BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LVII (LXI), Fasc. 6, 2011
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

A WEIGHTS DISTRIBUTION COMPARISON FOR THE RECURSIVE SYSTEMATIC DUO-BINARY CONVOLUTIONAL CODES USED IN TURBO-CODES

BY

HORIA BALTĂ*, MARIA KOVACI, A. ISAR, MIRANDA NAFORNIŢĂ and MARIA BALTĂ

“Politehnica” University of Timişoara

Received, May 31, 2011
Accepted for publication: July 25, 2011

Abstract. A comparative study on the code distances spectrum of recursive and systematic duo-binary convolutional codes (RSDBCC) is presented, which led to the construction of good turbo codes. For the selection of the RSDBCC, which represents the object of the present survey, we have used the convergence criterion. The restriction of the study at a limited number of codes is a consequence of the huge number of possibilities. The purpose of this comparative study, realized on the best codes, was the identification of the statistical characteristics which justify good performance.

Key words: convolutional code; weight spectrum; transfer function; turbo-codes.

1. Introduction

The history Forward Error Correction Coding (FECC) dates back to Shannon’s pioneering work in which he predicted that arbitrarily reliable communications are achievable by redundant FECC. Convolutional codes (CC), this proceeding being discovered by Elias (1955). Due to the simplicity of the

*Corresponding author: *e-mail*: horia.balta@etc.upt.ro

codes and the possibility to use decoding algorithm such as SISO (Soft Input Soft Output), the CC are the most used component codes of the turbo codes.

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

$$\mathcal{C} : M_{k\times\infty} \to M_{n\times\infty}. \tag{1}$$

Thus, using the $\mathcal{C}$ transformation, for each matrix $U \in M_{k\times\infty}$, of the form

$$U = \begin{bmatrix} u_{10} & u_{11} & \dots & u_{1j} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ u_{k0} & u_{k1} & \dots & u_{kj} & \dots \end{bmatrix}, \ u_{sj} \in \{0,\ 1\},\ \forall j = \overline{0,\ \infty},\ s = \overline{1,\ k}, \tag{2}$$

is attached a matrix $V \in M_{n\times\infty}$, having the form

$$V = \begin{bmatrix} v_{10} & v_{11} & \dots & v_{1j} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ v_{n0} & v_{n1} & \dots & v_{nj} & \dots \end{bmatrix}, \ v_{sj} \in \{0,\ 1\},\ \forall j = \overline{0,\ \infty},\ s = \overline{1,\ n}. \tag{3}$$

The matrix U contains the information bits and the matrix V the coded sequences. Using a polynomial representation, the encoding rule is

$$V(D) = G(D) \cdot U(D), \tag{4}$$

where

$$U(D) = \left[\sum_{j=0}^{\infty} u_{1j} D^j \quad \sum_{j=0}^{\infty} u_{2j} D^j \quad \dots \quad \sum_{j=0}^{\infty} u_{kj} D^j \right]^T, \tag{5}$$

$$V(D) = \left[\sum_{j=0}^{\infty} v_{1j} D^j \quad \sum_{j=0}^{\infty} v_{2j} D^j \quad \dots \quad \sum_{j=0}^{\infty} v_{kj} D^j \quad \dots \quad \sum_{j=0}^{\infty} v_{nj} D^j \right]^T \tag{6}$$

and

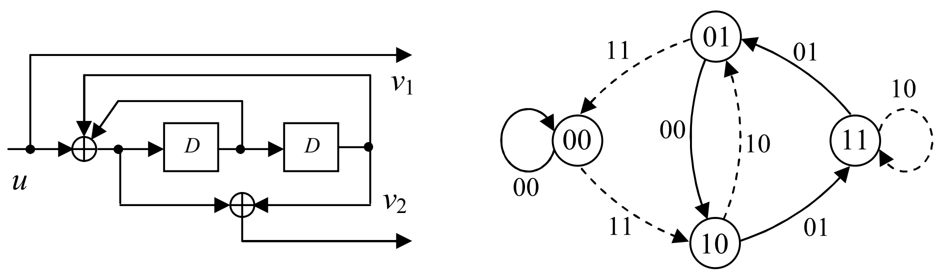
$$G(D) = \begin{bmatrix} g_{11}(D) & g_{12}(D) & \dots & g_{1k}(D) \\ \dots & \dots & \dots & \dots \\ g_{n1}(D) & g_{n2}(D) & \dots & g_{nk}(D) \end{bmatrix} \tag{7}$$

is the generator matrix of the code.

