Iterative Least-Squares Decoding of Analog Product Codes

Marzia Mura, Werner Henkel, and Laura Cottatellucci
Telecommunications Research Center (ftw.)
Donau-City-Str. 1, A-1220 Vienna, Austria
Email: {mura, cottatellucci}@ftw.at

W. Henkel is now with:
University of Applied Sciences
Neustadtswall 30, D-28199 Bremen, Germany
Email: werner.henkel@ieee.org

***Abstract*— We define analog product codes as a special case of analog parallel concatenated codes. These codes serve as a tool to get some more insight into the convergence of "Turbo" decoding. We prove that the iterative "Turbo"-like decoding of analog product codes indeed is an estimation in the least-squares sense and thus correspond to the optimum maximum-likelihood decoding.**

I. Introduction

Since the invention of "Turbo" codes in 1993 [1], the coding community has developed some insights into the behavior of iterative decoding. So-called EXIT charts [5] show the development of the mutual information between the transmitted data and the extrinsic information (a-priori of the following decoding). Thus they are a suitable tool to view the convergence in the water-fall region of the bit-error rate curves. At the high-SNR far end of the curve, the distance properties of the code determine the performance. The minimum distance can be estimated by sending simple weight-2 or weight-3 patterns and checking for the weight of the encoded sequence.

The intuitive understanding of the iterative behavior, however, is still not very developed. In this paper, we try an approach based on so-called Analog Codes [2–4], which are codes over real or over complex numbers. Without discretizing (quantizing), the iterative Turbo-like decoding algorithm can be understood with the tools of numerical mathematics. We show that the iterative decoding indeed converges to the least-squares solution which corresponds to the maximum-likelihood decoding.

After describing the encoding of an analog product code in the next section, we discuss the decoding in Section III, followed by a more detailed analysis of the convergence in sections IV to VI. An exemplary simulation result and some remarks conclude the paper.

II. Encoding

The considered discrete time communication system consists of a source with statistically independent analog (real) symbols $\boldsymbol{X}$, a block analog encoder, an additive noise channel, and a decoder. For simplicity, we consider the information sequence $\boldsymbol{X}$ of length N to be the realization of a white Gaussian stochastic process with mean 0 and variance $\sigma^2 = 1$.
We assume that $N = n^2$ so that the sequence can be represented as a matrix X of dimension $n \times n$:

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,n} \end{bmatrix}, \tag{1}$$

whose rank is not greater than n. The block analog encoder we refer to extends the idea of parity check block codes for binary sources to analog sources. It maps the X matrix into the matrix X^*,

$$\boldsymbol{X}^* = \left[\begin{array}{ccc|c} x_{1,1} & \cdots & x_{1,n} & x_{1,n+1} \\ \vdots & \ddots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,n} & x_{n,n+1} \\ \hline x_{n+1,1} & \cdots & x_{n+1,n} & x_{n+1,n+1} \end{array}\right], \tag{2}$$

such that

$$\begin{aligned} x_{j,n+1} &= -\sum_{i=1}^{n} x_{j,i} \,, \\ x_{n+1,j} &= -\sum_{i=1}^{n} x_{i,j} \,, \\ x_{n+1,n+1} &= \sum_{i=1}^{n}\sum_{j=1}^{n} x_{i,j} \,. \end{aligned}$$

This describes a product code with analog 'parity-check' component codes, *i.e.*, DC-free component codes.
The average power of the information components is chosen to be 1. Since the sum of n Gaussian random variables with zero mean and variance σ^2 is still a Gaussian random variable with zero mean and variance $\sigma_n^2 = n \cdot \sigma^2$, the average power of the parity symbols is equal to $n \cdot \sigma^2 = n$. The mean power of the

codeword is

$$P_a = \frac{4n^2}{(n+1)^2} \ . \tag{3}$$

The rank of the matrix that represents the codeword is still equal to n. Then the number of redundant symbols is equal to $2n+1$ and consequently the code rate is equal to $n^2/(n^2+2n+1)$.

