

CONTRIBUȚII LA DEZVOLTAREA ȘI PROIECTAREA TURBO-CODURILOR BINARE ȘI NEBINARE

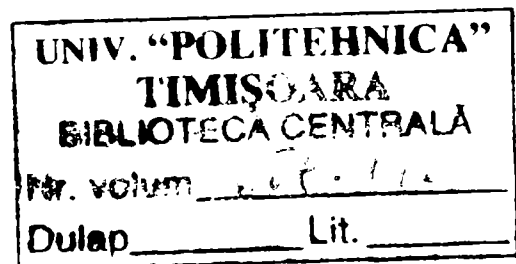
Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea "Politehnica" din Timișoara
în domeniul Inginerie Electronică și Telecomunicații
de către

ing. Horia Gheorghe Balta

Conducător științific:
Referenți științifici:

prof.univ.dr.ing. Miranda Naforniță
prof.univ.dr.ing. Monica Borda
prof.univ.dr.ing. Daniela Tărniceriu
prof.univ.dr.ing. Alexandru Isar

12.2008



Seriile Teze de doctorat ale UPT sunt:

- | | |
|------------------------|---|
| 1. Automatică | 7. Inginerie Electronică și Telecomunicații |
| 2. Chimie | 8. Inginerie Industrială |
| 3. Energetică | 9. Inginerie Mecanică |
| 4. Ingineria Chimică | 10. Știința Calculatoarelor |
| 5. Inginerie Civilă | 11. Știința și Ingineria Materialelor |
| 6. Inginerie Electrică | |

Universitatea „Politehnica” din Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2008

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității „Politehnica” din Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Prezenta lucrare constituie rezultatul unei activități doctorale de peste șapte ani și, totodată, a unei activități didactice de peste șaptesprezece ani. Mai precis, activitatea mea didactică se identifică, ca și durată, cu perioada post-decembristă. Inevitabil, tumulturile și indeciziile vieții sociale de după Decembrie '89 și-au lăsat amprenta și asupra propriei mele vieți. Au fost momente în care neajunsurile, frustrările și chiar umilințele la care a ajuns statutul de dascăl, m-au făcut să iau în calcul renunțarea la cariera universitară. Domnul Prof. Dr. Ing. Ioan Naforniță este cel care m-a determinat să continui și, de asemenea, să încep pregătirea unui doctorat, a cărui finalitate o constituie prezenta teză.

Un alt fapt pe care consider că îl datorez, în egală măsură, d-lui Prof. Dr. Ing. Ioan Naforniță, și conducătorului meu de doctorat, doamna Prof. Dr. Ing. Miranda Naforniță, este îndrăzneala abordării unui domeniu cu totul nou pentru școala noastră, timișoreană, anume turbo-codurile.

Privind retrospectiv activitatea doctorală trebuie să recunosc că am avut parte de câteva șanse. În primul rând am beneficiat de experiența unei școli doctorale bine fundamentate și bine organizate: Politehnica din Timișoara. Remarc de asemenea „atmosfera” propice studiului și cercetării științifice a colectivului din care fac și eu parte. Colegii mai tineri sau mai vârstnici, motivați de pasiunea pentru domeniul fascinant al telecomunicațiilor, au constituit un real sprijin pentru mine. Aceste atuuri mi-au permis să compensez lipsa de experiență din domeniul abordat.

Cu totul aparte a fost șansa de a audia seminariile susținute în Timișoara de doi dintre profesorii colectivului de la Brest¹, Franța, colectiv din care face parte și domnul profesor Claude Berrou, părintele turbo-codurilor. Este vorba despre domnul profesor Michel Jezequel și doamna profesoară Catherine Douillard. Această șansă mi-a permis să aflu detalii și să elimin incertitudini.

În ceea ce privește munca de cercetare depusă, aş putea sintetiza astfel:

- 1) căutare;
- 2) asimilare-construcție;
- 3) simulare-verificare.

Cele trei etape/faze ale cercetării nu se referă la perioade distincte de timp ci la teme concrete. Pe parcurs am abordat mai multe subiecte astfel că la un moment dat de timp ne aflam (echipa) simultan în toate cele trei etape, dar cu subiecte diferite.