The code is *systematic* if $v_{js} = u_{js}, \forall j$, and $\forall s = \overline{1,k}$. If at least one generator polynomial that compose $G(D)$ is of infinite degree, then the corresponding code is *recursive*. The turbo-codes use almost exclusively recursive systematic convolutional codes due to their superior performance.

1.1. State Diagram

A convolutional encoder can be assimilated with some finite-state machines, (Viterbi, 1971), which can be characterized by the state transition diagrams. We define the encoder state diagram with the aid of the example in Fig. 1. Given that there are two bits in the shift register (denoted by D) at any moment, there are four possible states (00, 10, 01 and 11) in the state machine and the state transitions are governed by the incoming bit, u. So, the nodes of each diagram represent the possible states, and the labels of each branch give the corresponding output bits sequences. The state diagram for the encoder from Fig.1 *a* is shown in Fig. 1 *b*. A state transition due to a logical zero is indicated by a continuous line in Fig. 1 *b*, while a transition activated by a logical one is represented by a broken line.



a *b*

Fig. 1 – *a* – The implementation of the recursive systematic code (RSC) with the generator matrix $G = \left[1,\ \dfrac{1+D^2}{1+D+D^2}\right]^T$; *b* – the corresponding state diagram.

1.2. Transfer function

The practical information sequences U has finite length, N. A CC encodes a sequence of information that starts from the state 00 and ends to the same state. So, any encoded sequence corresponds to a path, made up by a succession of branches through the state diagram which begins and ends in the zero state (Viterbi, 1971). An error of the decoder supposes the selection of another path different from the corresponding path from the encoder. Due to the linearity of the CCs (eq. (1)) the difference between the two paths (from the encoder and from the decoder) can be an admissible sequence, which corresponds to a path through the state diagram, from the 00 state toward the 00 states. In other words, the paths that correspond to a minimum number of branches indicate the possibility of the error. More exactly, the weights sequences (the number of information bits equal with 1 of the branches) represent the number of errors resulted. So, it is useful to find the weight spectrum of different paths (Viterki, 1971). This spectrum will be a measure of the decoder error probability. The node 00 is split in two parts, in the starting state and finishing state respectively, to find all paths that leave and return back to the zero state, as well as their weights. The state diagram can be understood as a graph of nodes and transmittances. For example the state diagram in Fig. 1 *b* is equivalent with the state diagram represented in Fig. 2, where we have used the following denotations: δ – the Hamming weight of the encoded output sequence, β – the Hamming weight of the information sequence and λ – the length of the branch.

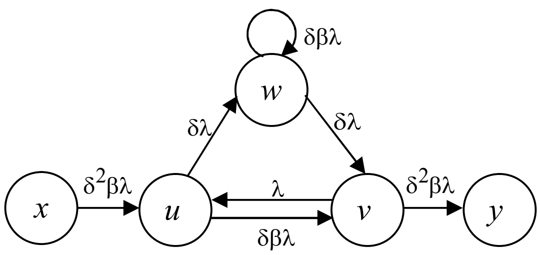


Fig. 2 – State diagram labeled with distance, length, and number of input ones, of Fig.1 *a*.

Using the rule of Mason (1966) we can find out the transfer function of the graph in Fig. 2

$$T(\delta,\beta,\lambda)=\frac{\delta^5\beta^3\lambda^3+\delta^6\beta^2\lambda^4-\delta^6\beta^4\lambda^4}{1-\delta\beta\lambda-\delta\beta\lambda^2-\delta^2\lambda^3+\delta^2\beta^2\lambda^3}. \quad (8)$$

Developing T following the powers of δ (powers that indicate the weight of a certain path), we can find

$$T(\delta,\beta,\lambda)=\delta^5\beta^3\lambda^3+\delta^6\beta^2\lambda^4+\delta^6\beta^4\lambda^5+2\delta^7\beta^3\lambda^5+\delta^7\beta^3\lambda^6+\ldots \quad (9)$$

The obtained result in the eq. (9) can be understood as follows. The term $\delta^5\beta^3\lambda^3$ indicates a path that leaves from 00 and arrives to 00; it has three branches (the exponent of λ) and its weight is 5 (the exponent of δ). This path is: 00 → 10 → 01 → 00 (Fig. 1 *b*). The corresponding encoded output sequence is v = 110111 with the weight 5. This path has also three branches, all represented with broken line in Fig. 1 *b*, corresponding to an input sequence u = 111. We can interpret all others terms of the development in eq. (9) in the same way.

