Coding Schemes for Crisscross Error Patterns

Simon Plass
German Aerospace Center (DLR)
Institute of Communications and Navigation
82234 Wessling, Germany
simon.plass@dlr.de

Gerd Richter
Ulm University
Department of TAIT
89081 Ulm, Germany
gerd.richter@uni-ulm.de

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

***Abstract*— This paper addresses two coding schemes which can handle emerging errors with crisscross patterns. For example, these error patterns can occur in time-stationary fading channel conditions of a next generation multi-carrier based transmission system. 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 which can be decoded by the concepts of permutation trellis codes.**

I. 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. Fig. 1 shows a crisscross error pattern that is limited to two columns and three rows.

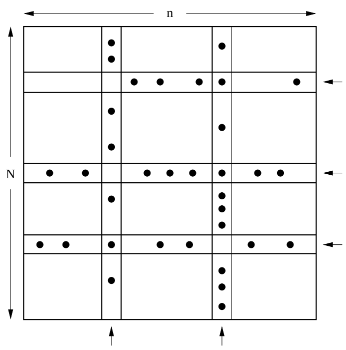

Fig. 1. Crisscross error pattern

In next generation communications systems based on multi-carrier transmission schemes the symbols are transmitted in a frame structure which can be presented in a matrix from [1]. In hot spot scenarios the channel generates error patterns which are mainly limited to several sub-carriers due to time-stationarity of the channel. Solitary errors in a transmitted frame can be recovered by an inner coding scheme, e.g., a Reed-Solomon code, and therefore, the addressed coding schemes in this paper can be applied for the redundant crisscross errors as an outer code in these scenarios.

Since the Hamming metric is not appropriate for these error patterns, Delsarte [2] introduced the rank of a matrix as a metric for error correction purpose. Gabidulin [3] and also Roth [4] introduced codes with maximum rank distance (Rank-Codes) that are capable of 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 paper. The concepts of permutation trellis codes allow an efficient decoding scheme for these permutation codes.

This paper 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 its decoding is also addressed.

II. Rank-Codes

In this section, we describe some fundamentals of Rank-Codes that were introduced by Gabidulin in 1985 [3]. Later, a decoding scheme based on a modified Berlekamp-Massey algorithm is introcuded

A. 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

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

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, $r(\mathbf{x}|q) \geq 0, \quad r(\mathbf{x}|q) = 0 \iff \mathbf{x} = 0$. 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 [3] 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 [3] and in [4], 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 is given by

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_0^q & h_1^q & \cdots & h_{n-1}^q \\ h_0^{q^2} & h_1^{q^2} & \cdots & h_{n-1}^{q^2} \\ \vdots & \vdots & \ddots & \vdots \\ h_0^{q^{d-2}} & h_1^{q^{d-2}} & \cdots & h_{n-1}^{q^{d-2}} \end{bmatrix}$$

and the corresponding generator matrix can be written as

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

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.

B. Decoding of Rank-Codes

There exist different algorithms for the decoding of Rank-Codes. Gabidulin [3] introduced the decoding with Euclid's Division algorithm based on linearized polynomials. In 1991, Roth described another decoding algorithm [4] that is similar to the Peterson-Gorenstein-Zierler algorithm for Reed-Solomon codes.

In 1968, Berlekamp introduced a very efficient technique for the decoding of Reed-Solomon codes. One year later, Massey [5] 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 $\mathbf{c}$, $\mathbf{r}$, and $\mathbf{e}$ 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 $\mathbf{e}$. 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

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

or

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

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 that has norm $N(F) < v$.

Proof: From the definition of linearized polynomials we know that

$$\Lambda(x) \otimes S(x) = \sum_{p=0}^{v+d_r-2} \left(\sum_{i+l=p} \Lambda_i S_l^{q^i} \right) x^{q^p}.$$

Since all coefficients $p \geq d-1$ vanish because of the modulo operation of (6) and the symbolic product of two linearized polynomials results in another linearized polynomial, we have to prove that $F_p = 0$ for $v \leq p \leq d_r - 2$.

$$\sum_{i+l=p} \Lambda_i S_l^{q^i} = \sum_{i=0}^{p} \Lambda_i S_{p-i}^{q^i} = \sum_{i=0}^{p} \Lambda_i \left(\sum_{s=0}^{v-1} E_s z_s^{q^{p-i}} \right)^{q^i}$$

$$= \sum_{s=0}^{v-1} z_s^{q^p} \left(\sum_{i=0}^{p} \Lambda_i E_s^{q^i} \right) = \sum_{s=0}^{v-1} z_s^{q^p} \Lambda(E_s) = 0$$

because p is equal to $v = N(\Lambda)$ or larger and $E_0, E_1, \ldots, E_{v-1}$ are roots of $\Lambda(x)$. ■

