

Performanțele codurilor convoluționale în configurație turbo

Horia *Balta*, Iancu *Polverea*, Iulică *Gheorghiiță*

Introducere

Turbo-codurile formează o clasă de coduri corectoare de erori ce operează în apropierea limitei Shannon. În (fig.1) se prezintă schema unui turbo-cod în configurație paralelă, [1]. Secvența de informație, notată u , este codată de către codorul C1 rezultând secvența de paritate x_1 . Aceeași secvență de biți, u , este furnizată, însă în altă ordine prin întretesere cu ajutorul dispozitivului de întretesere „I”, codorului C2, care generează la rândul său, secvența de paritate x_2 .

Secvențele rezultate $x_0=u$, x_1 și x_2 , prin multiplexare și modulare (operații omise în Fig.1.) constituie ieșirea turbocodului, semnal ce va fi emis în canal. La ieșirea acestuia, prin demodulare și demultiplexare rezultă secvențele (soft) recepționate, corespunzătoare, y_0 , y_1 și y_2 .

Fiecare decodor calculează logaritmul raportului de plauzibilitate (Log Likelihood Ratio –LLR) pentru fiecare bit din u :

$$LLR(u_i) = \ln \frac{p(u_i = 1|y)}{p(u_i = 0|y)} \quad (1)$$

(în figură este prezentat doar logaritmul raportului de plauzibilitate al primului decodor, notat LLR1) și de asemenea informația extrinsecă destinată celui alt decodor.

Fiecare decodor primește informație extrinsecă și, pe baza ei și pe baza secvențelor venite din canal (y_0 și y_1 respectiv y_0 întretesută și y_2) furnizează la rândul-i informație extrinsecă. Acest proces se repetă iterativ de un anumit număr de ori (impus sau calculat, funcție de tipul turbocodului). După efectuarea tuturor iterațiilor impuse, se face o decizie hard asupra logaritmului raportului de plauzibilitate generat după ultima iterație de unul din cele două decodare (în fig.1 s-a ales LLR1). Secvența rezultată prin operația de decizie hard constituie ieșirea turbo-decodorului.

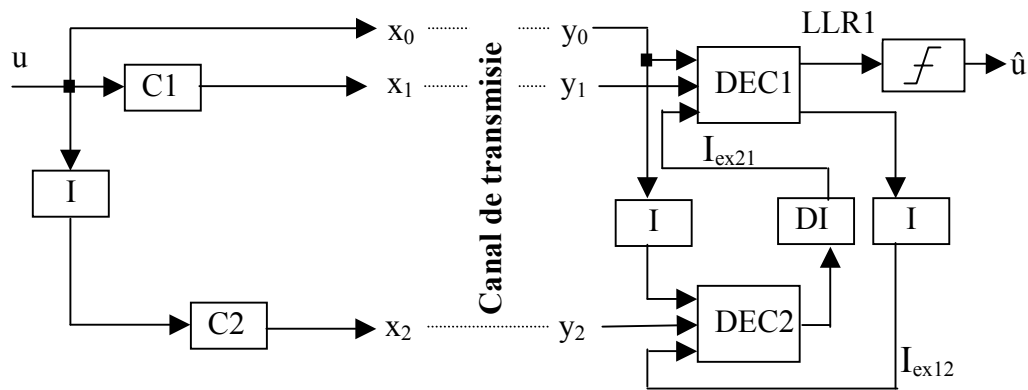


Fig. 1. Turbocod –schemă generală

Codoarele utilizate în turbo-coduri sunt în special cele recursive și sistematice. Acestea vor fi descrise mai detaliat în paragraful următor.

În contrast cu codorul convoluțional, care are o implementare hard simplă, după cum se vede în (fig.2), un decodor, pentru a putea fi component al unui turbocod, trebuie să accepte intrare soft și, de asemenea, să ofere ieșire soft.

