Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 2, 2004

The Performances of Convolutional Codes used in Turbo Codes

Horia Baltă[1], Maria Kovaci[2]

Abstract – In this paper are presented and compared the BER performances obtained by the simulation of a transmission system, which utilizes the forward error correcting by codes concatenation and iterative decoding (turbo coding). There have been investigated all the systematic convolutional codes having the constraint length K less or equal to 6, under three diffrent concatenated forms: parallel PCCC (pure turbo code), serial SCCC and hybrid HCCC, at the following rates: 1/3, 1/4 and 1/3, respectively, all unpunctured.

Two interleaver types were used: pseudo-random and S-interleaver having the same length, 1784.

The AWGN channel and the BPSK modulation were employed.

The used iteration number was eight. For increasing the work speed an iterations stop criterion was used. When the resulting error number from the decoding of a data block is zero, the remaining iterations are not effectuated, passing to the next block.

For decoding, the MAP, MaxLogMAP and Log MAP algorithms were used. In all the cases, a tail off was em lo ed for the first code, with the decreasin transmission rate price.

The transmitted data block numbers for a simulation were chosen in function of the signal to noise ratio, SNR, i.e. to keep a good precision for obtained curve.

Keywords: convolutional codes, turbo codes, interleaver, iteratio .

I. INTRODUCTION

Two procedures which improve the performances of convolutional codes, CCs, (and block codes), from point of view of error rate (BER), are concatenation and iterative decoding.

Fig.1 illustrates the possible ways of convolutional codes concatenation. Concatenated convolutional schemes tend to fall into three categories: parallel concatenated convolutional codes, PCCCs, (as in Fig.1 a)), serial concatenated convolutional codes, SCCC, (as in Fig.1 b)) and hybrid concatenated convolutional codes, HCCC, (as in Fig.1 c)).

The PCCC (or Turbo code) was introduced by Berrou *et al.* [1], in 1993, and it was the beginning of turbo code era.

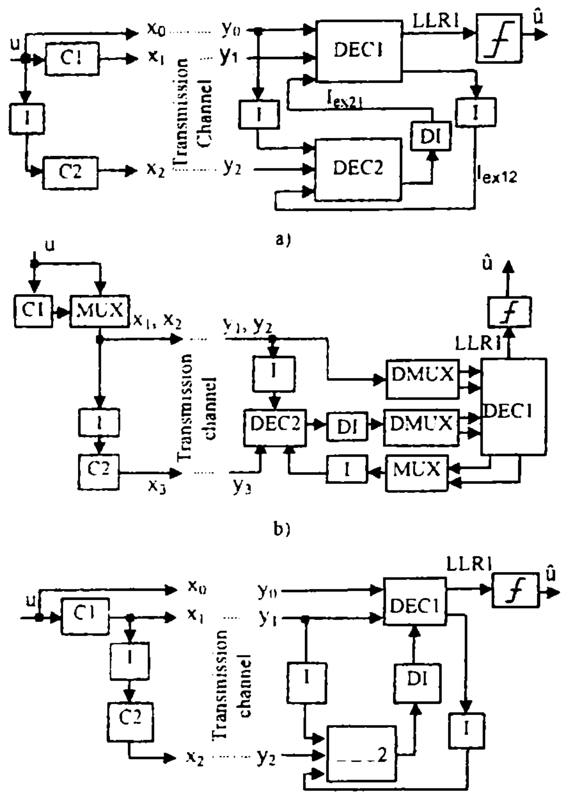


Fig 1 a) Parallel concatenated convolutional codes, b) Serial concatenated convolutional codes, c) Hybrid concatenated convolutional codes

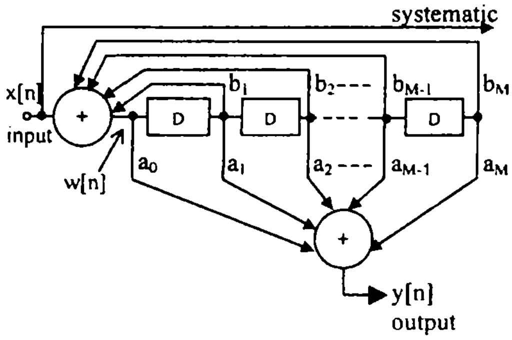
Blocks I and DI realize interleaving and deinterleaving functions. We used in this paper two interleavers: pseudo-random [2] and S-interleaver [3]. Our simulations prove that the interleavers have an essential influence on performances of Turbo codes.