1.3. Weights Spectrum

The relation (9) includes a number of paths that increases exponentially with the weight path. This relation may become useful if is expressed put in a compact form. This can be done by constructing a function, named *weights spectrum*, which gives for each value of the exponent δ (weight path, *Pw*) the corresponding number of paths, *i.e*. the numbers of terms in eq. (9).

In Table 1 we present the weights spectrum of the RSC encoder defined in Fig. 1. *Nw* indicates the total number of path having the weight *Pw*. *Iw* represents the sum of the weights of information sequences that correspond to the *Nw* paths, and *Lw* is the maximal length of a path (between the *Nw* path).

Table 1 shows that there is a path with a length *Lw* = 3, having a weight *Pw* = 3 and corresponding to an input sequence with the weight *Iw* = 3. The weight of this path defines the minimum code distance, $d_{\min}$. Table 1 also shows the two paths of weight 6, which have input sequences with added weights equal with 6 = 2 + 4, the longest having the length 5, and so on.

Table 1
Weights Spectrum for the Encoder Defined in Fig. 1

Pw weight	*Nw* number of path-word	*Iw* total information weight for all words	*Lw* max. word length
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	1	3	3
6	2	6	5
7	4	14	7
8	8	32	9
9	16	72	11
10	32	160	13
11	64	352	15
12	128	768	17
13	256	1,664	19
14	512	3,584	21
15	1,024	7,680	23
16	2,048	16,384	25

2. The RSDBC Code of Rate 2/3

Fig. 3 shows the general scheme of a recursive and systematic convolutional encoder with *r* inputs, known in the literature (Johannesson & Zigangirov, 1999), as a canonical form „observer".

Using the denotations from Fig. 3 and the *D* transform ($g_i(D)=\sum_{j=0}^{M} g_{ij}D^j$

and $C(D)=\sum_{n=0}^{\infty} c^n D^n$), it can be shown that the transfer function matrix, $G(D)$, of the encoder is given by the following eq. (Johannesson & Zigangirov, 1999):

$$G(D)=\begin{bmatrix} 1 & \dots & 0 & 0 \\ \dots & \dots & 1 & 0 \\ 0 & \dots & 0 & 1 \\ \frac{g_r(D)}{g_0(D)} & \dots & \frac{g_2(D)}{g_0(D)} & \frac{g_1(D)}{g_0(D)} \end{bmatrix}, \quad (10)$$

so that $V(D)=[U(D)\ ;\ C(D)]$, where: $U(D)=[u_r(D)\ \dots\ u_2(D)\ u_1(D)]^T$. For the sake of simplicity, further we will use for the generator matrix the denotation $G=[g_r\ \dots\ g_1\ g_0]$, where the numbers g_i, (i = 0,…r), represent the decimal transposition of the binary sequence $[g_{i,m}\ \dots\ g_{i,1}\ g_{i,0}]$.

In this paper we will investigate the RSCC with r = 2 inputs (duo binary). So, for this family of codes, the generator matrix that identifies a particular encoder is of the form

$$G=[g_2\quad g_1\quad g_0], \quad (11)$$

where g_i, (i = 0, 1 or 2), are decimal numbers with values between 1 and $2^{m+1}-1$, m being the memory of the encoder.

