A Comparison between Weight Spectrum of Different Convolutional Code Types

Baltă Horia[1], Kovaci Maria[2]

Abstract: In this paper we present the non-recursive systematic, recursive systematic and non-recursive non-systematic 1/2 rate memory 2 convolutional code spectrums. On the bases of this spectrum it's made a quality comparison on the performance of this codes.

1. Introduction

The history of channel coding or Forward Error Correction (FEC) coding dates back to Shannon's pioneering work in which he predicted that arbitrarily reliable communications are achievable by redundant FEC coding. Convolutional FEC codes were discovered by Elias [ELI], in 1955. Due to the simplicity of the codes and the possibility to use the decoding algorithm such as SISO (Soft Input Soft Output), the convolutional codes are the most used component codes of the turbo codes.

A rate R=k/n convolutional code is an application from the semi-infinite set of binary matrix with a k line numbers towards semi-infinite set of binary matrix with a number of n lines, where $n > k$:

$$C : M_{k\times\infty} \to M_{n\times\infty} \tag{1.1}$$

Thus, using C transformation, each matrix $I \in M_{k\times\infty}$, of the form:

$$I = \begin{bmatrix} i_{01} & i_{11} & i_{21} & \cdots & i_{j1} & \cdots \\ i_{02} & i_{12} & i_{22} & \cdots & i_{j2} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ i_{0k} & i_{1k} & i_{2k} & \cdots & i_{jk} & \cdots \end{bmatrix} \qquad i_{js} \in \{0,1\} \quad \forall\, j = \overline{0,\infty},\; s = \overline{1,k} \tag{1.2}$$

is associated a matrix $V \in M_{n\times\infty}$, of the form:

$$V = \begin{bmatrix} a_{01} & a_{11} & a_{21} & \cdots & a_{j1} & \cdots \\ a_{02} & a_{12} & a_{22} & \cdots & a_{j2} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{0k} & a_{1k} & a_{2k} & \cdots & a_{jk} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{0n} & a_{1n} & a_{2n} & \cdots & a_{jn} & \cdots \end{bmatrix} \qquad a_{js} \in \{0,1\} \quad \forall\, j = \overline{0,\infty},\; s = \overline{1,n} \tag{1.3}$$

The matrix I contains the information bits, in order $i_{01}\, i_{02} \ldots i_{0k}\, i_{11} \ldots$, and the matrix V the coded sequences: $a_{01}\, a_{02} \ldots a_{0n}\, a_{11} \ldots$. The code is systematic if $a_{js} \equiv i_{js}, \forall$ j, and $\forall$ $s = 1 \div k$. Using a polynomial representation, the matrix I and V can be written as:

$$I(D) = \begin{bmatrix} \sum_{s=0}^{\infty} i_{s1} D^s \\ \sum_{s=0}^{\infty} i_{s2} D^s \\ \vdots \\ \sum_{s=0}^{\infty} i_{sk} D^s \end{bmatrix}, \quad V(D) = \begin{bmatrix} \sum_{s=0}^{\infty} a_{s1} D^s \\ \sum_{s=0}^{\infty} a_{s2} D^s \\ \vdots \\ \sum_{s=0}^{\infty} a_{sk} D^s \\ \vdots \\ \sum_{s=0}^{\infty} a_{sn} D^s \end{bmatrix} \tag{1.4}$$

With these notations, the coding relation, C, can be written as:

$$V(D) = G(D) \cdot I(D) \tag{1.5}$$

where G(D) is the code generator matrix:

1. 2 Univ. "Politehnica" Timişoara, Facultatea de ETC.
email: balta@etc.utt.ro; kmaria@etc.utt.ro.

$$G(D)=\begin{bmatrix} g_{11}(D) & g_{12}(D) & \cdots & g_{1k}(D) \\ \cdots & \cdots & \cdots & \cdots \\ g_{n1}(D) & g_{n2}(D) & \cdots & g_{nk}(D) \end{bmatrix} \quad (1.6)$$

The convolutional codes (CCs) can be classified as:
a) Systematic or non-systematic codes. If the first k lines from G(D) is the unit matrix of order k, I_k, then the codes are systematic, [HLY]. In this case the bits from I can be find again in V. With another words, in systematic codes the original information bits or symbols constitute part of the encoded codeword and hence they can be recognised explicitly at the output of the encoder. For non-systematic codes, the bits from V are linear combinations of bits from I, consequently, there aren't any information and control bits, as in previous case.
b) Recursive or non-recursive codes. If all generator polynomials that compose G(D) are finite, then the result code is non-recursive. If not, the generator polynomials $g_{js}(D)$ can be written as the form:

$$g_{js}(D)=\frac{a_{js}(D)}{b_{js}(D)} \quad (1.7)$$

where the polynomials a_{js} and b_{js} are finite. So, if there is a polynomial $b_{js}(d)\neq 1$, at least, then the code is recursive.