DEC1 and DEC2 are iterative decoder blocks [4], which implement algorithms like: the MAP algorithm, the first and the must important, proposed by Bahl and *al.* [5], the MAXLogMAP algorithm [6], and the Log MAP algorithm, proposed by Robertson and *al.* [7].

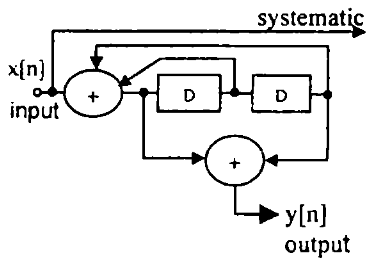
The constitutive codes can be convolutional codes or block codes. In this paper we studied the first exclusively. The general scheme of a recursive

[1,2] Facultatea de Electronică şi Telecomunicaţii, Departamentul Comunicaţii Bd. V. Pârvan Nr. 2, 300223 Timişoara, balta@etc.utt.ro; kmaria@etc.utt.ro

systematic convolutional code, RSC, is shown in Fig. 2 a) and an example of RSC is shown in Fig. 2 b).



a)



b)

Fig.2 Recursive systematic convolutional code: a) general scheme, b) example

The following ecuations result:

$$y[n]=\sum_{k=0}^{M} a_k w[n-k] \tag{1}$$

$$w[n]=\sum_{k=0}^{M} b_k w[n-k]+x[n] \tag{2}$$

$$Y(D)=\left(\sum_{k=0}^{M} a_k D^k\right)\cdot W(D) \tag{3}$$

$$W(D)\cdot \sum_{k=0}^{M} b_k D^k = X(D) \tag{4}$$

$$\frac{Y(D)}{X(D)}=\frac{\sum_{k=0}^{M} a_k D^k}{\sum_{k=0}^{M} b_k D^k} \tag{5}$$

The code generator matrix, G(D), is:

$$G(D)=\left[1,\frac{a(D)}{b(D)}\right] \tag{6}$$

The two polynomials attached, a and b, define in totality the convolutional code. The maximum degree of a and b polynomials give the coder's memory (.q..l wi... K-1) .nd i. is . ....asure of .he complexity and of the volume of the effectuated computation of each component decoder. It increases exponentially with K or M [6].

Practically, convolutional codes with K=3÷6 are used. Table 1 shows the generator polynomials with degree inferior or equal with 5 which can be selected like a or b. Because of the restriction that a and b to be prime the table shows also the possible divisors for each polynomial.

Table1

Degree	Ireductibil	Primitive	Polynomials in octal and their divisors								
0	*		1	1	0	0	0	0	0	0	0
1	*	*	3	1	0	0	0	0	0	0	0
2			5	1	3	0	0	0	0	0	0
	*	*	7	1	0	0	0	0	0	0	0
3			11	1	3	7	0	0	0	0	0
	*	*	13	1	0	0	0	0	0	0	0
	*	*	15	1	0	0	0	0	0	0	0
			17	1	3	0	0	0	0	0	0
4			21	1	3	0	0	0	0	0	0
	*	*	23	1	0	0	0	0	0	0	0
			25	1	0	7	0	0	0	0	0
			27	1	3	0	0	15	0	0	0
	*	*	31	1	0	0	0	0	0	0	0
			33	1	3	7	0	0	0	0	0
			35	1	3	0	13	0	0	0	0
	*		37	1	0	0	0	0	0	0	0
5			41	1	3	0	0	0	0	0	37
			43	1	0	7	0	15	0	0	0
	*	*	45	1	0	0	0	0	0	0	0
			47	1	3	0	13	0	0	0	0
	*	*	51	1	0	0	0	0	0	0	0
			53	1	3	0	0	0	0	31	0
			55	1	3	7	0	0	0	0	0
	*	*	57	1	0	0	0	0	0	0	0
			61	1	0	7	13	0	0	0	0
			63	1	3	0	0	0	0	0	0
			65	1	3	0	0	0	23	0	0
	*	*	67	1	0	0	0	0	0	0	0
			71	1	3	0	0	15	0	0	0
	*	*	73	1	0	0	0	0	0	0	0
	*	*	75	1	0	0	0	0	0	0	0
			77	1	3	7	0	0	0	0	0

II. EXPERIMENTAL RESULTS

Table 2 and Table 3 present the simulation results obtained with parallel concatenated convolutional code, Fig.1 a), rate R=1/3, RSC, with the pseudo-random interleaver (table 2), and S-interleaver (table 3).

