Generalized concatenated coding and Fourier transform

Valentine B. Afanassiev,
Alexander A. Davydov and Vladimir G. Potapov
afanv,adav,potapov@iitp.ru
Institute for Information Transmission Problems (Kharkevich Institute), Russian Academy
of Sciences, Bol’shoi Karetnyi per. 19, GSP-4, Moscow, 127994, Russian Federation

Seventh International Workshop on Optimal Codes and Related Topics
September 6-12, 2013, Albena, Bulgaria

Definition of Reed-Solomon (RS) codes

$\left(n,k,d\right),\ d=n-k+1,$ RS code over $GF\left(q\right)$

$n\,|\left(q-1\right)$ $\quad n\geq k$

Information symbols: $\mathbf{U}=\left(U_0,U_1,\ldots,U_{k-1},0\ldots0\right)$

$U_i\in GF\left(q\right)$

Codewords:

$\mathbf{C}=\mathbf{U}\Phi^{-1}$, $\quad \Phi=\left(\alpha^{ij}\right),\ i,\ j=\overline{0,n-1}$,

$\mathbf{\Phi}$ is the Fourier Transform matrix

$$\mathbf{C}\Phi=\mathbf{U}\Phi^{-1}\Phi=\mathbf{U}$$

Generalized Concatenated Code (GCC)

Matrix of information symbols: $\mathfrak{I}_k\left(a_{i,j}\right); n\times n$ bounded by *k* rows and *k* columns,

$$a_{i,j}\in GF\left(q\right), i, j=\overline{1,k}, a_{i,j}=0, i, j=\overline{k+1,n}$$

Encoding GCC of the level *k*: $\mathbf{C}_k=\Phi^{-1}\mathfrak{I}_k\Phi^{-1}$

(two dimensional Fourier transform)

GCC distance:

$$D_k=\min_{i,j,a_{ij}\neq const}\left[\left(n-i+1\right)\left(n-j+1\right)\right]$$

Example

$$\mathfrak{I}_k = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kk} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Generalized Concatenated Code (GCC) and Product code (PC):

Rectangular form of

$$\Im_k\left(a_{i,j}\right)$$

produces a **product code**,

and any other form

produces a **Generalized Concatenated** code.

Some GCC $\subset$ PC.

Any GCC $\supset$ PC .

Layers of GCC and PC

$$\Phi^{-1}\Delta_k\Phi^{-1}$$

$\Delta_k = \left(\delta_{ij}\right), \delta_{i,j} \in GF\left(q\right),$ (green cross on Example)

$\delta_{ij} = 0 \text{ if } i < k \;\&\; j < k, \text{ if } i = k \;\&\; j > k, \text{ if } i > k \;\&\; j = k.$

Layer by layer decomposition of GCC:

$$\mathbf{C}_{k+1} = \mathbf{C}_k + \Phi^{-1}\Delta_{k+1}\Phi^{-1} = \Phi^{-1}\left(\mathfrak{I}_k + \Delta_{k+1}\right)\Phi^{-1}$$

or

$$\mathbf{C}_{k-1} = \mathbf{C}_k - \Phi^{-1}\Delta_k\Phi^{-1} = \Phi^{-1}\left(\mathfrak{I}_k - \Delta_k\right)\Phi^{-1}$$

Encoding scheme of GCC or PC

Given $\mathfrak{I}_k\left(a_{i,j}\right)$

First step:

Find $\mathbf{A}_{Row} = \mathfrak{I}_k \Phi^{-1} = \begin{pmatrix} \mathbf{A}_{k\times n} \\ \mathbf{0}_{n-k\times n} \end{pmatrix}$ outer row codes

Or

Find $\mathbf{A}_{Col} = \Phi^{-1}\mathfrak{I}_k = \left(\mathbf{A}_{n\times k} \ \ \mathbf{0}_{n\times n-k}\right)$ outer column codes

Second step: $\mathbf{C}_k = \Phi^{-1}\mathbf{A}_{Row} \overset{or}{=} \mathbf{A}_{Col}\Phi^{-1}$