Nu în ultimul rând, doresc să subliniez buna colaborare pe care am avut-o cu studenți, din mai multe generații, ai Facultății de Electronică și Telecomunicații. Îmi doresc și am toate motivele să cred că această colaborare nu se va încheia curând.

Horia Balta

¹ Ecole Nationale Supérieure des Telecommunications de Bretagne, Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3 - FRANCE.

Balta, Horia-Gheorghe

Contribuții la dezvoltarea și proiectarea turbo-codurilor binare și nebinare

Teze de doctorat ale UPT, Seria 7, Nr. 4, Editura Politehnica, 2008, 184 pagini, 61 figuri, 26 tabele.

ISSN: 1842-7014

ISBN: 978-973-625-601-1

Cuvinte cheie:

turbo-coduri, întrețesere, trellis, algoritm Maxim APosteriori

Rezumat: Turbo-codul constituie cea mai performantă metodă de corecție a erorilor ce apar în sistemele de transmisiuni actuale. De la apariția lor, în 1993, turbo-codurile și implicit teoria informației și a codării a cunoscut o dezvoltare explozivă. În această direcție, lucrarea de față aduce câteva contribuții originale, atât teoretice cât și practice: o nouă metodă de generare a numerelor aleatoare cu distribuție Nakagami, o nouă metodă de analiză a interleaverelor (spectrul distanțelor de întrețesere), sunt construite două turbo-coduri duo-binare de memorie 3 cu performanțe superioare celor din literatură, sunt propuse noi familii de turbocoduri (cu intrări multiple și nebinare).

Cuprins

Lista figurilor	7
Lista tabelelor	10
Lista acronimelor	11
Lista simbolurilor	15

1. Introducere

1.1 Motivația cercetării	17
1.2 Scopul lucrării	18
1.5 Structura lucrării	18

2. Codarea canalului

2.1 Scurt istoric asupra codării canalului	20
2.2 Principiul „turbo”	22
2.3 Turbo-codorul	24
2.4 Configurații de turbo-coduri	25
2.5 Limite teoretice în teoria informației	28

3. Simularea sistemului de transmisie

3.1 Metoda de simulare Monte Carlo	30
3.2 Modele și metode de simulare a sistemelor de transmisie ce utilizează turbo-codare	31
3.3 Performanțele sistemelor de transmisie cu modulație BPSK, DQPSK deplasat cu $\pi/4$ și 16QAM, necodate	39
3.4 Generarea numerelor aleatoare cu distribuție Nakagami	41

4. Turbo-Coduri Uni-Binare

4.1 Structura unui turbo-cod	47
4.2 Codurile convoluționale componente	48
4.3 Configurația turbo-codului	51
4.4 Întreșeserea	60
4.5 Algoritmul de decodare	74
4.6 Închiderea trellisului	84
4.7 Puncturarea	89
4.8 Numărul de iterații și criteriile de oprire ale iterațiilor	94

5. Turbo-Coduri Multi-Binare

5.1 Configurații de codoare convoluționale multi-binare	103
5.2 Echivalența formelor canonice H și G	107

5.3	Construcția turbo-codurilor multi-binare.....	113
5.4	Închiderea trellisurilor MBC-lor.....	114
5.5	Decodarea per caracter a MBTC-lor	121
5.6	Criteriu de oprire a iterațiilor pentru MBTC-uri	122
5.7	Proiectarea MBTC-lor	126
6.	Turbo-Coduri Multi-Non-Binare	
6.1	Introducere	135
6.2	Codor convoluțional multi-non-binar	135
6.3	Decodarea multi-non-binară	138
6.4	Închiderea trellisului MNBTC-lor	140
6.5	Turbo-cod Reed-Solomon	141
6.6	Performanțele MNBTC-lor	142
7.	Contribuții originale.....	144
	Anexe	
	Anexa A Performanțele BER ale BTC-lor	146
	Anexa B MBCC-le de memorie 4 selectate după criteriul convergenței.....	154
	Anexa C Calculul inversei matricii ($I_M + D \cdot T$).....	158
	Anexa D Densitatea de probabilitate Nakagami.....	162
	Bibliografie	170

