance (MRD) code, if the Singleton-style bound is fulfilled with equality.

In [4] and in [5], a construction method for the parity-check matrix and the generator matrix of an MRD code is given as follows:

Theorem 2 (Construction of MRD codes) *A parity-check matrix $\mathbf{H}$, which defines an MRD code and the corresponding generator matrix $\mathbf{G}$ are given by*

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_0^q & h_1^q & \cdots & h_{n-1}^q \\ h_0^{q^2} & h_1^{q^2} & \cdots & h_{n-1}^{q^2} \\ \vdots & \vdots & \ddots & \vdots \\ h_0^{q^{d-2}} & h_1^{q^{d-2}} & \cdots & h_{n-1}^{q^{d-2}} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-1} \\ g_0^q & g_1^q & \cdots & g_{n-1}^q \\ g_0^{q^2} & g_1^{q^2} & \cdots & g_{n-1}^{q^2} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{q^{k-1}} & g_1^{q^{k-1}} & \cdots & g_{n-1}^{q^{k-1}} \end{bmatrix},$$

where the elements $h_0, h_1, \ldots, h_{n-1} \in GF(q^N)$ and $g_0, g_1, \ldots, g_{n-1} \in GF(q^N)$ are linearly independent over $GF(q)$.

In the following, we define $\mathcal{C}_{\mathcal{MRD}}(n, k, d_r)$ as an MRD code of length n, dimension k, and minimum rank distance $d_r = n - k + 1$. The decoding of Rank-Codes with the modified Berlekamp–Massey algorithm can be done based on linearized polynomials.

Definition 5 (*Linearized polynomials*) A linearized polynomial over $GF(q^N)$ is a polynomial of the form

$$L(x) = \sum_{p=0}^{N(L)} L_p x^{q^p},$$

where $L_p \in GF(q^N)$ and $N(L)$ is the *norm* of the linearized polynomial. The norm $N(L)$ characterizes the largest p, where $L_p \neq 0$. Let $\otimes$ be the symbolic product of linearized polynomials defined as

$$F(x) \otimes G(x) = F(G(x)) = \sum_{p=0}^{j} \sum_{i+l=p} \left(f_i g_l^{q^i} \right) x^{q^p},$$

where $0 \leq i \leq N(F)$, $0 \leq l \leq N(G)$, and $j = N(F) + N(G)$.

It is known that the symbolic product is associative and distributive, but it is non-commutative.

2.2 Decoding of Rank-Codes

There exist different algorithms for the decoding of Rank-Codes. Gabidulin [4] introduced the decoding with Euclid's Division algorithm based on linearized polynomials. In 1991, Roth described another decoding algorithm [5] that is similar to the Peterson–Gorenstein–Zierler algorithm for Reed–Solomon codes. Recently, a Welch–Berlekamp like decoding algorithm was introduced by Loidreau [6].

In 1968, Berlekamp introduced a very efficient technique for the decoding of Reed–Solomon codes. One year later, Massey [7] interpreted this algorithm as a problem of synthesizing the shortest linear feedback shift-register capable of generating a prescribed finite sequence of digits. Since the structure of Reed–Solomon codes is quite similar to the structure of Rank–Codes, another possible decoding method for Rank-Codes is a modified Berlekamp–Massey algorithm, which is introduced in this section.

Let **c**, **r**, and **e** be the codeword vector, the received vector, and the error vector of length n with elements from $GF(q^N)$, respectively. The received vector is $\mathbf{r} = \mathbf{c} + \mathbf{e}$. Let $v = r(\mathbf{e}|q)$ be the rank of the error vector **e**. Now we present a method of finding the correct codeword, if $2 \cdot v < d_r$. We can calculate the syndrome $\mathbf{s} = (S_0, S_1, \ldots S_{d_r-2})$ by