The constraint length, K is a very important parameter of CCs. The definition of constraint length is:

$$K = 1+\max_{j,s} \text{grad}\{a_{js}(D), b_{js}(D)\} \quad (1.8)$$

The problem of recursive and non-systematic CCs is that they can be catastrophic. The generator matrix for this type of code is:

$$G(D)=\frac{a(D)}{b(D)}\cdot G'(D); \quad a(D)\neq 1 \quad (1.9)$$

Suppose a convolution codes, rate $R=\frac{1}{2}$ and K=3 with the generator matrix:

$$G(D)=\begin{bmatrix} \dfrac{a_{01}+a_{11}D+a_{21}D^2}{1+b_{11}D+b_{21}D^2} \\ \dfrac{a_{02}+a_{12}D+a_{22}D^2}{1+b_{12}D+b_{22}D^2} \end{bmatrix}; \quad a_{js}\in\{0,1\} \quad (1.10)$$

The encoder of this code has the general scheme represented in Fig. 1. If the coefficient of this branch is 1, then there is physical connection, otherwise there is not.

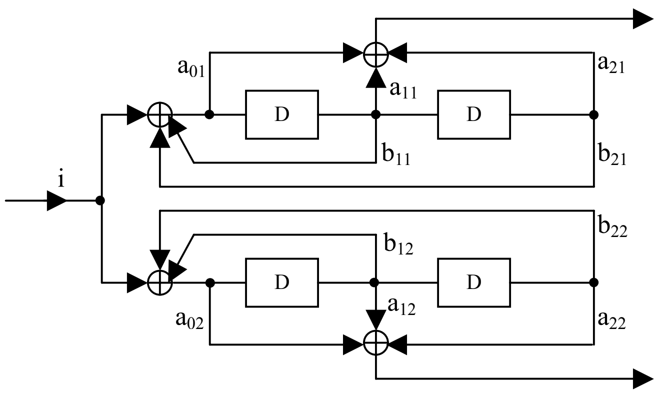


Fig. 1. Convolutional encoder with R=1/2, K=3.

We present one (representative) encoder for non-recursive systematic code (NRSC), recursive systematic code (RSC), non-recursive non-systematic code (NRNSC), $R=\frac{1}{2}$ and K =3, in Fig. 2.

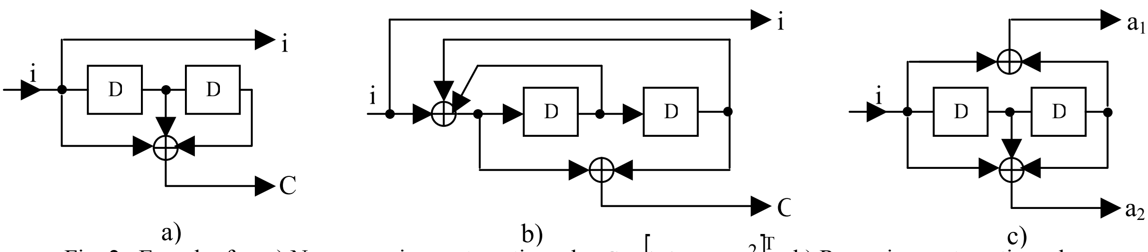


Fig. 2. Encoder for: a) Non-recursive systematic code, $G = \left[1,\ 1 + D + D^2\right]^T$; b) Recursive systematic code,

$G = \left[1,\ \dfrac{1 + D^2}{1 + D + D^2}\right]^T$; c) Non-recursive non-systematic code, $G = \left[1 + D^2,\ 1 + D + D^2\right]^T$

2. State diagram

The convolutional encoders presented in Fig. 2 can be assimilated with some finite-state machines, [DOU], characterized by the state transition diagrams. Given that there are two bits in the shift register (blocks D) at any moment, there are four possible states (00, 10. 01and 11) in the state machine and the state transitions are governed by the incoming bit *i*. So, the nodes of each diagram represent the possible states, and the labels of each branch give the corresponding output bits sequences. The state diagrams for the encoders from Fig.2 are shown in Fig.3. A state transition due to a logical zero is indicated by continuous line in the figure, while a transition activated by a logical one is represented by a dotted line.