Lista figurilor

Figura 2.1 Ilustrarea principiului „turbo”	23
Figura 2.2 Codor / decodor pentru un turbo-cod (Cod concatenat paralel)	24
Figura 2.3 Codor/ decodor pentru un turbo-cod triplu în structură paralelă	26
Figura 2.4 Codor/ decodor pentru un turbo-cod în configurație serială	27
Figura 2.5 Două scheme de codoare pentru coduri concatenate hibrid	27
Figura 3.1 Ilustrarea metodei de simulare Monte Carlo	30
Figura 3.2 Model pentru simularea unui sistem de transmisie turbo-codat	31
Figura 3.3 Modele de simulare: a) sistem de transmisie necodat cu modulație BPSK & canal cu fading plat și zgomot AWGN; b) sistemul de transmisie turbo-codat.	34
Figura 3.4 Modelul canalului digital cu fading plat și modulație DQPSK deplasat cu $\pi/4$	35
Figura 3.5 Modele de simulare pentru sisteme de transmisie cu modulație DQPSK deplasat cu $\pi/4$	36
Figura 3.6 Modelul canalului digital cu fading plat și modulație 16QAM	38
Figura 3.7 Modele de simulare pentru sisteme de transmisie turbo-codate cu modulație 16QAM	38
Figura 3.8 Performanțele BER obținute prin simularea sistemelor de transmisie cu modulație BPSK și $\pi/4$ DQPSK, în canale AWGN și cu fading plat de tip Rayleigh	39
Figura 3.9 Performanțele BER obținute prin simularea unui sistem de transmisie cu modulație BPSK în canale cu fading plat de tip Nakagami cu $m=2, 3, 4$ și 5 , [PRO00]	40
Figura 3.10 Densitatea de probabilitate și funcția de repartiție a probabilității pentru variabila aleatoare x , pentru $m=2$	42
Figura 3.11 Ilustrarea algoritmului de generare a unei variabile aleatoare cu distribuție Nakagami	44
Figura 4.1 Schema generică a unui turbo-cod și parametrii ce intervin în definirea sa	47
Figura 4.2. Codor convoluțional recursiv și sistematic cu sumatoare externe (forma canonică "controller")	38
Figura 4.3 Codor convoluțional recursiv și sistematic cu sumatoare interne (forma canonică „observer”)	50
Figura 4.4 Schema corespunzătoare unui cod convoluțional concatenat paralel (PCCC)	52
Figura 4.5 Performanțele unor TC concatenate paralel și interleaver de tip S	53
Figura 4.6 Schema corespunzătoare unui cod convoluțional concatenat serial (SCCC)	53
Figura 4.7 Performanțele SCCC-urilor	54
Figura 4.8 Două scheme de coduri convoluționale concatenate hibrid	56

Figura 4.9 Performanțele HCCC-urilor	57
Figura 4.10 Comparație între performanțele celor mai bune TC-uri în configurație: paralelă (17/15), hibridă (15/13) și serială (13/3)	59
Figura 4.11 Dispozitiv de întreținere aleator	61
Figura 4.12 Întreținere bloc	61
Figura 4.13 Spectrele distanțelor de întreținere: număr de perechi de poziții i și j ce au distanța d , funcție de distanța d , pentru interleaver-urile rectangular cu $N=1024$ (IR32), pseudo-aleator cu $k_1=10$ și $N=1000$ (IP6), pseudo-aleator cu $k_1=8$ și $N=1024$ (IP12), pseudo-aleator cu $k_1=8$ și $N=1000$ (IP10), aleator (IS1A) și S-interleaver-urile cu $S=5$ (IS5A), cu $S=10$ (IS10A), cu $S=20$ (IS20A), cu $S=23$ (IS23A)	64
Figura 4.14 a) Spectrele mediate pentru dispozitivele de întreținere de tip S; b) Detalii	67
Figura 4.15 Spectrul teoretic al dispozitivului de întreținere aleator cu $N=1000$	69
Figura 4.16 Rata erorii (BER) funcție de raportul semnal per zgomot (SNR) în dB	73
Figura 4.17 Organigrama algoritmului MAP	75
Figura 4.18 Performanțele algoritmului MAP funcție de estimatul Lc-ului	77
Figura 4.19 Aproximările funcției $f_c(x)$	81
Figura 4.20 Performanțele algoritmilor de decodare	83
Figura 4.21 Moduri de închidere a trellisurilor	85
Figura 4.22 Performanțele turbo-codurilor în diferitele variante de închidere a trellisurilor	88
Figura 4.23 Schema unui sistem de transmisie turbo-codat cu puncturare	89
Figura 4.24 Performanțele TC-ilor puncturate	92
Figura 4.25 Performanțele funcție de numărul de iterații ale TC-ului construit cu codul 5/7	95
Figura 4.26 Performanțele funcție de numărul de iterații ale TC-ului construit cu codul 15/13	96
Figura 4.27 Performanțele funcție de numărul de iterații ale TC-ului construit cu codul 25/31	97
Figura 4.28 Histograme ale valorilor LLR (pe abscisă – valori ale LLR-ului, pe ordonată – număr de biți ce au valoarea LLR dată pe abscisă)	100
Figura 4.29 Performanțele TCS-ului	102
Figura 5.1 Structura generală a unui codor convoluțional multi-intrare, recursiv și sistematic, de rată $R/(R+1)$: configurația H	104
Figura 5.2 Structura generală a unui codor convoluțional multi-intrare, recursiv și sistematic, de rată $R/(R+1)$: configurația G	106
Figura 5.3 Codul RS 15/13 ₈ : formele echivalente H și G pentru codor	107
Figura 5.4 Codorul pentru $G=[2 \ 3]$	109
Figura 5.5 Comparație între spectrele distanțelor ale MBCC-lui propus în [DOB05] (cu $G=[15 \ 13 \ 11]$) și cel mai bun MBCC găsit printr-o căutare exhaustivă (cu $G=[13 \ 11 \ 9]$)	112
Figura 5.6 Structura unui MBTC	114
Figura 5.7 Exemplu de cod duo-binar	115
Figura 5.8 Histograme APP normalizate pentru algoritmul de decodare MAP	123
Figura 5.9 Histogramele APP normalizate pentru algoritmul de	