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T. \tag{1}$$

Let us define a $(v \times n)$ matrix $\mathbf{Y}$ of rank v, whose entries are from the base field $GF(q)$. Thus, we can write

$$\mathbf{e} = (E_0, E_1, \ldots, E_{v-1})\mathbf{Y}, \tag{2}$$

where $E_0, E_1, \ldots, E_{v-1} \in GF(q^N)$ are linearly independent over $GF(q)$. Let the matrix $\mathbf{Z}$ be defined as

$$\mathbf{Z}^T = \mathbf{Y}\mathbf{H}^T = \begin{bmatrix} z_0 & z_0^q & \cdots & z_0^{q^{d-2}} \\ z_1 & z_1^q & \cdots & z_1^{q^{d-2}} \\ \vdots & \vdots & \ddots & \vdots \\ z_{v-1} & z_{v-1}^q & \cdots & z_{v-1}^{q^{d-2}} \end{bmatrix}. \tag{3}$$

It can be shown that the elements $z_0, z_1, \ldots, z_{v-1} \in GF(q^N)$ are linearly independent over $GF(q)$. Hence, (1) can be written as

$$(S_0, S_1, \ldots, S_{d_r-2}) = (E_0, E_1, \ldots, E_{v-1}) \cdot \mathbf{Z}^T ,$$

$$S_p = \sum_{j=0}^{v-1} E_j z_j^{q^p} \;, \;\; p = 0, \ldots, d_r - 2. \tag{4}$$

By raising each side of (4) to the power of q^{-p} we get

$$S_p^{q^{-p}} = \sum_{j=0}^{v-1} E_j^{q^{-p}} z_j \;, \;\; p = 0, \ldots, d_r - 2. \tag{5}$$

Hence, we have a system of $d_r - 1$ equations with $2 \cdot v$ unknown variables that are linear in $z_0, z_1, \ldots, z_{v-1}$. Note that also the rank v of the error vector is unknown. It is sufficient to find one solution of the system because every solution of $E_0, E_1, \ldots, E_{v-1}$ and $z_0, z_1, \ldots, z_{v-1}$ results in the same error vector $\mathbf{e}$.

Let $\Lambda(x) = \sum_{j=0}^{v} \Lambda_j x^{q^j}$ be a linearized polynomial, which has all linear combinations of $E_0, E_1, \ldots, E_{v-1}$ over $GF(q)$ as its roots and $\Lambda_0 = 1$. We call $\Lambda(x)$ the row error polynomial. Also, let $S(x) = \sum_{j=0}^{d-2} S_j x^{q^j}$ be the linearized syndrome polynomial.

Now it is possible to define the key equation by

Theorem 3 (Key equation)

$$\Lambda(x) \otimes S(x) = F(x) \bmod x^{q^{d_r-1}}, \tag{6}$$

where $F(x)$ is an auxiliary linearized polynomial with norm $N(F) < v$.

Hence, we have to solve the following system of equations to get $\Lambda(x)$, if $2 \cdot v < d_r$:

$$\sum_{i=0}^{p} \Lambda_i S_{p-i}^{q^i} = 0 \;, \;\; p = v, \ldots, 2v - 1.$$

We subtract $S_p \Lambda_0$ on both sides and obtain

$$-S_p = \sum_{i=1}^{v} \Lambda_i S_{p-i}^{q^i} \;, \;\; p = v, \ldots, 2v - 1.$$

because $\Lambda_0 = 1$ and $\Lambda_i = 0$ for $i > v$. This can be written in matrix form as

$$\mathbf{S} \begin{bmatrix} \Lambda_v \\ \Lambda_{v-1} \\ \Lambda_{v-2} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_v \\ -S_{v+1} \\ -S_{v+2} \\ \vdots \\ -S_{2v-1} \end{bmatrix}, \quad \text{with} \quad \mathbf{S} = \begin{bmatrix} S_0^{q^v} & \cdots & S_{v-1}^{q^1} \\ S_1^{q^v} & \cdots & S_v^{q^1} \\ S_2^{q^v} & \cdots & S_{v+1}^{q^1} \\ \vdots & \ddots & \vdots \\ S_{v-1}^{q^v} & \cdots & S_{2v-2}^{q^1} \end{bmatrix}. \tag{7}$$

It can be shown that the matrix $\mathbf{S}$ is non-singular. Thus, the system of equations has a unique solution. This solution can be efficiently found with a modified Berlekamp–Massey algorithm. The description of the modified Berlekamp–Massey algorithm is inspired by [8]. We can see (7) also as a feedback shift-register with tap weights given by $\Lambda(x)$. This is shown in Fig. 2. The symbols $f_1, f_2, \ldots, f_v$ stand for the powers of $q^1, q^2, \ldots, q^v$ (cf. (7)).