All rows and columns of $\mathbf{C}_k$ *are codewords of*

inner row and column (n,k,d) *RS codes.*

BASIC RELATIONS

1. $$\mathfrak{I}_k = \Phi \mathbf{C}_k \Phi$$

2. $$\mathbf{V} = \mathbf{C}_k + \mathbf{E}$$

3. $$\Phi \mathbf{V} \Phi = \mathfrak{I}_k + \Phi \mathbf{E} \Phi$$

4. $$\Phi \mathbf{V} = \mathbf{A}_{Row} + \Phi \mathbf{E} = \begin{pmatrix} \mathbf{A}_{k\times n} \\ \mathbf{0}_{n-k\times n} \end{pmatrix} + \begin{pmatrix} \varepsilon_{k\times n} \\ \varepsilon_{n-k\times n} \end{pmatrix} = \begin{pmatrix} * \\ \varepsilon_{n-k\times n} \end{pmatrix}$$

5. $$\mathbf{V}\Phi = \mathbf{A}_{Col} + \mathbf{E}\Phi = \left(\mathbf{A}_{n\times k} \quad \mathbf{0}_{n\times n-k} \right) + \left(\varepsilon_{n\times k} \mid \varepsilon_{n\times n-k} \right) = \left(* \mid \varepsilon_{n\times n-k} \right)$$

6. $\Phi \mathbf{E} \Phi$ contains syndromes of all row and column outer codes.

7. $\Phi \mathbf{E}$ and $\mathbf{E} \Phi$ contains syndromes of all inner column and row codes

Standard GCC decoding

1. Set the starting layer of GCC $i = \ell$.
2. Inner code $\mathbf{C}_{Col,i}$ decoding in all columns *(results: correction or rejection).*
3. Outer code $\mathbf{R}_{Row,i}$ decoding in the i- th row *(result: correction or rejection).*
4. If the result of Step 3 is rejection then Stop decoding,
else transition to the $(i-1)$- th layer of GCC, $i = (i-1) > 0$, and go to Step 2.

Iterative PC decoding (simple algorithm)

1. Column code C_{Col} decoding in all columns *(results: correction and/or rejection).*

2. Row code R_{Row} decoding in all rows *(results: correction and/or rejection).*

3. If no one correction and/or rejection were made on the Step 2 then stop decoding, else return to Step 1 until *additional stop condition is not satisfied.*

Iterative GCC decoding (general idea)

1. Set the starting layer of GCC $i = j = k$.
2. Iterative Product Code $\mathbf{R}_{Row,i} \times \mathbf{C}_{Col,j}$ **decoding**
(results: correction and/or rejection).
3. Outer codes $\mathbf{R}_{Row,i}$ **and** $\mathbf{C}_{Col,j}$ **decoding in i-th row and j-th column (*result: correction* and/or rejection).**
4. If the result of Step 3 is rejection in the both row and column then Stop decoding,
else transition to the *i = i-1* (*correction in the row*)
and /or j = j-1 (*correction in the column*) layer of GCC
and go to Step 2*.*

Notations: $d_{Col,i}, d_{Row,i}$ are code distances of inner column and row component codes,

$D_{Col,i}, D_{Row,i}$ are code distances of outer component codes of GCC.

***Lemma* 1. Minimal Stop-Set of GCC i-th layer is any configuration with $d_{Col,i}/2$ errors in $D_{Row,i}$ columns**.

***Lemma* 2. Minimal Stop-Set of PC is any configuration with $d_{Col}/2$ errors in d_{Row} columns such that there is no one row of less $d_{Row}/2$ errors**.

***Lemma* 3. Minimal Stop-Set of GCC *i-th* layer with using of symmetry of Fourier designed GCC is as follows:**
any configuration with $d_{Col,i}/2$ errors in $D_{Row,i}$ columns and with $d_{Row,i}/2$ errors in $D_{Col,i}$ rows.

***Lemma* 4. Not all non minimal stop-sets of GCC decoder (standard or iterative) include a PC decoder stop-set.**

Proof. Let us expand a minimal stop-set of Lemma 3 (for example) adding one row and one column. Then we get a configuration of $D_{Row,i}$ *+1* rows of the weight $d_{Col,i}/2$ *and* $D_{Col,i}$ *+ 1* columns of the weight $d_{Row,i}/2$. *Remove any* one error from this configuration. Then we get one row and one column of the weight 1 less than others and the rest of the configuration will be a stop-set for GCC decoder.

PC decoder iteratively and alternately finds and corrects the light column, then corrects all the light rows and so on up to correction all errors or to a stage when GCC decoder will be able to continue and finish correction of a rest of the error configuration.