Let us suppose that the codeword is corrupted by additive Gaussian noise on a memoryless channel. Then in general the rank of the received matrix is equal to $n+1$.

III. Decoding Strategy

In this section, the decoding algorithm is described. Given the received codeword $\boldsymbol{R}_{(0)}$, the basic idea is to compute two new matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ whose elements are

$$\begin{aligned} r_{1j,i} &= -\sum_{\substack{i=1 \\ i\neq j}}^{n+1} r_{j,i} \ , \qquad j = 1, \cdots, n+1 \ , \\ r_{2j,i} &= -\sum_{\substack{j=1 \\ j\neq i}}^{n+1} r_{j,i} \ , \qquad i = 1, \cdots, n+1 \ , \end{aligned} \tag{4}$$

with $r_{j,i}$ denoting the components of $\boldsymbol{R}_{(k)}$, $k = 0, 1, \ldots$. The algorithm is the analog counterpart of the one described in, *e.g.*, [6] for binary array codes. Equations (4) show the extrinsic information of the rows and columns, respectively, knowing that the rows and columns sum to Zero.

The steps in (4) can be written as the elements of two update matrices

$$\overline{\boldsymbol{I}}\boldsymbol{R}_{(n-1)} \ \text{ and } \ \overline{\boldsymbol{I}}\boldsymbol{R}^T_{(n-1)} \ , \tag{5}$$

where

$$\overline{\boldsymbol{I}} = -\begin{bmatrix} 0 & 1 & \cdots & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 1 & \cdots & \cdots & 1 & 0 \end{bmatrix} . \tag{6}$$

The actual matrix is then computed as the weighted sum of the previous matrix and the update matrices:

$$\begin{aligned} \boldsymbol{R}_{(n)} &= (\boldsymbol{R}_{(n-1)} - w\overline{\boldsymbol{I}}\boldsymbol{R}_{(n-1)} - w\overline{\boldsymbol{I}}\boldsymbol{R}^T_{(n-1)})/(1+2w) \\ &= (\boldsymbol{R}_{(n-1)} - w\overline{\boldsymbol{I}}(\boldsymbol{R}_{(n-1)} + \boldsymbol{R}^T_{(n-1)}))/(1+2w) \end{aligned} \tag{7}$$

In order to simplify the description of the method, the received matrix is written as a vector that contains the sequence of its rows. Let this vector be $\boldsymbol{y}_{(0)}$. The computation can then be described as

$$\begin{aligned} \boldsymbol{y}_{(k)} &= \boldsymbol{\Phi}\boldsymbol{y}_{(k-1)} \\ &= \boldsymbol{\Phi}^k\boldsymbol{y}_{(0)} \ . \end{aligned} \tag{8}$$

$\boldsymbol{y}_{(k)}$ is the vector obtained at the k^{th} iteration step. $\boldsymbol{\Phi}$ is the iteration matrix of dimension $(n+1)^2 \times (n+1)^2$ defined as

$$\begin{aligned} \boldsymbol{\Phi} &= (\boldsymbol{I} - w\boldsymbol{M}_1 - w\boldsymbol{M}_2)/(1+2w) \\ &= (\boldsymbol{I} - w(\boldsymbol{M}_1 + \boldsymbol{M}_2))/(1+2w) \ . \end{aligned} \tag{9}$$

$\boldsymbol{I}$ is the $(n+1)\times(n+1)$ identity matrix, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ describe the computation per columns and per rows, respectively, w is a weight which depends on the matrix dimension.

$$\boldsymbol{M}_1 = \begin{bmatrix} \overline{\boldsymbol{I}} & \boldsymbol{Z} & \cdots & \boldsymbol{Z} \\ \boldsymbol{Z} & \overline{\boldsymbol{I}} & \boldsymbol{Z} \ \cdots & \boldsymbol{Z} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Z} & \boldsymbol{Z} & \cdots & \overline{\boldsymbol{I}} \end{bmatrix} \tag{10}$$

$$\boldsymbol{M}_2 = \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{Z} & \boldsymbol{I} \ \cdots & \boldsymbol{I} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{Z} \end{bmatrix} \tag{11}$$

$\boldsymbol{Z}$ denotes a matrix of zeros of dimension $(n+1)\times(n+1)$. $\boldsymbol{\Phi}$ is Hermitian as well as the sum $\boldsymbol{M}_1 + \boldsymbol{M}_2$.