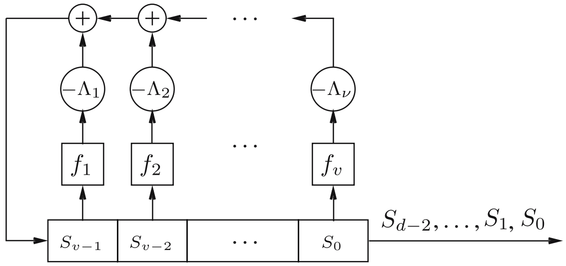

Fig. 2 Row error polynomial as a shift-register

The problem of solving the key equation is equivalent to a problem of finding the shortest feedback shift-register that generates the known sequence of syndromes. The design procedure is inductive. We start with iteration $r = 0$ and initialize the length of the shift-register $L_0 = 0$ and $\Lambda(x) = x$. For each iteration r we create a feedback shift-register that generates the first $r + 1$ syndromes and that has minimum length L_{r+1}. Hence, at the start of iteration r we have a shift-register given by $\Lambda^{(r)}(x)$ of length L_r. The notation of the exponent in brackets declares the iteration. To find $\Lambda^{(r+1)}(x)$ we determine the discrepancy of the output of the r-th shift-register and S_r by

$$\Delta_r = S_r + \sum_{j=1}^{L_r} \Lambda_j^{(r)} S_{r-j}^{q^j} = \sum_{j=0}^{L_r} \Lambda_j^{(r)} S_{r-j}^{q^j}. \tag{8}$$

For the case $\Delta_r = 0$, we set $\Lambda^{(r+1)}(x) = \Lambda^{(r)}(x)$ and the iteration is complete. On the other hand, if $\Delta_r \neq 0$, the shift-register taps have to be modified in the following way:

Theorem 4 (Shift-register modification) *The linearized polynomial $\Lambda^{(r+1)}(x)$ is given by*

$$\Lambda^{(r+1)}(x) = \Lambda^{(r)}(x) + Ax^{q^l} \otimes \Lambda^{(m)}(x), \tag{9}$$

where $m < r$. Thus, if we choose $l = r - m$ and $A = -\Delta_r \Delta_m^{-q^l}$, the new discrepancy $\Delta_r' = 0$.

Proof 1 From (8) it follows that

$$\Delta_r' = \sum_{j=0}^{L_{r+1}} \Lambda_j^{(r+1)} S_{r-j}^{q^j} \overset{(9)}{=} \sum_{i=0}^{L_r} \Lambda_i^{(r)} S_{r-i}^{q^i} + A \sum_{i=0}^{L_m} \left(\Lambda_i^{(m)} S_{r-i-l}^{q^i} \right)^{q^l}$$
$$= \Delta_r + A \cdot \Delta_m^{q^l} = \Delta_r - \Delta_r \Delta_m^{-q^l} \cdot \Delta_m^{q^l} = 0,$$

where the syndrome **s** in the second sum has to be shifted for l positions because of the symbolic product with x^{q^l}. □

The new shift-register denoted by $\Lambda^{(r+1)}(x)$ has either length $L_{r+1} = L_r$ or $L_{r+1} = l + L_m$. It can be shown that we get a shortest shift-register for every iteration, if we choose m as the most recent iteration, at which the shift-register length L_{m+1} has been increased. It was proved in [8] that the shortest feedback shift-register for Reed-Solomon codes in iteration r has length $L_{r+1} = \max\{L_r, r + 1 - L_r\}$. Furthermore, it is proved that the Berlekamp–Massey algorithm generates a shortest feedback shift-register in each iteration (see, e.g., [7] or [9]).