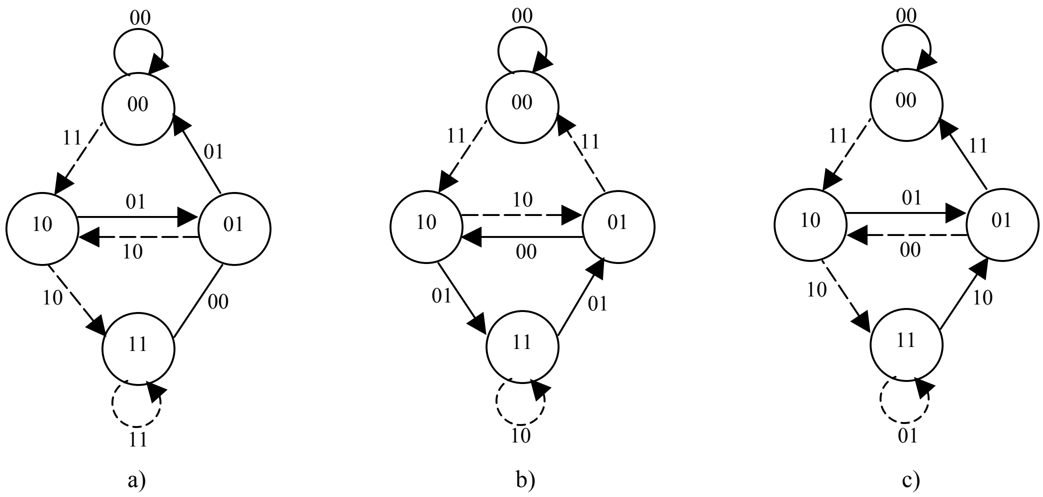


Fig. 3. The state diagrams for CCs: a) NRSC; b) RSC; c) NRNSC from Fig.2.

3. Transfer function

The practical information sequences, *I*, are finite length N, so, a convolutional code is coding a sequence of information that starts from 00 state and returns to that state, after decoding of N+K- 1 bits. To return to zero state the K-1 bits are used. So, any emissible sequence (obtained from a possible coding) corresponds to a path, made up by a succession of branches through state diagram, the path begins and ends in the zero state, [VUY]. An error of the decoder supposes the selection of another path, other than the emitted one, too. Due to linearity of the CCs (relation 1.5) the difference between the two paths (received, w, and emitted, v) obtained by the relation:

$$\varepsilon = w \oplus v \qquad (3.1)$$

is, also, a possible emissible sequence, which corresponds to a path through state diagram, from 00 to 00 states.

In others words, the paths that correspond to a minimum number of branches indicate the possibility of the error. More exactly, the weight sequences (the number of information 1s of the branches) represent the number of errors resulted. So, it's useful to find the weight spectrum of different paths, [WAD]. This spectrum will be a measure of decoder error probability.

The node 00 is split in two parts, in the starting state and finished state respectively, to find all paths that leave and return back to zero state, as well as their weights. The state diagram can be understood like a graph of nodes and transmittances. The state diagram labelled with various monomials in the letters δ, β and λ is:

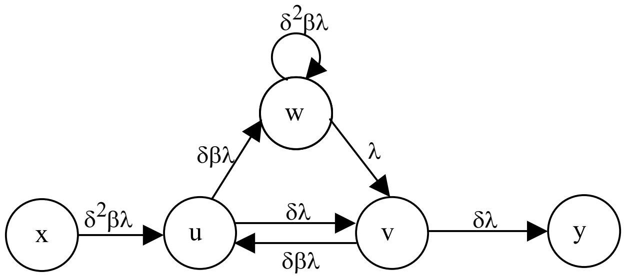


Fig. 4. State diagram labelled with distance, length, and number of input ones, of Fig.3 a) (NRSC)