Algoritmul Maximum A-Posteriori (MAP)

Este cel mai performant algoritm de decodare utilizat în TCC. El a fost propus de Bahl, Cocke, Jelinek și Raviv (1974). Algoritmul MAP calculează logaritmul raportului de plauzibilitate în forma [2]:

$$L(u_k|y) = \ln \left(\frac{\sum_{\substack{(\hat{s},s) \Rightarrow \\ u_k = +1}} P(S_{k-1} = \hat{s} \wedge S_k = s \wedge \underline{y})}{\sum_{\substack{(\hat{s},s) \Rightarrow \\ u_k = -1}} P(S_{k-1} = \hat{s} \wedge S_k = s \wedge \underline{y})} \right) = \ln \left(\frac{\sum_{\substack{(\hat{s},s) \Rightarrow \\ u_k = +1}} \alpha_{k-1}(\hat{s}) \cdot \gamma_k(\hat{s},s) \cdot \beta_k(s)}{\sum_{\substack{(\hat{s},s) \Rightarrow \\ u_k = -1}} \alpha_{k-1}(\hat{s}) \cdot \gamma_k(\hat{s},s) \cdot \beta_k(s)} \right) \quad (2)$$

unde:

$\alpha_{k-1}(\hat{s}) = P(S_{k-1} = \hat{s} \wedge y_{j < k})$ este probabilitatea ca trellisul să fie în starea \hat{s} la momentul $k-1$ iar secvența recepționată din canal până în acest moment să fie $y_{j < k}$,

$$\alpha_k(s) = \sum_{\text{toti } \hat{s}} \gamma_k(\hat{s},s) \cdot \alpha_{k-1}(\hat{s}) \quad (3)$$

$\beta_k(s) = P(y_{j > k} / S_k = s)$ este probabilitatea ca, dată fiind starea trellisului s la momentul k , secvența recepționată din canal, după acest moment, să fie $y_{j > k}$,

$$\beta_{k-1}(\hat{s}) = \sum_{\text{toti } s} \gamma_k(\hat{s},s) \cdot \beta_k(s) \quad (4)$$

$\gamma_k(\hat{s},s) = P(\{y_k \wedge S_k = s\} / S_{k-1} = \hat{s})$ este probabilitatea ca, dat fiind trellisul în starea \hat{s} la momentul $k-1$, să treacă în starea s iar secvența recepționată din canal pentru această tranziție este y_k .

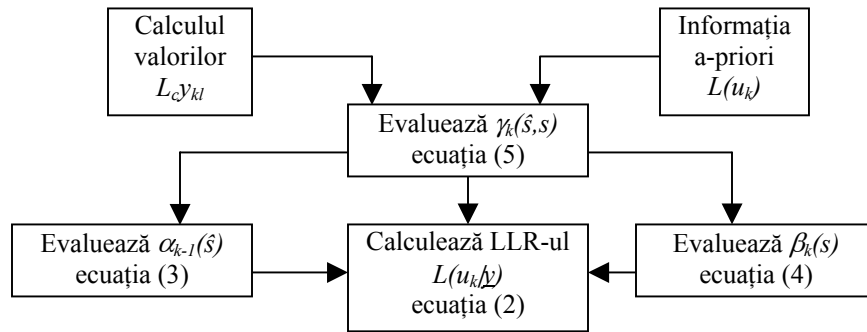


Fig. 2. Organigrama algoritmului MAP

$$\gamma_k(\hat{s}, s) = C \cdot e^{(u_k L(u_k)/2)} \cdot \exp\left(\frac{E_b}{2\sigma^2} \cdot 2 \cdot a \sum_{i=1}^n y_{ki} \cdot x_{ki}\right) =$$

$$C \cdot e^{(u_k L(u_k)/2)} \cdot \exp\left(\frac{L_c}{2} \sum_{i=1}^n y_{ki} \cdot x_{ki}\right), \quad (5)$$

în care $L_c = 2 \cdot a \cdot E_b / \sigma^2$ este coeficientul de încredere al canalului, $L(u_k)$ este informația extrinsecă reordonată (întreșesută), numită și apriori.

Dispozitivul de întreșesere (interleaver)

Este o componentă indispensabilă a oricărui turbocod. Datorită întreșeserii secvenței furnizate celui de-al doilea codor, se obține o decorelare între diferitele intrări ale unui decodor component, mai precis între secvențele: provenite din canal și cea provenită de la celălalt decodor component (informația intrinsecă).