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

$$\mathbf{S} \begin{bmatrix} \Lambda_v \\ \Lambda_{v-1} \\ \Lambda_{v-2} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_v \\ -S_{v+1} \\ -S_{v+2} \\ \vdots \\ -S_{2v-1} \end{bmatrix}, \tag{7}$$

with $\mathbf{S}$ defined as

$$\mathbf{S} = \begin{bmatrix} S_0^{q^v} & \cdots & S_{v-1}^{q^1} \\ S_1^{q^v} & \cdots & S_v^{q^1} \\ S_2^{q^v} & \cdots & S_{v+1}^{q^1} \\ \vdots & \ddots & \vdots \\ S_{v-1}^{q^v} & \cdots & S_{2v-2}^{q^1} \end{bmatrix}. \tag{8}$$

It can be shown that the matrix $\mathbf{S}$ is nonsingular. 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 [6]. 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 $q^1, q^2, \ldots, q^v$ (see (8)).

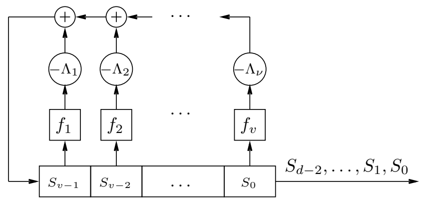

Fig. 2. Row error polynomial as a shift-register

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

$$\Delta_r = S_r + \sum_{j=1}^{L_r} \Lambda_j^{(r)} S_{r-j}^{q^j} = \sum_{j=0}^{L_r} \Lambda_j^{(r)} S_{r-j}^{q^j}. \tag{9}$$

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

$$\Lambda^{(r+1)}(x) = \Lambda^{(r)}(x) + Ax^{q^l} \otimes \Lambda^{(m)}(x), \quad (10)$$

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: From (9) it follows that

$$\Delta'_r = \sum_{j=0}^{L_{r+1}} \Lambda_j^{(r+1)} S_{r-j}^{q^j}.$$

With (10) we can write:

$$\Delta'_r = \sum_{i=0}^{L_r} \Lambda_i^{(r)} S_{r-i}^{q^i} + A \sum_{i=0}^{L_m} \left(\Lambda_i^{(m)} S_{r-i-l}^{q^i} \right)^{q^l}$$

$$= \Delta_r + A \cdot \Delta_m^{q^l} = \Delta_r - \Delta_r \Delta_m^{-q^l} \cdot \Delta_m^{q^l} = 0,$$

where the syndrome $\mathbf{s}$ 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 [6] 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., [5] or [7]). A similar proof as in [7] can be given for the modified Berlekamp-Massey algorithm of Rank-Codes.

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.

Now we can summarise 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 [8].
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 the error vector $\mathbf{e}$ by (2) and the decoded codeword $\hat{\mathbf{c}} = \mathbf{r} - \mathbf{e}$.

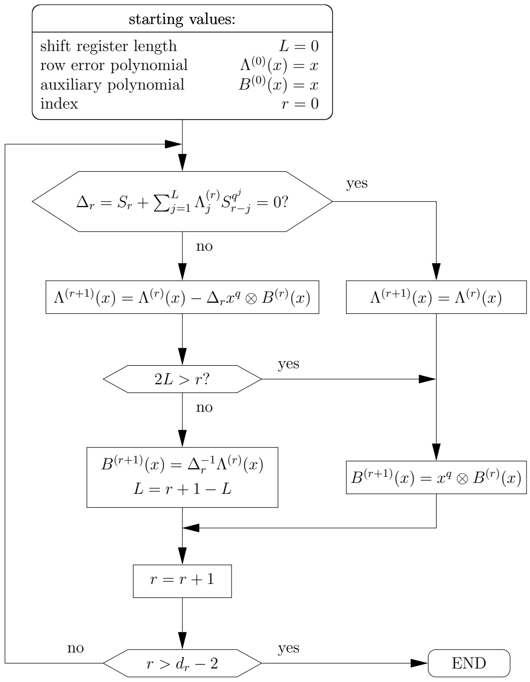

Fig. 3. Berlekamp-Massey algorithm for rank errors

III. Permutation Codes

For a binary matrix of dimension $(N \times n)$, where $n \leq N$, we can use the concept 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

$$|\mathcal{C}| \leq \frac{N!}{(d_{\min} - 1)!} \quad (11)$$

For specific values of N, we have equality in (11). 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 6 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.

A. Applications

Since Permutation codes are now defined over a binary matrix, we can use them also to correct crisscross errors. The combination of permutation codes and M-FSK (frequency shift keying) modulation can be used to correct narrowband- and impulsive noise, when these errors are considered as crisscross error patterns.