IV. Convergence

An iterative method defines a sequence of vectors

$$\boldsymbol{y}_{(0)}, \quad \boldsymbol{y}_{(1)}, \quad \cdots \quad \boldsymbol{y}_{(k)} \quad , \tag{12}$$

such that ideally the sequence would converge to

$$\boldsymbol{y}_{(k)} \xrightarrow{k\to\infty} \boldsymbol{x}^*_{(vect)} \tag{13}$$

or

$$\|\boldsymbol{y}_{(k)} - \boldsymbol{x}^*_{(vect)}\| \xrightarrow{k\to\infty} \boldsymbol{0} \ , \tag{14}$$

where $\boldsymbol{x}^*_{(vect)}$ indicates the vector composed of the rows of the matrix $\boldsymbol{X}^*$. Because of the equivalence of finite norms, $\|.\|$ may be any norm (*cf.* [9]). The necessary condition for the method not to diverge is

$$\rho(\boldsymbol{\Phi}) \leq 1 \ , \tag{15}$$

i.e., the spectral radius of the iteration matrix must be less than or equal to one.

V. Eigenvalues

In this section we prove that under some conditions the spectral radius of the iterative matrix Φ is less or equal to 1. A Hermitian matrix is diagonalized by its eigenvectors. It can be written as

$$\begin{aligned} (1+2w)\boldsymbol{\Phi} &= \boldsymbol{I} - w(\boldsymbol{M}_1 + \boldsymbol{M}_2) \\ &= \boldsymbol{T}(\boldsymbol{I} - w(\boldsymbol{\Gamma} + \boldsymbol{\Delta})\boldsymbol{T}^H \\ &= \boldsymbol{T}(\boldsymbol{\Lambda})\boldsymbol{T}^H \ , \end{aligned} \tag{16}$$

where $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$ are the diagonally similar matrices of $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, respectively. $\boldsymbol{\Lambda}$ is the diagonal matrix of the eigenvalues and the columns of $\boldsymbol{T}$ are the corresponding eigenvectors.

For simplicity, we rewrite the sum $\boldsymbol{M}_1 + \boldsymbol{M}_2$ as the sum of two other matrices $\tilde{\boldsymbol{M}}_1$ and $\tilde{\boldsymbol{M}}_2$

$$\boldsymbol{M}_1 + \boldsymbol{M}_2 = \tilde{\boldsymbol{M}}_1 + \tilde{\boldsymbol{M}}_2 \ , \tag{17}$$

where

$$\tilde{\boldsymbol{M}}_1 = \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \boldsymbol{I} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{I} & \cdots & \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix} \tag{18}$$

and

$$\tilde{\boldsymbol{M}}_2 = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{Z} & \cdots & \boldsymbol{Z} \\ \boldsymbol{Z} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{Z} & \cdots & \boldsymbol{Z} & \boldsymbol{E} \end{bmatrix} . \tag{19}$$

$\boldsymbol{E}$ is a matrix of all ones of dimension $(n+1) \times (n+1)$.
We can notice that $\tilde{\boldsymbol{M}}_1$ and $\tilde{\boldsymbol{M}}_2$ are two block circular matrices whose blocks have the same dimensions. Consequently, they can be block diagonalized by the same matrix [10]

$$\boldsymbol{U} = \boldsymbol{F} \otimes \boldsymbol{I}. \tag{20}$$

$\otimes$ denotes the Kronecker product and $\boldsymbol{F}$ is the Fourier transformation matrix:

$$\boldsymbol{F} = \frac{1}{\sqrt{n+1}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \cdots & \omega^n \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^n & \omega^{2(n)} & \cdots & \omega^{(n)(n)} \end{bmatrix} , \tag{21}$$