decodare MaxLogMAP	124
Figura 5.10 Performanțele BER și FER pentru două valori ale pragului μ	125
Figura 5.11 Eficiența criteriului de stop MLAPP (albastru-MAP cian-MAPgenial roșu-MaxLogMAP verde-MaxLogMAPgenial)	125
Figura 5.12 Rezultatele căutării codurilor componente cu $R=M=2$; Lungime întrețesere 752×2 biți; SNR=3.4dB; Roșu: $C_v \leq 3.5$; verde: $3.5 < C_v < 4.5$; negru: $C_v \geq 4.5$; alb: nedecodabile;	128
Figura 5.13 Performanțele codurilor componente selectate la memorie 2: albastru: $G_1=[7,3,5]$, roșu: $G_2=[7,6,5]$, verde: $G_3=[5,3,7]$, negru $G_4=[6,5,7]$	128
Figura 5.14 Rezultatele căutării codurilor componente cu $R=2$ $M=3$; Lungime întrețesere 752×2 biți; SNR=2.0dB; Roșu: $C_v \leq 3.5$; albastru: $3.5 < C_v < 3.7$; verde: $3.7 < C_v < 4.0$; negru: $C_v \geq 4.0$; alb: nedecodabile;	129
Figura 5.15 Performanțele BER și FER vs SNR ale MBTC-urilor de memorie 2 și 3	132
Figura 5.16 Rezultatele căutării codurilor componente cu $R=2$ $M=4$; Lungime întrețesere 752×2 biți; SNR=1.6dB; Roșu: $C_v \leq 3.5$; albastru: $3.5 < C_v < 3.7$; verde: $3.7 < C_v < 4.0$; negru: $C_v \geq 4.0$; alb: nedecodabile;	133
Figura 6.1 Codor convoluțional Multi-Non-Binar– schemă generală	136
Figura 6.2 Adunarea și multiplicarea în $GF(4)$	137
Figura 6.3 MNBTC – schema generală	138
Figura 6.4. Comparație între MNBTC-ul cu matricea generatoare $G=[2 \ 1 \ 2; 3 \ 2 \ 2; 0 \ 0 \ 2; 3 \ 1 \ 1]$ (tb: tail biting, zp: zero padding) și MBTC-urile (de memorie 3 și 4) propuse în [DOB05] peste un canal AWGN	143