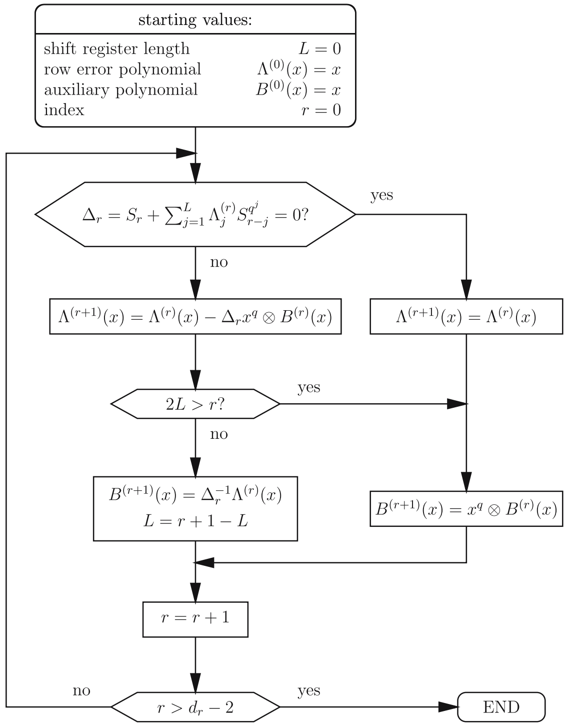

Fig. 3 Berlekamp–Massey algorithm for rank errors

Thus, $\Lambda^{(r+1)}(x)$ generates the first $r + 1$ syndromes. The shift-register of iteration m produces zeros at the first $m - 1$ iterations because there is an additional tap with weight one. At iteration m the shift-register produces $\Delta_m^{q^l}$, which is multiplicated by $A = -\Delta_r \Delta_m^{-q^l}$. This compensates Δ_r that was produced by the shift-register of iteration r. Hence, the new shift-register generates the sequence $S_0, S_1, \ldots, S_r$.

The modified Berlekamp–Massey algorithm for Rank-Codes is summarized as a flowchart in Fig. 3. $B(x)$ is an auxiliary linearized polynomial that is used to store $\Lambda^{(m)}(x)$, the row error polynomial of iteration m.

We can summarize the different steps of the decoding procedure.

1. Calculate the syndrome with (1).
2. Solve the key equation (7) with the modified Berlekamp–Massey algorithm to obtain $\Lambda(x)$.
3. Calculate the linearly independent roots $E_0, E_1, \ldots, E_{v-1}$ of $\Lambda(x)$. This can be done with the algorithm described in [10].
4. Solve the linear system of equations (5) for the unknown variables $z_0, z_1, \ldots, z_{v-1}$.
5. Calculate the matrix $\mathbf{Y}$ using (3).
6. Calculate $\mathbf{e}$ by (2) and the decoded codeword $\hat{\mathbf{c}} = \mathbf{r} - \mathbf{e}$.

3 Permutation Codes

This section describes a class of codes used for the correction of crisscross errors in a matrix of dimension $(N \times n), n \leq N$, where the entries are binary. We also show that the combination of coding and M-FSK (frequency shift keying) modulation can be seen as a kind of matrix transmission. Particular types of noise, like impulsive- or narrowband noise occur as column or row errors. Permutation Codes are shown to be very well suited for the correction of these types of errors. We first give a definition of Permutation Codes.

Definition 6 (*Permutation Code*) A Permutation Code $\mathcal{C}$ consists of $|\mathcal{C}|$ codewords of length N, where every codeword contains the N different integers $1, 2, \ldots, N$ as symbols.

For a Permutation Code of length N with N different code symbols in every code word and minimum Hamming distance $d_{\min}$, the cardinality is upper bounded by

$$|\mathcal{C}| \leq \frac{N!}{(d_{\min} - 1)!} \tag{10}$$

For specific values of N, we have equality in (10). For instance for $d_{\min} = N - 1$, N is a prime, and therefore, $|\mathcal{C}| = N(N - 1)$. As an example, for $N = 3$ and $d_{\min} = 2$, we have six codewords, $C = 123, 231, 312, 213, 321, 132$.