1) Crisscross and Random Errors: 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

$$d_{\min} > 2(t_{\text{row}} + t_{\text{column}}) + t_{\text{random}}\,, \quad (12)$$

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

2) M-FSK: In an M-FSK modulation scheme, symbols are modulated as one of $M = N$ orthogonal sinusoidal waves and a non-coherent demodulator detects N envelopes. In a hard decision detector (detecting presence of a frequency), we put the binary outputs in an $N \times n$ decoding matrix. We output the message corresponding to the codeword that has the maximum number of agreements with the demodulator output. Several channel disturbances can be considered:

- Narrow band noise causes large demodulator envelopes and thus a row is set equal to 1;
- Impulse noise has a broadband character and thus leads to a column with all entries equal to 1;
- Background noise introduces incorrect decisions, i.e., insertion or deletion of a symbol 1;
- Fading will cause disappearance of a particular envelope. In this case, a row of the decoding matrix is set to 0.

The hard decision non-coherent demodulation in combination with the permutation code allows the correction of $d_{\min} - 1$ incorrect demodulator outputs caused by narrow band- , impulse-, background noise and fading.

3) FFH/M-FSK Multi-Access: In frequency hopping, we can assign permutation codewords as signature sequences to particular users. As an example, for a distance $N-1$ code, we have $N(N-1)$ codewords and codewords differ in at least $N-1$ positions, [9]. Hence, signatures agree in maximum 1 position. Again, codewords can be represented in a binary $N \times n$ matrix. A decoder checks the presence of its signature in the matrix. The knowledge of the distance structure enables us to calculate the error probability more exactly, see [10]. In addition to interference, we can also correct (crisscross) errors due to narrowband- and impulsive noise.

B. Complexity of Decoding

Decoding of these 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 vector has to be compared with all codewords in the codebook. As an alternative approach, we developed the concepts of permutation trellis codes [11], where the Viterbi algorithm is used to do the decoding with low complexity. The principle is given in Fig. 4.

The mapping converts binary b-tuples from the convolutional code into N-tuples or codewords from a permutation code. The key idea is that the distance between any two N-tuples is at least as large as the distance between the corresponding b-tuples. This property is called distance preserving.

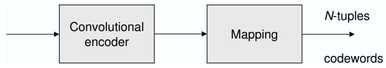

Fig. 4. Encoding process for a distance preserving convolutional/permutation code

As an example, we give the state transition diagram of a simple 4 state encoding process using a binary $R = 1/2$ convolutional code and an $N = 3$, $d_{\min} = 2$ permutation code in Table I.

TABLE I
ENCODING AND DISTANCE PRESERVED MAPPING

	input 0 1	input 0 1	input 0 1
old state	new state	binary 2-tuples	mapping
0	0 1	00 11	231 123
1	2 3	01 00	213 132
2	0 1	11 00	123 231
3	2 3	10 01	132 213

Note that the Hamming distance between any two 2-tuples is increased by one for the corresponding codewords from the permutation code. The binary free distance for this particular code is 5 before the mapping and by inspection it is easy to see that it is 8 after the mapping. Every branch in the trellis of the convolutional code corresponds to a permutation code word. The decoding is easy, since we compare the codewords along the trellis branches with the received N-tuple and perform ML Viterbi decoding.

IV. 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.

REFERENCES

[1] Mikael Sternad, Tommy Svensson, and Göran Klang. The WINNER B3G system MAC concept. In *Proceedings IEEE Vehicular Technology Conference (VTC 2006-Fall), Montréal, Canada*, September 2006.

[2] P. Delsarte. Bilinear forms over a finite field with applications to coding theory. *Journal of combinatorial theory. Series A*, 25(4):226–241, 1978.

[3] E. M. Gabidulin. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii*, 21(1):3–16, Januar–March 1985.

[4] R. M. Roth. Maximum-rank array codes and their application to crisscross error correction. *IEEE Trans. Inf. Theory*, 37(2):328–336, March 1991.

[5] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. Inf. Theory*, IT-15:122–127, January 1969.

[6] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley, Owego, New York 13827, 1983. ISBN 0-201-10102-5.

[7] K. Imamura and W. Yoshida. A simple derivation of the Berlekamp-Massey algorithm and some applications. *IEEE Trans. Inf. Theory*, IT-33:146–150, January 1987.

[8] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw Hill, New York: Mc Graw-Hill, 1968.

[9] E. L. Titlebaum. Time frequency hop signals. *IEEE Transaction on Aerospace and Electronic Systems*, AES-17:490–494, 1981.

[10] A. J. Han Vinck. On permutations codes. In *Proceedings International Symposium on Communication Theory and Applications, Ambleside, UK*, pages 491–495, July 2001.

[11] Hendrik Ferreira, A. J. Han Vinck, Th. Swart, and I. de Beer. Permutation trellis codes. *IEEE Trans. Commun.*, 53(11):1782–1789, 2005.