Table 2 (pseudo-random interleaver)

	1	3	5	7	11	13	15	17	21	23	25	27	31	33	35	37
1	0	2954035	3021300	1935725	2864349	1188340	1177130	1093049	3157698	1073326	1821748	710482	1242932	585061	560538	704876
3	3612356	0	0	1358637	0	527925	498583	0	0	243060	156083	0	318487	0	0	392376
5	4516335	0	0	100348	0	61305	62825	0	0	5311	1471412	0	5259	0	0	20042
7	1262331	919632	5710	0	0	5163	6663	378923	1419	1445	0	8216	1916	0	5840	289992
11	4684497	0	0	0	0	37148	37295	0	0	47924	0	0	37591	0	0	437
13	4684497	33594	6755	7043	1720	0	312	292	220	3144	243	8839	265	1192	0	155
15	108671	34188	9633	6457	1294	235	0	403	204	579	313	0	2910	1077	7302	229
17	445882	0	0	230633	0	9302	9040	0	0	8749	1741	0	10009	0	0	249017
21	4886023	0	0	13579	0	11594	11975	0	0	176919	159130	0	173697	0	0	8037
23	178171	20128	1102	521	2061	1629	230	691	6165	0	578	476	1032	1283	3005	586
25	1420029	5267	966445	0	0	751	1159	926	86082	34388	0	3327	1676	0	1904	1999
27	21498	0	0	6201	0	12060	0	0	0	345	1105	0	373	0	0	546
31	429482	6548	1855	600	2888	418	1823	344	10420	3162	207	828	0	776	534	1443
33	10949	0	0	0	0	2602	2471	0	0	417	0	0	450	0	0	1374
35	10609	0	0	6446	0	0	9675	0	0	241	558	0	368	0	0	270
37	135016	123269	1379	114297	1445	236	449	104996	956	757	10480	1418	737	901	731	0
41	4107943	0	0	2430	0	1991	1651	0	0	2242	2031	0	1643	0	0	0
43	424030	8621	7204	0	0	456	0	2488	6217	551	0	0	2140	0	6606	3117
45	299338	13935	5521	1638	3316	901	555	6442	1873	12670	4607	854	3971	5999	4711	13557
47	38916	0	0	10547	0	0	3771	0	0	1679	9612	0	13089	0	0	6179
51	205797	7216	6101	1502	3468	678	630	2024	1006	3910	5735	3119	17711	5515	3467	15763
53	9846	0	0	599	0	2125	6769	0	0	962	681	0	0	0	0	1890
55	16898	0	0	0	0	3796	3708	0	0	1940	0	0	13015	0	0	16663
57	5160	1513	2592	2879	913	748	4673	879	4271	1351	1768	8903	30764	20257	3587	2084
61	510340	12737	2421	0	0	0	607	721	6592	4919	0	18487	744	0	0	5890
63	125731	0	0	38990	0	29225	38412	0	0	39576	43539	0	36760	0	0	38388
65	28245	0	0	698	0	2550	3709	0	0	0	2436	0	2649	0	0	1497
67	2075	8583	2910	2383	8190	355	2117	607	22152	26281	20200	1009	2018	1191	3883	11581
71	21641	0	0	7941	0	18221	0	0	0	4869	8983	0	945	0	0	2437
73	2587	6523	3030	4425	12835	5061	471	313	114070	997	16231	1540	12697	977	424	3386
75	9336	4653	11238	1249	968	12662	961	2685	6851	104145	2325	7577	781	16290	1835	450
77	94043	0	0	0	0	1404	2169	0	0	1751	0	0	2123	0	0	73523