We represent codewords in a binary matrix of dimension $N \times N$, where every row and every column contains exactly one single symbol 1. A symbol 1 occurs in row i and column j if a codeword symbol has the value i at position j. If the dimensions of the array are $N \times n$, we simply shorten the code and also reduce the minimum distance with the equivalent amount. Since Permutation Codes are now defined over a binary matrix, we can use them also to correct crisscross errors.

3.1 Crisscross and Random Errors

We assume that a row (or column) error changes the entries in a particular row from 0 to 1 or from 1 to 0. A row (or column) error reduces the distance between any two codewords by a maximum value of two. The reason for this is, that a row or column error can agree with a codeword only in one position. The same argument can be used for random errors. A random error reduces the distance only by one. Hence, we can correct these errors if

$$d_{\min} > 2(t_{\text{row}} + t_{\text{column}}) + t_{\text{random}}, \tag{11}$$

where t_{row}, t_{column}, and t_{random} are the number of row, column, and random errors.

3.2 M-FSK

The combination of Permutation Codes and M-FSK modulation can be used to correct narrowband- and impulsive noise, when these errors are considered as crisscross error patterns. In an M-FSK modulation scheme, symbols are modulated as one of M orthogonal sinusoidal waves. For notational consistence, we set $M = N$ and we use the integers $1, 2, \ldots, N$ to represent the N frequencies, i.e., the integer i represents frequency f_i. The symbols of a codeword are transmitted in time as the corresponding frequencies.

Example 1 $N = 4$, $|C| = 4$. $C = \{(1, 2, 3, 4),(2, 1, 4, 3),(3, 4, 1, 2),(4, 3, 2, 1)\}$. As an example, message three is transmitted in time as the series of frequencies (f_3, f_4, f_1, f_2). Note that the code $\mathcal{C}$ has four codewords with the property that two codewords always differ in four positions.

For Additive White Gaussian Noise channels, the "classical" coherent or non-coherent demodulator detects N envelopes and it gives as an estimate the frequency with the largest envelope.

No noise	Background noise	Background noise	narrowband	impulsive	fading
1 0 0 0	1 0 *1* 0	1 0 0 0	*1 1 1 1*	1 0 0 *1*	1 0 0 0
0 1 0 0	0 1 0 0	0 *0* 0 0	0 1 0 0	0 1 0 *1*	*0 0 0 0*
0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 *1*	0 0 1 0
0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 *1*	0 0 0 1

Fig. 4 The effect of several types of noise

The performance of an optimal demodulator for coherent or non-coherent detection will be severely degraded by the occurrence of impulsive- or narrow band noise. Frequency disturbances like *narrowband noise* are permanent over a long period of time. On the other hand, *impulsive noise* may put energy in larger parts of the spectrum in use for a short period. Impulsive noise can also have a periodic character. To overcome these problems, we modify the demodulator output. The modified demodulator uses a threshold T for every envelope detector. For values above the threshold we output $a = 1$, otherwise $a = 0$. Instead of a single demodulator output we then have N binary outputs per transmitted symbol. The outputs of the demodulator are put in a binary $(N \times N)$ matrix.

The effect of the different kinds of noise on the detector outputs can be seen from Fig. 4. We assume that $N = 4$ and transmit the codeword (1, 2, 3, 4) as the sequence of frequencies (f_1, f_2, f_3, f_4). We output the message corresponding to the codeword that has the maximum number of agreements with the demodulator outputs. Several channel disturbances can be considered:

- Narrow band noise may cause large detector outputs and thus may set the entries of a particular row equal to 1;
- Impulsive noise has a broadband character and thus may set the entries in a particular column equal to 1;
- Absence of a frequency, caused by for instance fading, may cause the entries of a row of the decoding matrix set to be equal to 0;
- Background noise introduces incorrect detector outputs, i.e., insertions or deletions of a symbol 1.

Note that the above types of row and column errors are assumed to be uni-directional, i.e., the particular entries change into either 0 or 1 direction.