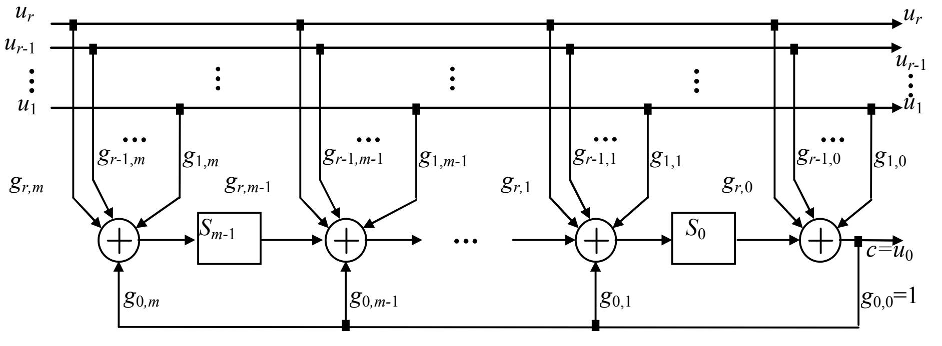


Fig. 3 – The general structure of a recursive and systematic multi-input convolutional encoder, with $r/(r+1)$ rate: the observer canonical form.

2.1. The Weights Spectrum of the Memory 2 RSDBC

Unlike single input convolutional codes, the state diagram for the duo-binary convolutional codes has a higher number of branches. More specifically, for these codes in each node of the state diagram leave and enter $2^{r} = 4$ branches. Consequently, their transfer functions have much more terms. These transfer functions and the corresponding weights spectra can be calculated only using dedicated programs.

Table 2
Weights Spectra for the Memory 2 RSDBC Encoders

	G_1=[7 3 5]			G_2=[7 6 5]			G_3=[5 3 7]			G_4=[6 5 7]		
Pw	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	1	2	2	1	2	2	1	3	2	1	3	2
4	5	14	4	6	17	4	4	10	4	5	14	4
5	18	58	6	21	69	6	15	49	6	18	62	6
6	56	211	8	64	244	8	45	176	8	53	218	8
7	162	704	10	183	803	10	132	610	10	150	720	10
8	474	2,348	12	532	2,655	12	388	2,056	12	440	2,404	12
9	1,379	7,675	14	1,553	8,709	14	1,133	6,769	14	1,284	7,876	14
10	4,032	24,923	16	4,541	28,257	16	3,311	22,010	16	3,754	25,552	16
11	11,784	80,077	18	13,273	90,741	18	9,672	70,804	18	10,971	82,059	18
12	34,446	255,218	20	38,798	289,056	20	28,271	225,980	20	32,069	261,450	20
13	100,685	807,799	22	113,406	914,517	22	82,635	716,145	22	93,737	827,305	22
14	294,303	2,541,852	24	331,486	2,876,610	24	241,543	2,255,878	24	273,994	2,602,642	24
15	860,247	7,957,872	26	968,933	9,003,086	26	706,030	7,069,150	26	800,884	8,146,582	26

	G_5=[5 1 7]			G_6=[5 4 7]			G_7=[7 1 5]			G_8=[3 2 7]		
Pw	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	5	3	2	5	3	2	5	3	3	7	3
4	6	19	5	6	19	5	6	19	5	5	13	4
5	18	64	7	17	61	7	17	61	7	11	32	6
6	53	221	9	52	218	9	52	218	9	32	115	7
7	144	690	11	140	675	11	140	675	11	67	270	9
8	402	2165	13	393	2,123	13	393	2,123	13	170	770	10
9	1,119	6,,710	15	1,092	6,569	15	1,092	6,569	15	398	2,015	12
10	3,113	20,560	17	3,039	20,128	17	3,039	20,128	17	939	5,196	13
11	8,669	62,521	19	8,462	61,182	19	8,462	61,182	19	2,251	13,612	15
12	24,137	188,759	21	23,561	184,686	21	23,561	184,686	21	5,319	34,825	16
13	67,204	566,420	23	65,600	554,101	23	65,600	554,101	23	12,653	89,118	18
14	187,119	1,690,885	25	182,653	1,653,868	25	182,653	1,653,868	25	30,067	226,941	19
15	521,001	5,024,792	27	508,566	4,914,163	27	508,566	4,914,163	27	71,370	574,276	21