	41	43	45	47	51	53	55	57	61	63	65	67	71	73	75	77
1	2558092	1291785	1193479	523168	1207492	480732	644618	277521	1286689	589732	508488	246833	586229	251741	312032	409436
3	0	145046	50884	0	41916	0	0	31688	119502	0	0	137957	0	139590	32654	0
5	0	6347	50865	0	52682	0	0	20946	5037	0	0	1560	0	1861	23048	0
7	348	0	262	13007	216	329	0	737	0	257	363	18076	11557	14284	351	0
11	0	0	43879	0	44683	0	0	401	0	0	0	416	0	445	662	0
13	7029	416	199	0	250	1679	1008	195	0	392	242	153	215	1689	265	424
15	1396	0	198	122	151	482	1639	499	370	282	1975	2524	0	466	235	359
17	0	8663	1597	0	5878	0	0	8822	9446	0	0	8446	0	8800	9174	0
21	0	155602	9816	0	9225	0	0	1372	159927	0	0	14240	0	12664	3965	0
23	1917	254	616	140	17637	261	604	368	3818	1717	0	1465	13100	471	14498	2881
25	1267	0	865	15090	359	2107	0	2429	0	4654	2191	17575	15184	9580	1931	0
27	0	0	586	0	1273	0	0	4623	23549	0	0	297	0	585	13015	0
31	6477	3924	21692	5640	332	0	490	27704	250	1406	185	379	253	3520	365	14592
33	0	0	13682	0	16959	0	0	6718	0	0	0	1994	0	2053	2180	0
35	0	12497	820	0	265	0	0	15250	0	0	0	547	0	251	3434	0
37	0	617	6125	912	13870	3399	21406	885	723	229	5265	1037	837	972	909	90250
41	0	52386	51113	0	54221	0	0	3930	44754	0	0	74738	0	11696	4972	0
43	12883	0	7156	5074	4128	6676	0	2928	0	8412	18832	18168	0	14370	14586	0
45	7550	7623	0	4591	23549	16302	4950	16167	4498	37402	2142	19979	28922	17354	20076	24917
47	0	29280	13511	0	24500	0	0	13531	0	0	0	13803	0	16293	113221	0
51	8407	4568	39701	11210	0	525	12873	24937	3149	14515	129624	24140	3039	15473	27847	69808
53	0	11421	16531	0	4511	0	0	23816	149866	0	0	13064	0	9364	7155	0
55	0	0	8543	0	6700	0	0	21371	0	0	0	11158	0	14395	11214	0
57	17131	27777	26594	1752	14839	10324	8576	0	23236	15364	10750	15603	6537	6747	23012	6439
61	10064	0	5965	0	2505	23247	0	35109	0	3933	3802	9639	3637	89620	14001	0
63	0	7429	35644	0	29862	0	0	19276	19300	0	0	6196	0	14240	18842	0
65	0	113080	5937	0	20680	0	0	14321	7637	0	0	19730	0	14989	4491	0
67	12965	25181	13423	15669	12916	7387	7400	13738	3109	13093	9074	0	6612	2022	20297	30407
71	0	0	15047	0	13372	0	0	14385	5929	0	0	14642	0	12603	7323	0
73	12240	1979	40387	16590	24624	14869	5216	19117	35211	7683	12892	3100	4207	0	6536	93671
75	15321	20015	23252	40156	9457	106484	6020	112287	78940	18715	2690	26072	2081	9212	0	18490
77	0	0	33231	0	26997	0	0	1144	0	0	0	1291	0	1185	1035	0

We used an AWGN noise and a BPSK modulation. All the simulations were made for signal/noise ratio equal with 1 dB and for a number of 500 errors, at least.

An iterations stop criterion was used for each decoder. When the resulting errors number for a data block is zero, the remaining iterations are not effectuated, passing to the next block.

The tables contain bit error rate ($BER \cdot 10^8$) obtained for each polynomial pair indicated in octal, at the beginning of each row (the denominator, b(D), from relation 6), or column (the numerator, a(D), from relation 6).

Table 3 (S- interleaver)