Lista tabelelor

Tabelul 3.1 Valorile parametrilor ce indică (orientativ) volumul simulărilor	32
Tabelul 4.1 Polinoamele utilizabile în TC-uri	49
Tabelul 4.2 Parametrii BTC-lui utilizați în simulările pentru investigarea codurilor convoluționale componente	51
Tabelul 4.3 Parametrii SCCC-lui utilizați în simulări	54
Tabelul 4.4 Performanțele codurilor convoluționale în concatenare serială	55
Tabelul 4.5 Parametrii HCCC-lui utilizați în simulări	57
Tabelul 4.6 Performanțele codurilor convoluționale în concatenare hibridă	58
Tabelul 4.7 Parametrii interleaverelor investigate	65
Tabelul 4.8 Valorile IDS pentru un interleaver rectangular $M \times M$	71
Tabelul 4.9 Numărul minim de erori funcție de SNR	72
Tabelul 4.10 Valorile factorului de calitate pentru interleaverelor investigate	74
Tabelul 4.11 Parametrii TC-lui pentru analiza algoritmului MAP	76
Tabelul 4.12 Parametrii TC-lui pentru studiul algoritmilor de decodare	82
Tabelul 4.13 Variante de închidere a trellisului	84
Tabelul 4.14 Parametrii TC-lui utilizat la investigarea variantelor de închidere a trellisurilor	87
Tabelul 4.15 Matricile de puncturare	90
Tabelul 4.16 Parametrii TC-lui puncturat	91
Tabelul 4.17 Parametrii TC-lui utilizat pentru investigarea criteriului de stop	94
Tabelul 4.18 Timpul de decodare (fără criteriu de oprire)	98
Tabelul 4.19 Variația câștigului de codare cu numărul de iterații	98
Tabelul 4.20 Parametrii histogramelor LLR-urilor	99
Tabelul 4.19 Performanța TC-lui 5/7 funcție de pragul din criteriul de stop	101
Tabelul 4.20 Performanța TC-lui 15/13 funcție de pragul din criteriul de stop	101
Tabelul 4.21 Performanța TC-lui 25/31 funcție de pragul din criteriul de stop	101
Tabelul 5.1 Cardinalul seturilor G_0 , H_0 și $\xi(H_0)$	111
Tabelul 5.2 Matricele generatoare echivalente pentru cele două coduri de rată 2/3 propuse în [DOB05]	111
Tabelul 5.3 Codurile duobinare componente de memorie 3 selectate	129
Tabelul 5.4 Rezultatele căutării interleaverelor pentru MBTC-uri cu $M=3$	131

Lista acronimelor

APP	- A Posteriori Probability
AWGN	- Additive White Gaussian Noise
BCH	- (codul) Bose-Chaudhuri-Hocquenghem
BER	- Bit Error Rate
BPSK	- Binary Phase Shift Keying
BTC	- Binary Turbo-Code
CC	- Convolutional Code
CCSDS	- Consultative Committee for Space Data Systems
CD	- Compact Disc
C_n	- codor component numărul n
DBTC	- Double-Binary Turbo-Code
DEC_n	- decodorul component numărul n
DI, dilv	- de-interleaver
dmux	- demultiplexor
DQPSK	- Differential QPSK
DVB	- Digital Video Broadcasting
ETSI	- European Telecommunications Standards Institute
FEC	- Forward Error Correction
FER	- Frame Error Rate
G	- forma canonică observator
H	- forma canonică controlor
HCCC	- Hybrid Concatenated CC
I, ilv	- interleaver
IDS	- Interleaving Distances Spectrum
IP_n	- interleaver pseudo-aleator cu distanța minimă de întretesere egală cu n
IR32	- interleaver de tip rectangular cu dimensiunea 32×32 biți
$IsnA$	- interleaver de tip S cu $S=n$
IZ1, IZ0	- blocuri pentru inserare de zerouri
$L_c = +xx\%$	- varianta de estimare eronată a SNR-lui canalului cu $xx\%$
LFSR	- Linear Feedback Shift Register
LLR	- Logarithm Likelihood Ratio
LogMAP	- Logarithm MAP
LogMAPL	- LogMAP liniar
LogMAPR	- LogMAP rectangular
MAP	- Maximum A Posteriori
MaxLogMAP	- Maximum Logarithm MAP
MBTC	- Multi-Binary Turbo-Code
MBCC	- Multi-Binary CC
mlAPP	- minimum likelihood APP (criteriul de stop a iterațiilor după minimul valorilor APP)
mLLR	- minimum LLR (criteriul de stop a iterațiilor după minimul valorilor LLR)
MNBTC	- Multi-Non-Binary Turbo-Code
MNBTCtb	- MNBTC circular (tail-biting)
MNBTCzp	- MNBTC cu închidere la zero a primului trellis (zero-padding)
mux	- multiplexor
PCCC	- Parallel Concatenated CC
PGZ	- Peterson-Gorenstein-Zierler (decoding algorithm)