Un dispozitiv de întreșesere realizează o permutare a unei secvențe de numere. Altfel spus, un dispozitiv de întreșesere implementează o funcție bijectivă de forma:

$$\pi: I \rightarrow I, \text{ cu } I = \{1, 2, \dots, N\}, \quad (6)$$

unde N reprezintă lungimea secvenței ce trebuie întreșesută. Pentru refacerea ordinii inițiale se utilizează un dispozitiv pereche, de de-întreșesere, ce implementează funcția inversă:

$$\pi^{-1}: I \rightarrow I, \text{ cu } \pi^{-1}(\pi(i)) = i, \forall i \in I. \quad (7)$$

Un bun interleaver trebuie să îndeplinească două condiții: a) să aibă o distanță minimă de întreșesere de valoare cât mai mare și b) să aibă un grad de împrăștiere cât mai bun.

Coduri convoluționale recursive și sistematice

În (fig.3) este prezentată schema generală a unui codor convoluțional recursiv și sistematic, de rată $1/2$:
Cu notațiile din figură rezultă ecuațiile:

$$y[n] = \sum_{k=0}^M a_k w[n-k] \quad (8)$$

$$w[n] = \sum_{k=0}^M b_k w[n-k] + x[n] \quad (9)$$

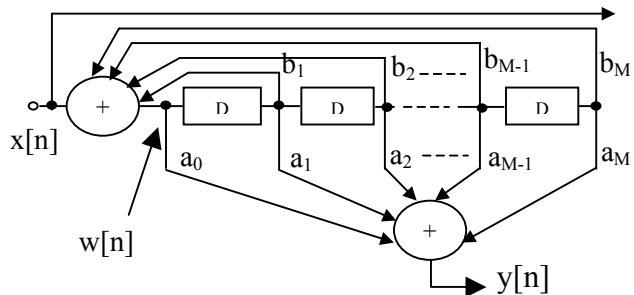


Fig. 3. Codor convoluțional recursiv și sistematic

Aplicând transformata "D" celor două ecuații (8) și (9) obținem:

$$Y(D) = \left(\sum_{k=0}^M a_k D^k \right) \cdot W(D) \quad (10)$$

$$W(D) \cdot \sum_{k=0}^M b_k D^k = X(D) \quad (11)$$

Eliminând pe $W(D)$ rezultă:

$$\frac{Y(D)}{X(D)} = \frac{\sum_{k=0}^M a_k D^k}{\sum_{k=0}^M b_k D^k} \quad (12)$$

ecuație care definește funcția de transfer (intrare-ieșire) a codorului. Matricea generatoare a codorului este:

$$G(D) = \left[1, \frac{a(D)}{b(D)} \right] \quad (13)$$

În mod evident polinoamele $a(D)$ -numit polinomul de ieșire și $b(D)$ -polinomul de reacție trebuie să fie prime între ele. Lungimea de constrângere K este dată de gradul maxim al polinoamelor $a(D)$ și $b(D)$ plus 1. În (tab.1) sunt indicate polinoamele cu coeficienți binari, până la gradul 4 ce pot fi utilizate cu rol de $a(D)$ sau $b(D)$. Nu sunt compatibile polinoamele ce au cel puțin un divizor comun. În prezenta lucrare au fost investigate toate turbo-codurile construite cu toate combinațiile posibile de coduri prezentate în (tab.1) exceptând cele de grad 0 sau 1 pentru care nu sunt așteptate rezultate notabile.

Tabelul 1. Polinoamele în octal și divizorii lor

Grad	Ireductibil	Primitiv	Polinomul	Divizorii
0	DA		1	
1	DA	DA	3	
2	DA		5	3
			7	
3			11	3 7
	DA	DA	13	
	DA	DA	15	
			17	3
4			21	3
	DA	DA	23	
			25	7
			27	3 15
	DA	DA	31	
			33	3 7
			35	3 13
	DA		37	

Rezultate experimentale

Turbo-codul utilizat în simulările a căror rezultate sunt prezentate în cele ce urmează are schema dată în (fig.1). Dispozitivul de întreșere (interleaver-ul) a fost de tip S (interleaver aleator cu distanța minimă de întreșere impusă de valoare S). Simulările au fost făcute, pentru toate perechile de coduri, la două lungimi de întreșere, 400 și 1784 biți. Lungimea de 1784 a fost aleasă conform cu [3]. La lungimea de 400 biți s-a utilizat un S-interleaver cu $S=13$ iar la 1784 un S-interleaver cu $S=29$. Canalul de transmisie a fost de tip AWGN iar modulația utilizată BPSK.