	1	3	5	7	11	13	15	17	21	23	25	27	31	33	35	37
1	0	2782307	2943825	1786715	3066143	1207492	1339953	1123232	2853139	1250243	1811238	611821	1159192	575409	691787	657002
3	3713565	0	0	1513452	0	480592	586229	0	0	188714	158108	0	235426	0	0	515695
5	4532351	0	0	38628	0	54530	45274	0	0	2482	1401345	0	3080	0	0	11095
7	1040109	1022982	3436	0	0	2994	2956	305910	832	817	0	6151	850	0	4025	241462
11	3979820	0	0	0	0	36253	41956	0	0	41470	0	0	35644	0	0	477
13	3979820	24035	5447	4684	3999	0	143	242	167	763	185	1311	479	182	0	100
15	118335	36556	7559	4703	1166	267	0	185	266	738	145	0	707	401	1421	228
17	394618	0	0	162101	0	332	365	0	0	302	3075	0	307	0	0	193675
21	4155989	0	0	594	0	781	830	0	0	42699	46773	0	48925	0	0	10501
23	186127	5812	539	658	1333	752	144	333	24686	0	313	743	6509	1428	2622	794
25	1198150	2598	1069026	0	0	539	162	640	2893	4516	0	731	2887	0	908	758
27	7770	0	0	2034	0	1874	0	0	0	258	732	0	506	0	0	432
31	180019	6177	1006	568	2092	220	939	501	5303	5826	437	1721	0	945	439	1285
33	5135	0	0	0	0	1273	1560	0	0	302	0	0	252	0	0	1161
35	9702	0	0	2503	0	0	3214	0	0	460	487	0	439	0	0	324
37	102566	114577	322	76369	275	385	317	81253	543	397	4446	565	421	570	404	0
41	4212043	0	0	1934	0	1139	824	0	0	639	1355	0	903	0	0	0
43	186387	5332	2908	0	0	1305	0	2480	9810	550	0	0	1105	0	15923	6397
45	359028	10573	6894	2182	7325	413	737	2858	1180	2040	3133	2808	4826	15890	3769	14232
47	9265	0	0	13877	0	0	2067	0	0	992	9972	0	7480	0	0	9044
51	228619	14099	4668	2885	2697	1039	1665	9634	1735	8191	3568	3668	8333	13754	1056	13976
53	12047	0	0	1625	0	8100	2027	0	0	777	1633	0	0	0	0	2386
55	29230	0	0	0	0	4696	6396	0	0	3125	0	0	1994	0	0	6292
57	8908	4020	1333	2419	612	1843	19325	949	4723	739	16047	1396	71654	16720	4934	4592
61	227980	4904	2968	0	0	0	846	703	6697	2943	0	10200	291	0	0	3813
63	10090	0	0	263	0	417	384	0	0	307	167	0	190	0	0	387
65	8205	0	0	817	0	2784	2383	0	0	0	1867	0	2581	0	0	1138
67	3536	4241	1694	2789	12315	190	2025	287	28519	130035	26804	1066	648	1183	1525	2540
71	6549	0	0	2588	0	2093	0	0	0	2634	2889	0	1310	0	0	24800
73	3793	12398	1442	43816	15414	2714	672	788	33805	1276	113225	12719	19350	2428	714	3802
75	12376	2481	3854	2155	1844	10375	687	377	11232	19005	8164	5724	329	15031	934	2206
77	70523	0	0	0	0	2842	2491	0	0	5198	0	0	3437	0	0	48073

	41	43	45	47	51	53	55	57	61	63	65	67	71	73	75	77
1	2993273	1207492	1223435	556141	1188807	549547	717488	262103	1134529	583417	583417	260213	494650	250240	333025	367943
3	0	131135	31141	0	28824	0	0	39378	117391	0	0	104040	0	102492	33592	0
5	0	3172	41831	0	40679	0	0	9642	3278	0	0	358	0	727	8419	0
7	276	0	127	9802	74	197	0	373	0	258	125	8786	8039	11305	363	0
11	0	0	41854	0	49803	0	0	508	0	0	0	485	0	519	569	0
13	1649	83	80	0	61	386	473	127	0	147	935	196	121	396	193	287
15	1299	0	286	94	73	1313	499	208	186	128	518	350	0	342	122	575
17	0	333	532	0	533	0	0	335	335	0	0	488	0	782	402	0
21	0	45040	1133	0	588	0	0	1887	35638	0	0	1074	0	2472	2761	0
23	11315	233	1940	136	25743	437	1122	391	3591	2305	0	2710	4540	330	15135	2778
25	1331	0	538	13856	204	1010	0	1071	0	15702	878	13606	19005	9179	708	0
27	0	0	202	0	752	0	0	982	16096	0	0	520	0	249	152923	0
31	13931	2866	13021	16983	425	0	600	69932	269	1091	217	286	167	13009	207	4345
33	0	0	7457	0	14444	0	0	17526	0	0	0	919	0	1416	1690	0
35	0	23136	258	0	336	0	0	7477	0	0	0	218	0	335	2291	0
37	0	731	5433	390	22609	3707	14937	376	331	384	4299	1593	534	923	798	58997
41	0	36471	47788	0	40759	0	0	30444	46317	0	0	18534	0	401227	6071	0
43	9174	0	2550	2094	1518	4200	0	4059	0	5968	20458	20065	0	444	15899	0
45	10600	14247	0	23325	26952	22341	4008	22238	5201	31271	3309	16184	8513	98094	24113	10032
47	0	2094	5453	0	16476	0	0	3385	0	0	0	4066	0	30534	26549	0
51	20778	1979	25027	13750	0	2564	4858	21800	8835	52297	15758	99546	3659	14286	16219	68429
53	0	28429	44149	0	3700	0	0	4375	206608	0	0	14589	0	15480	13301	0
55	0	0	16959	0	14568	0	0	5918	0	0	0	15839	0	923983	6148	0
57	22967	36629	28408	616	8852	1526	3734	0	6688	29861	39126	15876	27662	17594	32804	18244
61	13068	0	3898	0	2653	23247	0	32727	0	2570	3929	3518	686	64571	4821	0
63	0	19832	53766	0	30534	0	0	8299	3021	0	0	7494	0	13151	30265	0
65	0	65910	14435	0	26699	0	0	8136	10792	0	0	33623	0	39829	3974	0
67	61580	63321	111671	5047	66312	7881	24005	29670	21168	10619	13226	0	18438	1958	19668	16879
71	0	0	27579	0	12537	0	0	192134	7742	0	0	22430	0	20300	5443	0
73	27606	3694	31173	17325	108604	14879	4281	17458	122750	99527	2136	3568	14395	0	61241	25116
75	24360	10319	17105	15974	18931	12671	2340	119031	27355	24284	3416	22923	1731	27310	0	48372
77	0	0	27526	0	27316	0	0	64	0	0	0	1055	0	1720	582	0