$PmCn$	- TC puncturat cu matricea de puncturare n și cod component m (vezi pag. IV.37)
$P1, P0$	- blocuri de puncturare
QPSK	- Quaternary Phase Shift Keying
r.i.e.	- rata iterațiilor efectuate (per bloc, din totalul de 15 iterații);
r.b.s.	- rata blocurilor stopate (procentul din numărul total de blocuri la care s-a aplicat criteriul de stop);
t.b.s.	- numărul total de blocuri simulate.
RRSN	- Redundant RNS
RS	- Reed-Solomon (code)
RSN	- Residue Number System
RSTC	- Reed-Solomon Turbo-Code
SCCC	- Serial Concatenated CC
S_m	- celula cu numărul " m " din componența codorului
SNR	- Signal to Noise Ratio
SOVA	- Soft Output Viterbi Algorithm
SPTSI	- suma ponderilor tuturor secvențelor de intrare
TC	- Turbo-Code
$TCCn$	- TC Circular în varianta n (variantele sunt descrise la pag. IV.34)
TCS	- TC dotat cu criteriu de stop a iterațiilor
$Tcxy\omega$	- TC în varianta de închidere a trellisurilor x (primul cod) y (al doilea cod) combinat cu varianta ω de inițializare a coeficienților recurențiali (din algoritmul MAP), (variantele sunt descrise la pag. IV.33)
$\pi 4dqTCP$	- schemă pentru simularea sistemului turbo-codat cu modulație DQPSK deplasat cu $\pi/4$, varianta cu puncturare
$\pi 4dqTCDB$	- schemă pentru simularea sistemului turbo-codat cu modulație DQPSK deplasat cu $\pi/4$, varianta cu DBTC
$\pi 4dq1TC$	- schemă pentru simularea sistemului turbo-codat cu modulație DQPSK deplasat cu $\pi/4$, varianta cu un BTC și multiplexare interioară
$\pi 4dq2TC$	- schemă pentru simularea sistemului turbo-codat cu modulație DQPSK deplasat cu $\pi/4$, varianta cu două BTC-uri și multiplexare exterioară
16QAM	- 16 Quadrature Amplitude Modulation
16QAM2TCDB	- schemă pentru simularea sistemului turbo-codat cu modulație 16QAM deplasat cu $\pi/4$, varianta DBTC
16QAM2TCP	- schemă pentru simularea sistemului turbo-codat cu modulație 16QAM deplasat cu $\pi/4$, varianta cu puncturare