All types of disturbances interfere with a particular permutation codeword in only one position. Narrow band-, impulsive-, and background noise may together generate a possible codeword in the decoding matrix. For this to occur, we need $d_{\min}$ error events. A row with entries set equal to 0 reduces the number of agreements for the correct codeword with one. Hence, if less than $d_{\min}$ disturbances of the above type occur, the decoder using minimum distance decoding can still find the transmitted (correct) codeword. It is easy to see that the modified demodulation in combination with the Permutation Code then allows the correction of the error events caused by narrow band-, impulsive-, background noise and fading if

$$d_{\min} > t_{\text{row}} + t_{\text{column}} + t_{\text{random}}. \tag{12}$$

Remarks As a practical value for the threshold T we suggest to average the values of the noiseless output of a detector. We can optimize T for additive white Gaussian noise (AWGN), but our main concern is non-AWGN. “Soft” outputs can only be used if the probability distributions of these types of noise are known.

Decoding of the Permutation Codes can be done by using regular minimum distance decoding. However, for codes with large cardinality, this can be a complex operation, since a received matrix has to be compared with all codewords in the codebook. As an alternative approach, we developed the concept of permutation trellis codes [11], where the Viterbi algorithm is used to do the decoding with low complexity.

4 Conclusions

We presented two coding schemes which can handle so-called crisscross error patterns. Rank-Codes were described and a modified Berlekamp–Massey algorithm for this coding scheme was introduced. Further, a presented permutation based coding scheme can also scope with crisscross errors.

Acknowledgement This work has been performed in the framework of the COST 289 action, which is funded by the European Union.

References

1. Levine, L., & Meyers, W. (1976). Semiconductor memory reliability with error detecting and correcting codes. *Computers, 9*, 43–50.
2. Patel, A. M., & Hong, S. J. (1974). Optimal rectangular code for high density magnetic tapes. *IBM J. Res. Dev., 18*, 579–588.
3. Delsarte, P. (1978). Bilinear forms over a finite field with applications to coding theory. *Journal of combinatorial theory. Series A, 25*(4), 226–241.
4. Gabidulin, E. M. (1985). Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii, 21*(1), 3–16.
5. Roth, R. M. (1991). Maximum-rank array codes and their application to crisscross error correction. *IEEE Transactions Information on Theory, 37*(2), 328–336.
6. Loidreau, P. (2005). A Welch-Berlekamp like algorithm for decoding Gabidulin codes. In *4th International Workshop on Coding and Cryptography*, Bergen, Swiss, March 2005.
7. Massey, J. L. (1969). Shift-register synthesis and BCH decoding. *IEEE Transactions Information on Theory, IT-15*, 122–127.
8. Blahut, R. E. (1983). *Theory and practice of error control codes*. New York: Addison Wesley, Owego, 13827, 1983. ISBN 0-201-10102-5.
9. Imamura, K., & Yoshida, W. (1987). A simple derivation of the Berlekamp–Massey algorithm and some applications. *IEEE Transactions Information on Theory, IT-33*, 146–150.
10. Berlekamp, E. R. (1968). *Algebraic coding theory*. McGraw Hill, New York.
11. Ferreira, H., Han Vinck, A. J., Swart, Th., & de Beer, I. (2005). Permutation trellis codes. *IEEE Transactions Communication, 53*(11), 1782–1789.

Author Biographies

Simon Plass is with the Institute of Communications and Navigation at the German Aerospace Center (DLR), Oberpaffenhofen, Germany since 2003. His current interests are wireless communication systems with special emphasis on cellular multi-carrier spread spectrum systems. He is co-editor of *Multi-Carrier Spread Spectrum 2007* (Springer, 2007).

Gerd Richter received his Dipl.-Ing. degree in Electrical Engineering from the University of Ulm, Germany, in 2002. Since 2002 he has been a research assistant in the Institute of Telecommunications and Applied Information Theory at Ulm University. His research interests are low-density parity-check codes and rank codes.

A. J. Han Vinck is a full professor in Digital Communications at the University of Duisburg-Essen, Germany, since 1990. In 2003 he was elected president of the IEEE Information theory Society. IEEE elected him in 2006 as a fellow for his *"Contributions to Coding Techniques"*.