The zeros in the tables contain mark that the respective codes have common divisors (see Table 1) and can not be used together. With little exceptions, the results obtained with S-interleaver are superior to the results corresponding the pseudo-random interleaver. We also remark the superior results obtained with both interleavers in the case of the use at denominator, b(D), of the primitive polynomials, comparing to non-primitive polynomials, at the same constraint length. Despite of the fact that global performances increase proportionally with K, it must be remarked the performances obtained using, for the Feed Back loop, the polynomials $b_1=7=111$, $b_2=13=1011$ and $b_3=15=1101$ as in combination with the polynomials with the same degree (ex. 15/13 or 13/15) as in combination with the polynomials with superior degree (ex. 51/7, 51/13 or 51/15).

Good performances are obtained using the Feed Back with $b_4=23=10011$, indicated in [2] (25/23, 33/23, 37/23): The last two combinations (33/23 and 37/23), correspond to the situation when the pseudo-random interleaver is superior to S-interleaver.

In the following are presented some practical results.

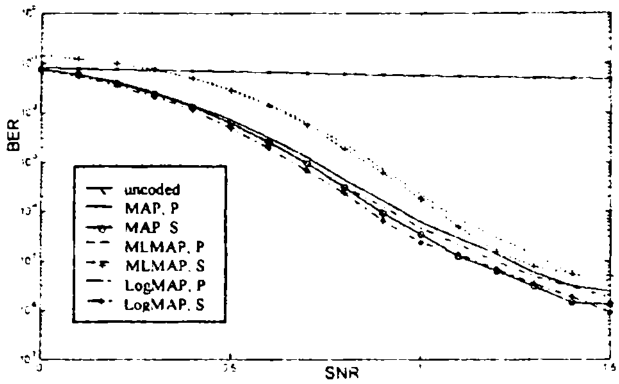


Fig.3 Simulation of rate 1/3 PCCC, 5/7 code, pseudo-random interleaver and S-interleaver

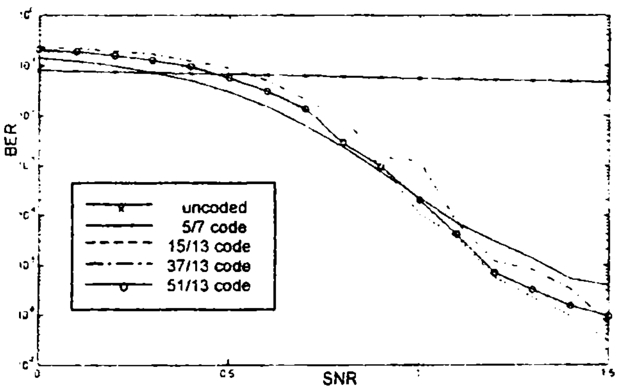


Fig.4 Simulation of rate 1/3 PCCC, P-interleaver, MaxLogMAP, for 5/7, 15/13, 37/13 and 51/13 codes