where $\omega = e^{-\frac{2\pi j}{n+1}}$. We obtain

$$\tilde{\boldsymbol{M}}_1 = \boldsymbol{U}(\boldsymbol{\Gamma})\boldsymbol{U}^H , \tag{22}$$

where $\boldsymbol{\Gamma}$ is diagonal. Since $\tilde{\boldsymbol{M}}_2$ is already block diagonal

$$\tilde{\boldsymbol{M}}_2 = \boldsymbol{U}(\tilde{\boldsymbol{M}}_2)\boldsymbol{U}^H = \tilde{\boldsymbol{M}}_2 , \tag{23}$$

to diagonalize $\tilde{\boldsymbol{M}}_2$, we can independently diagonalize every block $\boldsymbol{E}$ with the matrix $\boldsymbol{F}$. Therefore, the matrix that diagonalizes $\tilde{\boldsymbol{M}}_2$ is

$$\widehat{\boldsymbol{F}} = \boldsymbol{I} \otimes \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{Z} & \cdots & \cdots & \boldsymbol{Z} \\ \boldsymbol{Z} & \boldsymbol{F} & \boldsymbol{Z} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{Z} \\ \boldsymbol{Z} & \cdots & \cdots & \boldsymbol{Z} & \boldsymbol{F} \end{bmatrix} \tag{24}$$

and

$$\begin{aligned} \text{diag}(\tilde{\boldsymbol{M}}_1 + \tilde{\boldsymbol{M}}_2) &= \widehat{\boldsymbol{F}}^H \boldsymbol{U}^H (\tilde{\boldsymbol{M}}_1 + \tilde{\boldsymbol{M}}_2) \boldsymbol{U} \widehat{\boldsymbol{F}} \\ &= \boldsymbol{T}^H (\tilde{\boldsymbol{M}}_1 + \tilde{\boldsymbol{M}}_2) \boldsymbol{T} . \end{aligned} \tag{25}$$

Now, we concentrate on the eigenvalues of $\tilde{\boldsymbol{M}}_1$ and $\tilde{\boldsymbol{M}}_2$ that constitute the diagonal matrix. The eigenvalues of a circular matrix can be found as the Discrete Fourier Transform (DFT) of its first column. Since $\tilde{\boldsymbol{M}}_1$ is block circular but not circular, we look for a block diagonal matrix

$$\begin{aligned} \boldsymbol{D}_b &= \boldsymbol{U}\tilde{\boldsymbol{M}}_1\boldsymbol{U}^H \\ &= \begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{Z} & \cdots & \boldsymbol{Z} \\ \boldsymbol{Z} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{Z} & \cdots & \boldsymbol{Z} & \boldsymbol{D}_n \end{bmatrix} . \end{aligned} \tag{26}$$

The element d_{ij} of $\boldsymbol{D}_k$ $(0 \leq k \leq n)$ is the k^{th} element of the DFT of the vector consisting of the elements on position (i,j) of every block of $(-\boldsymbol{I}, \boldsymbol{I}, ..., \boldsymbol{I})$, which is the first "block column" of $\tilde{\boldsymbol{M}}_1$. Since the vectors of elements $d_{i,j}$ with $i \neq j$ are always equal to zeros, the elements outside the main diagonal are equal to zero and $\boldsymbol{D}_b$ is diagonal. Because of the symmetries, for every element d_{ii} we have to consider the same vector v:

$$\boldsymbol{v} = (-1, 1.....,1) , \tag{27}$$

whose DFT is:

$$V_1 = \sum_{k=0}^{k=n} v_k = n-1 , \tag{28}$$

$$V_2 = \sum_{k=0}^{k=n} v_k e^{(\frac{-2\pi ji}{n+1})2} , \tag{29}$$

$$\vdots \tag{30}$$

$$V_k = \sum_{k=0}^{k=n} v_k e^{(\frac{-2\pi ji}{n+1})k} ; \tag{31}$$

we obtain

$$\begin{aligned} V_k &= -1 + \sum_{k=1}^{k=n} v_k e^{(\frac{-2\pi ji}{n+1})k} \\ &= -1 + \frac{1 - e^{(2\pi ji)}}{1 - e^{\frac{(2\pi ji)}{n+1}}} - e^0 \\ &= -2 . \end{aligned} \tag{32}$$