The branches transmittances are products of three arguments: δ,β and λ. The exponents of these arguments indicate, [VIT], for: δ - the indeterminate associated with the Hamming weight of the encoded output sequence, β - the indeterminate associated with the Hamming weight of the information sequence (0 or 1), λ - the indeterminate associated with each branch (for a branch is 1). In this case we can write the equations:

$$\begin{aligned} u &= \delta^2\beta\lambda \cdot x + \delta\beta\lambda \cdot v \\ w &= \delta\beta\lambda \cdot u + \delta^2\beta\lambda \cdot w \\ v &= \delta\lambda \cdot u + \lambda \cdot w \\ y &= \delta\lambda \cdot v \end{aligned} \tag{3.2}$$

which describe the state transitions in Fig.4. Solving the set of equations (3.2) we find the transfer function:

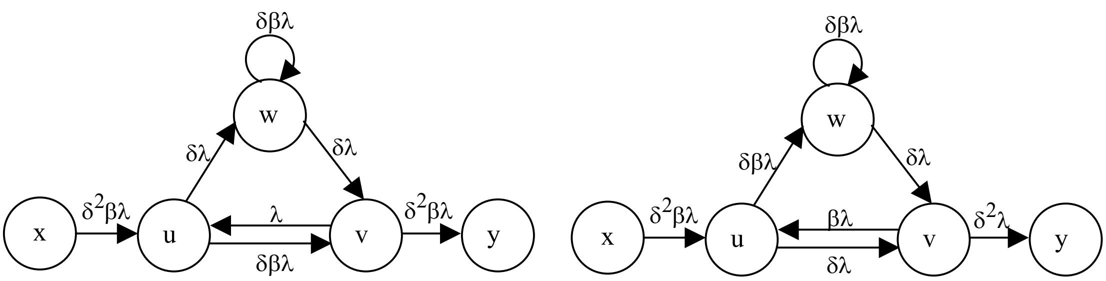
$$T_a(\alpha,\beta,\lambda) = \frac{\delta^4\beta\lambda^3\left(1-\delta^2\beta\lambda\right)+\delta^4\beta^2\lambda^4}{1-\delta^2\beta\lambda-\delta^2\beta\lambda^2-\delta^2\beta^2\lambda^3+\delta^4\beta^2\lambda^3} \tag{3.3}$$

Expanding $T_a(\delta,\beta,\lambda)$, after the powers of λ (powers that indicate the branches of a certain path), we can find:

$$T_a(\alpha,\beta,\lambda) = \delta^4\beta\lambda^3+\delta^4\beta^2\lambda^4+\delta^6\beta^2\lambda^5+\delta^6\beta^3\lambda^5+\ldots \tag{3.4}$$

The obtained result (the equation 3.4) can be understood as follows. The term $\delta^4\beta\lambda^3$ indicates a path that leaves from 00 and arrives to 00; it has 3 branches (the exponent of λ) and its weight is 4 (the exponent of δ). This path is: 00 $\rightarrow$ 10 $\rightarrow$ 01 $\rightarrow$ 00 (Fig. 3. a)). The corresponding encoded output sequence is v =110101 with the weight 4. This path has also 3 branches, one with broken line and two with continuous lines, corresponding to an input sequence i=100.

The term $\delta^4\beta^2\lambda^4$ indicates a path (the path 00 $\rightarrow$ 10 $\rightarrow$ 11 $\rightarrow$ 01$\rightarrow$00) with 4 branches and the weight 4, which correspond to the input sequence with the weight 2 (the path has two branches with broken lines). We can interpret all others terms of expanding (3.4) in the same way. In Fig. 5, we present the state diagrams labelled with various monomials in the letters δ, β and λ of the Fig.3 b, c.



a) b)

Fig. 5. State diagram labelled with distance, length, and number of input ones, of Fig. 3.b) (RSC) and 3.c) (NRNSC).