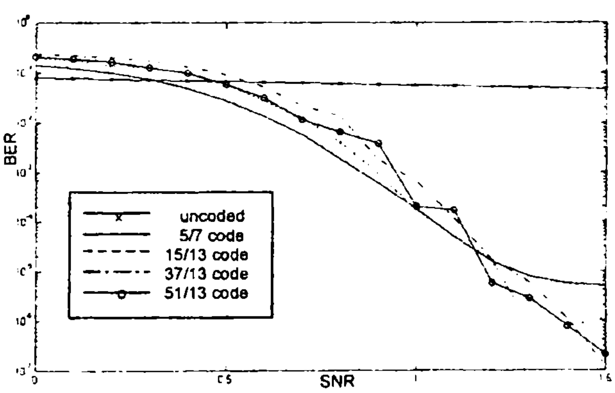


Fig.5 Simulation of rate 1/3 PCCC, S-interleaver, MaxLogMAP, for 5/7, 15/13, 37/13 and 51/13 codes.

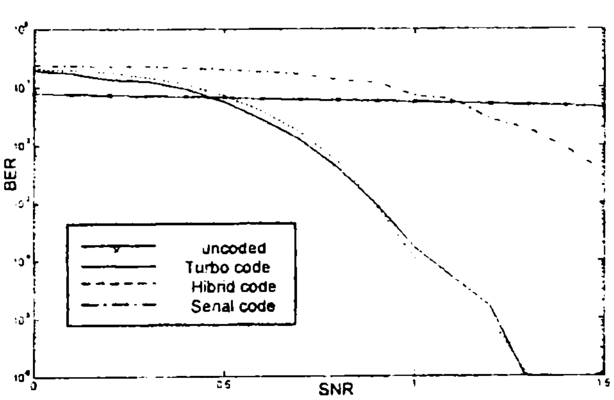


Fig.6 Simulation of rate 1/3 PCCC, 1/3 HCCC, 1/4 SCCC, S-interleaver, MaxLogMAP algorithm, 15/13 code.

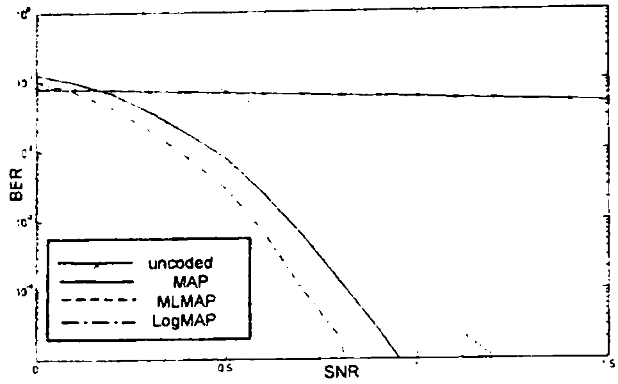


Fig.7 Simulation of rate 1/3 HCCC, 15/13 code, S-interleaver

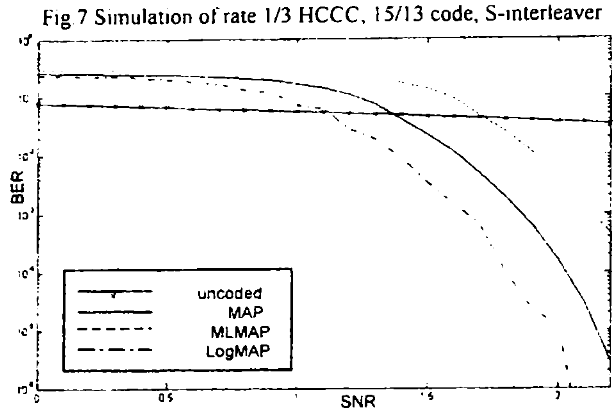


Fig.8 Simulation of rate 1/3 SCCC, 15/13 code, S-interleaver.

Th- ...l- -b-i--u ...i-- ...ffe-en- -co---g algorithms (MAP, MaxLogMap and LogMAP), with two types of interleavers: pseudo-random and S-type interleaver, with S=29, are comparted in Fig.3. In all th- ca-es ... -s-d -- 5/7 -od- (-h- m--- p-rf-rm-n- from all codes which have K=3) with data blocks size equal with N=1784 bits. We can remark the superiority of S-interleaver versus the pseudo-random interleaver. For example, in the case of MAP algo.ithm, .he ..-in.e.leaver b.in-s a SNR i--------e-t -f 0,1dB, -t BER=10^{-5}, -er--- th- pseudo-random interleaver. Surprisingly, in the algorithms performances hierarchy the first place is took by the LogMAP algorithm, especially under 1dB. The performances of MAP and Log MAP algor'thms are w'th approx'mation equa's over 1 -. As we expected, MaxLogMAP algorithm is with 0,2 dB inferior than the two enounced above.