Example: $d_{Col,i} = d_{Row,i} = 3$, $D_{Col,i} = D_{Row,i} = 9$

1	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	1

The main result

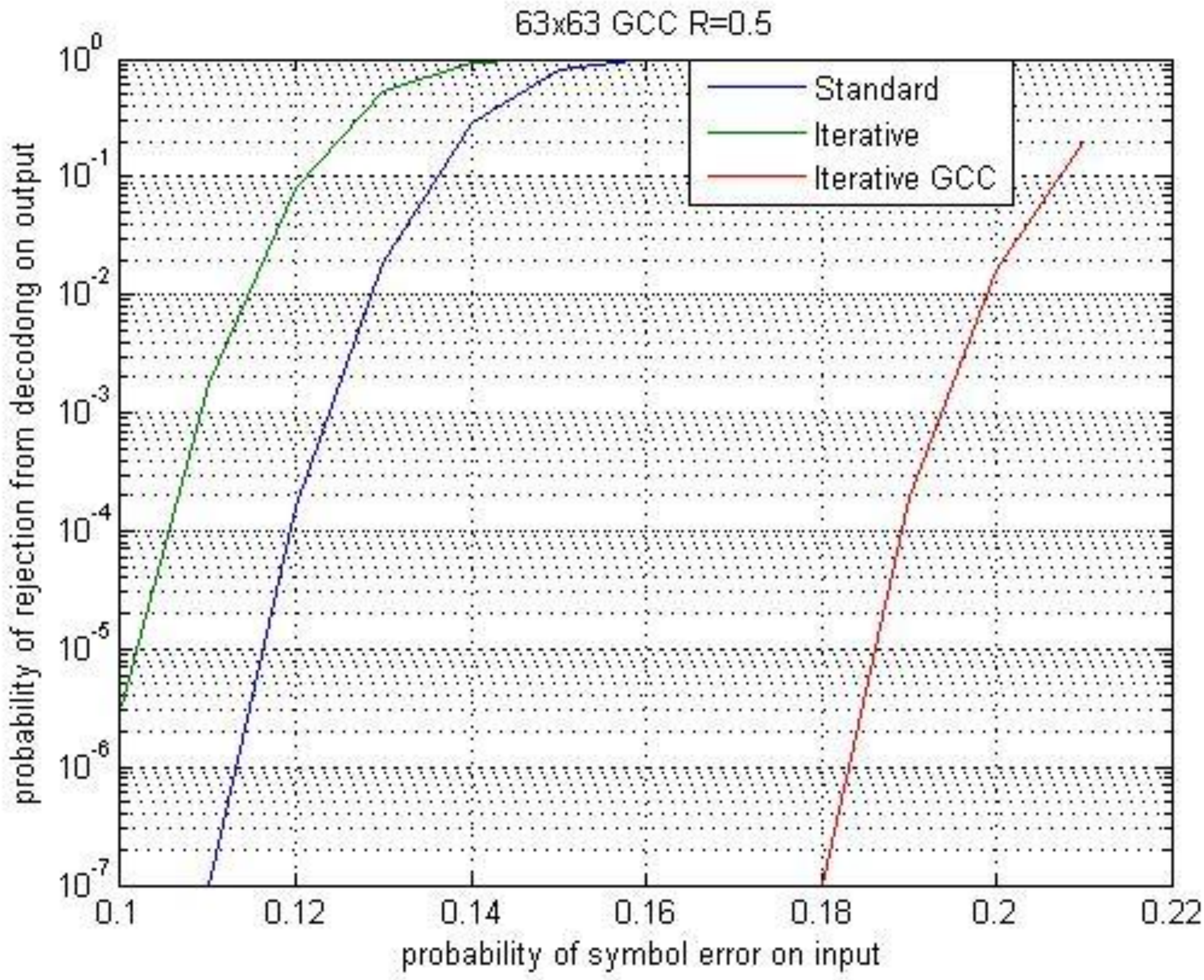
The considered configuration will stop standard GCC decoding algorithms but,

as a corollary of Lemma 4, the PC decoding can execute few successive iterations,

after wards the GCC decoding will be able to finish the error correction with success.

That means that union of PC and GCC decoding will expand a

set of correctable error configurations.


63x63 GCC R=0.5
Standard
Iterative
Iterative GCC
probability of rejection from decodong on output
probability of symbol error on input
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
0.1
0.12
0.14
0.16
0.18
0.2
0.22

% 63x63 3960 code symbols
%
% 37xRS(63,45)
% 2xRS(63,43)
% 2xRS(63,41)
% 2xRS(63,39)
% 2xRS(63,37)
%
% 1985 - information symbols in GF(2^6)
% r=0.5001 – the code rate

Codeword area 63x63.
Information symbols area 45x45.