Their transfer functions are:

$$T_b(\alpha,\beta,\lambda) = \frac{\delta^5\beta^3\lambda^3-\delta^6\beta^4\lambda^4+\delta^6\beta^2\lambda^4}{1-\delta\beta\lambda-\delta\beta\lambda^2-\delta^2\lambda^3+\delta^2\beta^2\lambda^3} \tag{3.5}$$

$$T_C(\alpha,\beta,\lambda)=\frac{\delta^{54}\beta\lambda^3}{1-\delta\beta\lambda-\delta\beta\lambda^2} \quad (3.6)$$

4. Weight spectrum

Obviously the relation (3.4) will include a number of paths that increase exponentially with the weight path. In order to make the relation useful it should be reformulated in a compact manner [DOU]. This can be done by constructing a function, named weight spectrum, that gives for each value of exponential δ (weight path, Pc) the corresponding number of paths (the numbers of terms from relation (3.4)).

We present the possible weights spectrum (the column N) for CCs with rate R=1/2 and K=3, in table 1, except the RNSC ones. The column Sp, gives for all the N paths, with the same Pc, the weights of the input sequences.

If, there are errors after decoding, the erroneous decode sequence is a wordcode, conformable to the relation (3.1). In another words, the spectrum distances give a measure of the errors probability. When the Signal/Noise Ratio (SNR) is high enough, the most probable errors words are to be looked for in the paths with small weights. If, instead, for small SNR, the errors can arise in bigger groups, then, we should take in consideration the paths with bigger weight (respectively, the bottom section of Table 1, that corresponds to the high value of Pc).

The comparison in the first basis of Table 1 can be made from the point of view of the minimum distance code. So, there are two codes with minimum distance d_{min}=3, the NRSC [1,5] and RSC[1,1/5], five codes with d_{min}=4 (NRSC [1,7], RSC[1,7/3], RSC[1,1/7], RSC[1,3/7] and NRNSC [3,7]), three codes with d_{min}=5: RSC[1,7/5], RSC[1,5/7] and NRNSC[5,7]. We noted in parenthesis the generator matrix in octal. For examples, to $G(D)=[1,1/1+D^2]$ correspond G=[1,1/5], in octal. The superior minimum distance of the last three code, specified above indicates a superiority from the point of view of the correction capacity, fact that it will be demonstrated, especially, for the great SNR, where the words (the paths) with small weights are important.

Table 1

Cod	NRSC [1,5] (1)		NRSC [1,7] (2)		RSC [1,7/3] (3)		RSC [1,1/5] (4)		RSC [1,7/5] (5)	
Pc	N	Sp	N	Sp	N	Sp	N	Sp	N	Sp
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	1	2	0	0
4	1	2	2	3	1	2	3	6	0	0
5	1	3	0	0	2	4	5	10	1	2
6	2	6	5	15	2	6	8	18	2	6
7	4	14	0	0	5	18	12	29	4	14
8	7	30	13	58	8	32	19	49	8	32
9	11	57	0	0	13	62	31	84	16	72
10	17	102	34	201	24	128	51	145	32	160
11	27	181	0	0	40	236	81	239	64	352
12	44	324	89	655	69	452	130	401	128	768
13	72	580	0	0	120	856	210	678	256	1664
14	117	1028	233	2052	205	1586	341	1151	512	3584
15	189	1801	0	0	354	2956	553	1944	1024	7680
16	305	3130	610	6255	610	5458	885	3218	2048	16384
Cod	RSC [1,1/7] (6)		RSC [1,3/7] (7)		RSC [1,5/7] (8)		NRNSC [3,7] (9)		NRNSC [5,7] (10)	
Pc	N	Sp	N	Sp	N	Sp	N	Sp	N	Sp
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	2	5	1	2	0	0	1	2	0	0
5	0	0	2	6	1	3	2	4	1	1
6	5	15	2	6	2	6	2	8	2	4
7	0	0	5	17	4	14	5	21	4	12
8	13	46	8	3	8	32	10	48	8	32
9	0	0	13	55	16	72	15	87	16	80
10	34	139	24	112	32	160	28	188	32	192
11	0	0	40	204	64	352	54	394	64	448
12	89	413	69	376	128	768	85	698	128	1024
13	0	0	120	704	256	1664	146	1350	256	2304
14	233	1210	205	1284	512	3584	269	2664	512	5120
15	0	0	354	2354	1024	7680	460	4906	1024	11264
16	610	3505	610	4302	2048	16384	770	9008	2048	24576

where: Pc - the path weight; N – the numbers of paths with Pc weight, Sp – the weight sum of the information sequences corresponding to N paths.
We present in Fig. 6 the distance spectrums of RSC[1,5/7]:

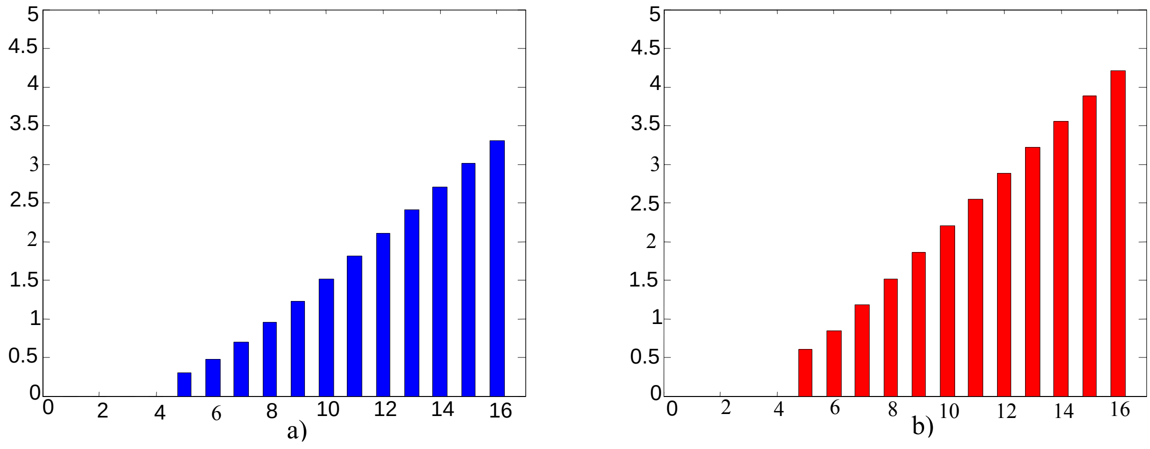


Fig.6 Distance spectrums for RSC[1,5/7]: a) log(N+1) / Pc; b) log(Sp+1) / Pc.

Another remark is the resemblance of spectrum codes RSC[1,7/5] and RSC[1,5/7]. This means that these codes will give the same performances.

On the basis of the previous remarks and because the function Sp(5) takes the value 1 for NRNSC[5,7] and 2 for RSC[1,7/5] we can conclude that the most performant code of the big signal/noise ratios will be NRNSC[5,7], [FOR]. For the small SNR the paths with big weights are valuable. In another words, here we recommend the codes NRSC[1,7] and RSC[1,1/7] or even NRSC[1,5] and RSC[1,3/7].

5. Conclusions and perspectives

The paper presents an analogy between CCs with rate R=1/2 and constraint length K=3, on the basis of the weight spectrum. All encoders have memory 2 (the same numbers of cellules, two) and there are four possible states for each of them. Because the convolutional encoders have the same numbers of states (four) they have the same complexity. The comparison shows that, for big signal/noise ratios, the non-recursive codes, are better then the recursive ones. The best weight spectrum was found for NRNSC[5,7]. The best recursive codes are RSC[1,7/5] and RSC[1,5/7] that have the same performances.

This analysis can be extended at CCs with another coding rate and another constraint length.

The analysis can be continued in the study of the performances of turbo-codes.

Acknowledgement

The authors want to thanks their Ph.D. director, Professor Miranda Nafornită, for the suggestion to write this paper and for her continuous help.

References

[ELI] P. Elias, „Coding for noisy channels", IRE Convention Record, pt.4, pp.37-47, 1955

[FOR] G.D. Forney Jr, "Convolutional Codes I: Algebraic Structure", IEEE Transaction on Information Theory", nov.1970, pp.720-738

[HLY] L.Hanzo, T.H.Liew, B.L.Yeap, "Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels", John Wiley & Sons Ltd, England, 2002

[DOU] C.Douillard, „Turbo codes (convolutifs)", seminar Timişoara, 15-18 martie 2004

[VUY] B.Vucetic, J.Yuan, "Turbo Codes Principles and Applications", Kluwer Academic Publishers, USA, 2000

[VIT] A.J. Viterbi, "CDMA. Principles of Spread Spectrum Communication", Addison-Wesley Publishing Company

[WAD] G.Wade, "Coding Techniques, An Introduction to Compression and Error Control", Creative Print and Design, Ebbw Vale, Great Britain, 2000