Fig. 4 and Fig. 5 compare the most performant codes -rom eac' cons rain leng' : for K=3 is he 5/- code, for K=4 is the 15/13 code, for K=5 is the 37/13 code and for K=6 is the 51/13 code, in the case of using MaxLogMAP algorithm (the fastest in simulations) for pseudo-random interleaver and S-interleaver. Thought we used 10^8 transmitted bits (or more) sometimes it was insufficient to obtain smooth curves. Besides of this observation, we notice that for SNRs inferior to 1 dB, the 5/7 code is the best. When SNR increases more than 1 dB, the codes with K>3 are more performants. This fact leads to the idea that the --d-- -i-r-r--y --n --.ng-- -t -ig--- SNR. N-ti-- that turbo codes which have Feed Back loop realized on the basis of polynomial 13, for 1dB, are the best. Probably, for SNR higher than 1dB, maybe the xx/13 codes can lose their supremacy. This verification constitute the objective of a future study which we propose us.

The diagrams, for the last three figures, show the BER performances obtained with different concatenation modes (parallel, hybrid and serial) and with different algorithms (MAP, MaxLogMAP and LogMAP). In all the cases we used the 15/13 code and the S-interleaver.

Obvious, the serial concatenation is less performant than parallel and hybrid concatenations. Because of multiplexing and restriction using of interleavers with the same length, N=1784, in the case of SCCC code it results a number of N=1784/2=892 information bits per block, than N=1784 for PCCC and HCCC codes. Moreover, because the SCCC transmission rate is of 1/4 versus 1/3 in the case of PCCC and HCCC codes, for the equivalence of the ratio between the transmitted energy in information bit (E_0) and the noise power spectral energy ($N_0/2$), for the 1/4 transmission rate case, the channel noise power is higher than the corresponding power for the 1/3 transmision rate case.

We remark the falling of the SCCC codes, for SNR of 2 dB, falling what it's not find in the case of the PCCC and HCCC codes.

Finally, we also notice the good behavior of the LogMAP algorithm in the case of SCCC and HCCC, fact which "invite" us to make an investigation more detailed of this algorithm for the future.

III. CONCLUSIONS

In this paper we presented and compared the BER performances obtained by the simulation of a transmission system, which utilizes the forward error correcting by codes concatenation and iterative decoding (turbo coding). We used two interleaver types, pseudo-random and S-interleaver with the length equal with N=1784. The BPSK modulation and the AWGN channel were employed. The MAP, MaxLogMAP and Log MAP algorithms were used for decoding.

There have been investigated all the systematic convolutional codes having the constraint length K less or equal to 6, under three different concatenated forms: parallel PCCC (pure turbo code), serial SCCC and hybrid HCCC, at the following rates: 1/3, 1/4 and 1/3, respectively, all unpunctured.

REFERENCES

[1] C. Berrou, A. Glavieux, P. Thitimajshima "*Near Shannon limit error-correcting coding and decoding: Turbo-codes*", *Proc.ICC'93*, Geneva, Switzerland, May 1993, pp. 1064-1070.

[2] Consultative Committee for Space Data Systems "*Telemetry Channel Coding*", CCSDS 101.0-B-6, Blue Book, October, 2002., http://www.ccsds.org/documents/101x0b6.pdf

[3] P. Ha, "*A Fast Algorithm for Generating Random Interleavers*", www.sarim.changwon.ac.kr

[4] J.Hagenauer, E. Offer, L. Papke, "*Iterative Decoding of Binary Block and Convolutional Codes*", IEEE Transactions on Information Theory, Vol 42 No 2, March 1996 pp 429-445

[5] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "*Optimal Decoding of Linear Codes for Minimising Symbol Error Rate*", IEEE Transactions on Information Theory, Vol. 20, pp. 284-287, March 1974.

[6] 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

[7] P. Robertson, E.Villebrun, P.Hoeher, „*A Comparison of Optimal and Sub*-Optimal MAP Decoding Algorithms Operating in the Log Domain", Proceedings of the International Conference on Communications, Seattle, USA, pag. 1009-1013, iunie 1995