The eigenvalues of $\boldsymbol{E}$ are the DFT of the vector $[1, 1, \cdots, 1]$ which are $[n+1, 0, \cdots, 0]$.
If we indicate the main diagonal of $\boldsymbol{\Lambda}$ as maindiag(Λ), we have:

$$\begin{aligned} &(1+2w)\text{maindiag}(\boldsymbol{\Lambda}) = \\ &\underbrace{[1, 1, \cdots, \cdots, 1]}_{(n+1)^2} + \\ &-w * \underbrace{[\underbrace{n-1, \cdots, n-1}_{n+1}, \underbrace{-2, \cdots, -2}_{n+1}, \cdots, \underbrace{-2, \cdots, -2}_{n+1}]}_{(n+1)^2} + \\ &-w * \underbrace{[\underbrace{n+1, 0, \cdots, 0}_{n+1}, \underbrace{n+1, 0 \cdots, 0}_{n+1}, \cdots, \underbrace{n+1, 0, \cdots, 0}_{n+1}]}_{(n+1)^2} \end{aligned} \tag{33}$$

Therefore, there are:

$$\begin{array}{lll} 1 & \text{eigenvalue equal to} & \frac{1-2wn}{1+2w}\ , \\ 2n & \text{eigenvalues equal to} & \frac{1-w(n-1)}{1+2w}\ , \\ n^2 & \text{eigenvalues equal to} & 1\ . \end{array} \tag{34}$$

We see that there exist n^2 eigenvalues equal to one, which is the dimension of the subspace of the solutions of the system.
For convergence we require

$$\begin{cases} |1-2wn| \leq 1\ , \\ |1-w(n-1)| \leq 1\ , \end{cases} \tag{35}$$

which yields the condition

$$0 \leq w \leq 1/n\ . \tag{36}$$

VI. Least-Squares Decoding

We found that

$$\boldsymbol{\Phi} = \boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^H\ . \tag{37}$$

$\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues of $\boldsymbol{\Phi}$ described in the previous section, $\boldsymbol{T}$ consists of the eigenvectors $t_1....t_n$ of $\boldsymbol{\Phi}$, and since $\boldsymbol{\Phi}$ is Hermitian, they are mutually independent, and eigenvectors associated with different eigenvalues are mutually orthogonal. Without loss of generality, we can assume that

$$t_i^T t_j = \delta_{ij}\ , \quad i,j = 1.....n\ . \tag{38}$$

Equation (8) can be rewritten as

$$\boldsymbol{y}_{(k)} = \boldsymbol{\Phi}^{(k)}\boldsymbol{y}_{(0)} = \boldsymbol{T}\boldsymbol{\Lambda}^{(k)}\boldsymbol{T}^H\boldsymbol{y}_{(0)}\ , \tag{39}$$

Since

$$\lim_{k\to\infty}(\text{maindiag}(\boldsymbol{\Lambda}^k)) = [\underbrace{\underbrace{0,\cdots,0}_{n+1},0,\underbrace{1,\cdots,1}_{n},\cdots,0,\underbrace{1,\cdots,1}_{n}}_{(n+1)^2}]\ , \tag{40}$$

the vector $\boldsymbol{y}^{(k)}$, $k \to \infty$ is the projection of $\boldsymbol{y}^{(0)}$ onto the subspace spanned by the eigenvectors associated with the eigenvalue equal to one. For a three dimensional illustration, see Fig. 1.