	G_9=[3 1 7]			G_{10}=[4 3 5]			G_{11}=[7 4 5]			G_{12}=[6 4 7]		
Pw	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*	*Nw*	*Iw*	*Lw*
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	3	7	3	3	7	3	2	4	3	3	7	3
4	6	16	4	10	28	5	7	18	5	8	23	4
5	16	51	6	27	86	7	18	54	7	17	57	6
6	41	153	7	74	274	9	54	200	9	47	181	7
7	92	388	9	192	809	11	145	614	11	105	462	9
8	230	1,089	10	505	2,408	13	407	1,980	13	257	1,252	10
9	534	2,797	12	1,325	7,061	15	1,132	6,188	15	607	3,271	12
10	1,272	7,284	13	3,489	20,555	17	3,150	19,132	17	1,435	8,441	13
11	3,031	18,900	15	9,184	59,245	19	8,772	58,616	19	3,423	21,846	15
12	7,174	48,299	16	24,179	169,492	21	24,424	178,046	21	8,113	55,862	16
13	17,070	123,457	18	63,647	481,736	23	68,003	537,064	23	19,278	142,333	18
14	40,534	313,519	19	167,540	1,361,778	25	189,344	1,610,476	25	45,808	361,204	19
15	96,261	792,600	21	441,021	3,831,308	27	527,196	4,804,556	27	108,771	912,084	21

In Table 2 are presented the weights (distances) spectra for the best 12 memory two RSDBC encoders. The ranking criterion used was the convergence of the turbo decoders formed with these codes (Baltă *et al*., 2010). An initial assessment is that all the spectra presented in Table 2 have $d_{\min} = 3$, a lower value than the corresponding value in Table 1.

3. Conclusions

Comparing the spectra from Table 2 we can observe the following aspects:

1. The encoders with the „poorest" spectrum at small weight have better performance. The encoders G_1, G_2, G_3 and G_4 generate a single path with the weight of 3. For G_1 and G_2 this path corresponds to an input sequence with a weight of 2.

2. At higher weights, "good" spectra can become rich. This will lead to a reversal of the encoders' hierarchies at small SNRs, where these weights are important (the probability of the error of the words with greater weights is increasing at low SNRs).

Based on previous remarks, it is possible to formulate an opinion about the quality of an RSDBCC encoder based on the analysis of its distances spectrum. We will continue in the future the analysis of the weights spectrum of the RSDBCC encoders with memory bigger than two.

Acknowledgments. This paper was supported by the project "Development and Support of Multidisciplinary Postdoctoral Programmes in Major Technical Areas of National Strategy of Research – Development – Innovation" 4D-POSTDOC, contract no. POSDRU/89/1.5/S/52603, project co-funded by the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013.

REFERENCES

Baltă H., Kovaci M., Nafornită M., Baltă M., *Multi-Binary Turbo-Code Design Based on Convergence of Iterative Turbo-Decoding Process*. 5th Europ. Conf. on Circ. a. Syst. for Commun. (ECCSC'10), Nov. 23–25, 2010, Belgrade, Serbia.

Elias P., *Coding for Noisy Channels*. IRE Conv. Record, pt.4, 1955, 37-47.

Johannesson R., Zigangirov K. Sh., *Fundamentals of Convolutional Coding*. J. Wiley, IEEE Press, NY, 1999.

Mason S. J., *Feedback Theory - Further Properties of Signal Flow Graphs*. Proc. of the IRE, **44**, *7* , 920-926 (1956).

Viterbi A. J., *Convolutional Codes and their Performance in Communication Systems*. IEEE Trans. Comm., **COM-19**, 751-772 (1971).

COMPARAȚIE ÎNTRE DISTRIBUȚIA PONDERILOR CONVOLUȚIONALE, RECURSIVE ŞI SISTEMATICE, DUO-BINARE ÎN TURBO-CODURI

(Rezumat)

Se prezintă rezultatele unui studiu comparativ asupra spectrului distanțelor pentru codurile convoluționale, recursive şi sistematice, duo-binare, ce contribuie la construcția unor turbo-coduri cu performanțe superioare. Selecția codurilor ce constituie obiectul studiului prezentat a fost efectuată utilizând criteriul convergenței procesului iterativ de turbo-decodare. Restricția studiului asupra unui număr limitat de coduri este o consecință a numărului foarte mare de coduri posibile. Scopul acestui studiu a fost identificarea unor caracteristici statistice ale spectrelor de distanță care să justifice performanțele opținute.