Decodarea componente DEC1 și DEC2 implementează algoritmul MAP cu 15 iterații și un criteriu de stop a iterațiilor bazat pe valorile LLR-ului definit de relația (2). Astfel, dacă după efectuarea unei iterații toate valorile LLR sunt în afara unui interval de forma $[-prag, +prag]$, atunci se întrerupe prematur procesul iterativ și se procedează la efectuarea deciziei hard. Valoarea pragului a fost aleasă în toate cazurile egală cu 10.

Nu s-a utilizat puncturare încât rata de turbo-codare rezultată a fost aproximativ $1/3$. Diferența se datorează utilizării unui procedeu de închidere a trellisului primului cod, C1, fapt realizat cu prețul inserării a $K-1$ biți redundanți în secvența de date. În acest fel rata de turbo-codare are valoarea $(N-K+1)/(3 \cdot N)$.

În (tab.2) și în (tab.3) sunt prezentate rezultatele simulărilor. Numerele înscrise în prima linie și prima coloană indică, în octal, valorile polinoamelor $a(D)$, respectiv $b(D)$. Perechile de polinoame incompatibile (ce au divizori comuni) sunt marcate printr-un zero. Rezultatele furnizate de ambele tabele (pentru perechile de polinoame compatibile) sunt: primul număr –rata erorii de bit (BER) înmulțită cu 10^6 pentru (tab.2) și cu 10^9 pentru (tab.3) și al doilea număr –numărul de biți eronați pentru care s-a calculat respectiva rată a erorii.

Toate simulările au fost făcute la un raport semnal per zgomot $SNR=1dB$.

Pentru $K=3$, rezultatele indică pe 5/7 codul cel mai performant, iar pentru $K=4$ pe 13/15 și 17/15. La $K=5$ există mai multe coduri cu performanțe foarte bune, 21/13, 25/13, 31/13, 23/15, 23/21, 31/23, 31/25, 13/31, 23/31, 25/31 și 33/31.

Concluzii

În această lucrare sunt comparate performanțele tuturor codurilor convoluționale recursive și sistematice, având lungimea de constrângere $K=3, 4$ și 5 , în configurație turbo. Comparația s-a făcut din punct de vedere al ratei erorii de bit la un raport semnal per zgomot de $1dB$. Pentru fiecare lungime de constrângere sunt indicate cele mai performante coduri.

Tabelul 2. Performanțele codurilor convoluționale cu $3 \leq K \leq 5$ la $N=400$

	5	7	11	13	15	17	21	23	25	27	31	33	35	37
5	0	1840	0	2112	2098	0	0	680	38	0	713	0	0	695
		73271		41943	41655			26943	360255		28247			27525
7	1245	0	0	1220	1178	6249	644	1	0	1296	564	0	1377	16
	49567			48465	46797	248119	25528	16969		51331	22349		54530	161704
11	0	0	0	1711	1730	0	0	1865	0	0	1762	0	0	609
				67964	68695			73889			34891			24122
13	1658	1329	1397	0	763	715	881	898	781	1254	1006	1129	0	660
	65862	52788	55482		30313	28403	34908	35575	30962	49668	39874	44715		26170
15	1718	1350	1427	759	0	719	887	1034	723	0	1	1100	1261	713
	68205	53617	56669	30140		28560	35156	40968	28645		22640	43591	49948	28267
17	0	5311	0	500	554	0	0	545	1069	0	531	0	0	14
		210882		19861	22018			21606	42368		21057			135771
21	0	1001	0	771	1	0	0	1973	13	0	12	0	0	1491
		39646		30567	22899			78146	78377		62485			59078
23	1288	1121	1792	1425	838	954	2425	0	940	980	1370	1198	1391	1
	51023	44403	71002	56439	33200	37801	96043		37244	38825	54271	47468	55093	48396
25	27	0	0	755	770	628	4	1220	0	1	1192	0	537	1
	318819			29926	30528	24903	64578	48351		11525	47213		21287	12694
27	0	1418	0	1289	0	0	0	777	1039	0	838	0	0	916
		56174		51049				30790	41169		33219			36310
31	1336	1153	1744	824	1423	927	2383	1346	1953	1336	0	1279	1084	1187
	52908	45673	69101	32634	56378	36721	94378	53316	437759	52908		50676	42947	47036
33	0	0	0	896	864	0	0	796	0	0	842	0	0	1197
				35515	34252			31541			33366			47418
35	0	3	0	0	1271	0	0	762	970	0	867	0	0	892
		51985			50357			30210	38440		34369			35357
37	1300	4498	935	855	883	4234	649	557	1906	610	612	700	616	0
	51508	89078	37063	33894	34933	83849	25726	22085	75489	24177	24253	27729	24431	