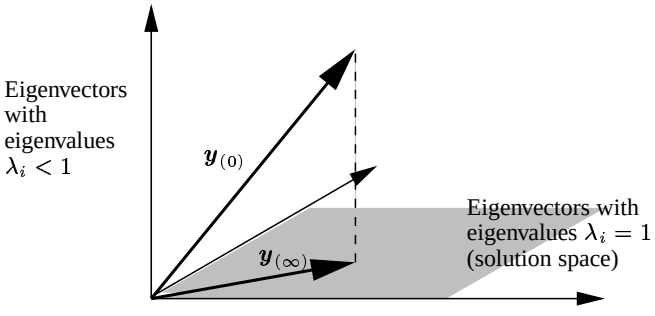

Fig. 1. The projection onto the solution subspace spanned by the eigenvectors with eigenvalues 1 – a 3-dimensional simplification

We would need to prove that the solution space is really spanned by the eigenvectors with eigenvalues one. Due to space limitations we have to postpone this to a later publication.

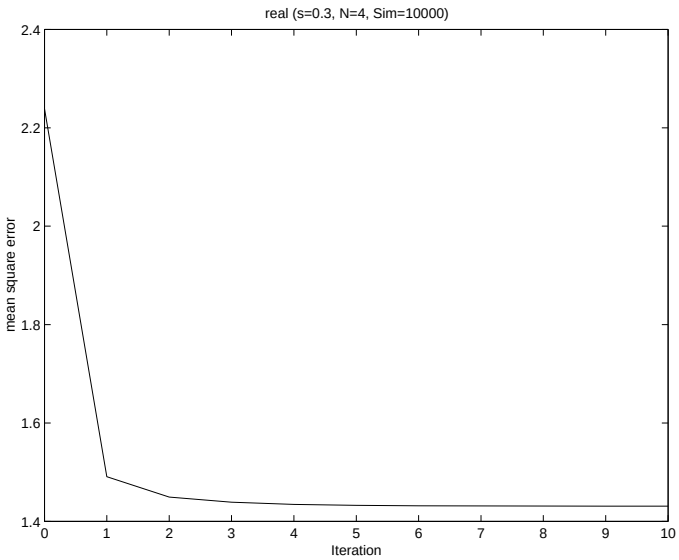

Fig. 2. Simulation Result and Concluding Remarks

VII. Simulation Results

Simulations show a good improvement of the mean square error after just a few iteration cycles as shown in Fig. 2, but then, it does not decrease any further. The values we obtain agree with the values obtained by applying the classical least squares algorithm. As expected, this method is not able to eliminate the noise components that lie in the solution subspace. Our further work will be devoted to how a further improvement is achieved by restricting the alphabet again to be discrete. This brings us nearer to standard 'Turbo' codes.

Acknowledgments

We would like to thank Markus Rupp for some early contributions regarding the convergence conditions and Jossy Sayir for valuable discussions and critical reading of the manuscript.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proc. ICC*, May 1993, pp. 1064-1070.

[2] J.K. Wolf, "Analog Codes", *IEEE Int. Conf. on Comm. (ICC '83)*, Boston, MA, USA, June 19-22, 1983, Vol. 1, pp. 310-12.

[3] W. Henkel, "Multiple Error Correction with Analog Codes," *AAECC-6*, Rome, and *Springer Lecture Notes in Computer Science* 357, 1989, pp. 239-249, (http://userver.ftw.at/∼henkel).

[4] W. Henkel, *Zur Decodierung algebraischer Blockcodes über komplexen Alphabeten*, VDI-Verlag, Düsseldorf, 1989. (in German), (http://userver.ftw.at/∼henkel).

[5] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Tr. on Comm.*, Vol. 49, No. 10, October 2001, pp. 1727-1737.

[6] J. Hagenauer, E. Offer, L. Papke: "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Tr. on Inf. Th.*, Vol. IT-42, No. 2, March 1996, pp. 429-445.

[7] ITUU-TR-1998/4

[8] F. Hlawatsch, (2001) "Statistical Signal Processing: Parameter Estimation Methods," Institut Für Nachrichtentechnik und Hochfrequenztechnik.

[9] V. Comincioli, "Analisi Numerica, Metodi Modelli e Applicazioni," McGraw-Hill 1990. (in Italian)

[10] R.M. Gray, "Toeplitz and Circulant Matrices: A review", Information System Laboratory Department of Electrical Engineering Stanford University, 2001.