Tabelul 3. Performanțele codurilor convoluționale cu $3 \leq K \leq 5$ la $N=1784$

	5	7	11	13	15	17	21	23	25	27	31	33	35	37
5	0	339944 30289	0	575070 51210	6247 33459	0	0	20511 3651	6299 167950	0	22663 4034	0	0	71247 12682
7	15090 2689	0	0	6485 1155	164 1392	5257 83062	113 629	7258 1292	0	59 582	4399 783	0	71 823	52808 331024
11	0	0	0	322628 28730	320045 28500	0	0	831 148	0	0	7175 43739	0	0	4152 1478
13	7221 1286	23571 4198	803 143	0	427 152	59 21	8 21	2056 366	6 14	1893 337	96 17	73 26	0	404 144
15	4874 868	4116 733	803 143	2 5	0	2 4	5 27	11 2	32 152	0	9933 1768	87 31	2758 491	376 134
17	0	2129 25709	0	2965 1056	955076 33866	0	0	8247 20965	219 39	0	46 156	0	0	5646 35682
21	0	291 1190	0	3359 10719	106940 28553	0	0	28066 133694	403603 71852	0	25240 120013	0	0	124 22
23	646 230	783 209	2899 516	217 58	213 76	10 70	34 6	0	37 10	73 13	34 6	34 6	67 12	45 8
25	5148 122203	0	0	69 236	854 304	749 200	271933 48404	483 86	0	22551 4014	3 6	0	20607 5502	6809 1818
27	0	33022 5878	0	72067 12828	0	0	0	331 118	337 60	0	340 121	0	0	82 336
31	1112 198	28 56	978 174	5 10	298 53	11 45	22 4	<1 n	1 4	34 6	0	34 6	48 17	11 2
33	0	0	0	19663 3500	9230 1643	0	0	58 210	0	0	63 152	0	0	963 257
35	0	8427 1500	0	0	8298 1477	0	0	293 104	14 60	0	376 134	0	0	694 247
37	489 174	1374 9889	312 111	343 122	829 294	7817 64851	75 293	3573 1272	34 6	403 1595	3500 1246	246 677	6654 2369	0

Bibliografie

1. C. Berrou, A. Glavieux, P. Thitimajshima, „Near Shannon Limit Error –Correcting Coding and Decoding: Turbo –Codes”, Proc. of ICC, Geneve, may 1993, pp. 1064-1070
2. 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
3. Consultative Committee for Space Data Systems “Telemetry Channel Coding”, Blue Book, CCSDS 101.0-B-6, Oct 2002.
4. M. Jézéquel, C. Berrou, C. Douillard, „Turbo codes (convolutifs)”, seminar, Timisoara, Romania, 15-18 March, 2004, <http://hermes.etc.utt.ro/cercetare/carti.html>.
5. H. Balta, M. Kovaci - “The BER Performances of Convolutional Codes used in Turbo Codes”, Scientific Bulletin of the “Politehnica” University of Timișoara, România, Oct 2004, pp. 38 –43.