Lista simbolurilor

a	- variabilă
A	- matrice de transformare a formei canonice H în forma canonică G
$A_k(s)$	- coeficienții recurenței „înainte” (forward) în algoritmul MaxLogMAP
$A_{M \times M}$	- matrice generatoare utilizată la închiderea trellisului MBTC-lor
b	- variabilă
B	- banda semnalului
$B_k(s)$	- coeficienții recurenței „înapoi” (backward) în algoritmul MaxLogMAP
$B_{M \times P}$	- matrice generatoare utilizată la închiderea trellisului MBTC-lor
C	- capacitatea canalului
C, \hat{C}	- constante
$C_{M \times M}$	- matrice generatoare utilizată la închiderea trellisului MBTC-lor
C_v	- convergența (TC-lor)
d	- variabilă
D	- variabila (argumentul) în cazul utilizării transformatei „ D ”
dC_v	- parametru utilizat în criteriul de căutare a codurilor componente / interleaverelor
$d(i, j)$	- distanța de întretesere dintre pozițiile i și j
d_{min}	- distanța minimă de întretesere
d_{max}	- distanța maximă de întretesere
$D_{M \times P}$	- matrice generatoare utilizată la închiderea trellisului MBTC-lor
e	- variabilă
E_b	- energia per bit
E_{bc}	- energia per bit codat
E_{bi}	- energia per bit de informație
$f(\cdot), F(\cdot)$	- funcții (oarecare)
\hat{f}	- estimatul lui f
$f_c(\cdot)$	- termenul de corecție în cazul algoritmului LogMAP
$F_x(\cdot)$	- funcția de repartiție a probabilității v.a. x
F_c	- factorul de calitate al unui interleaver
$g_c(\cdot)$	- termenul de corecție în cazul algoritmului LogMAP rectangular, aproximare a lui $f_c(\cdot)$
$g_r(D)$	- polinoam generator al codorului convoluțional (construit în forma canonică observator)
$g_{t,m}$	- coeficienții polinomului generator $g_r(D)$
$G, \hat{G}_0, \hat{G}_L, \hat{G}_F, \hat{G}_T$	- matrice generatoare aferente formei canonice observator
G_p	- matricea de puncturare
$h_c(\cdot)$	- termenul de corecție în cazul algoritmului LogMAP liniar, aproximare a lui $f_c(\cdot)$
$h_r(D)$	- polinoam generator al codorului convoluțional (construit în forma canonică controlor)
$h_{r,m}$	- coeficienții polinomului generator $h_r(D)$
H, H_0, H_L, H_F, H_E	- matrice generatoare aferente formei canonice controlor
i	- index
I	- mulțime de numere întregi
$I_{ex10} (I_{ex01})$	- informația extrinsecă de la DEC1 către DEC0 (de la DEC0 către DEC1)
$ids(\cdot)$	- funcția „spectru al distanțelor de întretesere” (IDS)

ids_{max}	- valoarea maximă a spectrului distanțelor de întrețesere
I_m	- integrală ce depinde de parametrul m
I_M	- matricea unitate de ordin M
$I_M(p)$	- matricea unitate de ordin M rotită spre stânga cu p poziții
$iter, iter_{max}$	- numărul de iterații, numărul maxim de iterații
j	- index
J	- mulțime de numere întregi
K, k	- variabilă număr întreg / index
k_c	- numărul de biți de informație, per moment de tact, furnizați codorului sau turbo-codorului
$L_j^{n,i}(d), Le_j^{n,i}(d)$	- probabilitățile (a posteriori – APP, respectiv extrinsecă, calculate de decodorul $j, j = 0$ sau 1 , la iterația i și pasul n) ca u^n să ia valoarea d
LLR_i	- secvența logaritmului raportului de plauzibilitate generată de decodorul i
$LLR(u_i)$	- logaritmul raportului de plauzibilitate pentru bitul u_i
Lm	- lungimea de mediere pentru medierea spectrului distanțelor de întrețesere
m	- index al celulelor de întârziere din componența codoarelor
m	- orginul distribuției Nakagami
M	- memoria codorului, numărul de celule de întârziere
$M_g(D)$	- matricea de transfer aferentă formei canonice G
$M_h(D)$	- matricea de transfer aferentă formei canonice H
n	- indexul tactului de bit, simbol sau cuvânt
N	- lungimea de întrețesere / lungimea secvenței de date
N	- mulțimea numerelor naturale
nb, Nb	- numărul de blocuri de date transmise
n_c	- numărul de biți (de informație + redundanți) generați, per moment de tact, la ieșirea codorului sau turbo-codorului
n_p	- perioada polinomului generator de reacție, respectiv a matricii de tranziție T
$nr[cond]$	- numărul cazurilor în care condiția $cond$ este îndeplinită
Nt	- numărul total de biți
$N_0/2$	- densitatea spectrală de putere a zgomotului AWGN
N_1	- număr de 1-uri
ρ	- lungimea de puncturare
P, P_1, P_2, P_3	- parametri definitorii ai interleaverului definit în [DOB]
$P\{ev\}$	- probabilitatea evenimentului ev
$\rho_N(\cdot)$	- densitatea de probabilitate Nakagami
P_s	- puterea semnalului util
$p_x(\cdot)$	- densitatea de probabilitate a v.a. x
P_z	- puterea zgomotului
q, Q	- variabilă
q	- indexul bitului din simbol, în cazul MNBTC-lor
Q	- numărul de biți per simbol, în cazul MNBTC-lor
r	- indexul intrării / ieșirii (turbo-)codorului
R	- numărul de intrări ale (turbo-)codorului
R_c	- rata de codare (binară)
S	- parametrul „distanță” al interleaverelor de tip S
s, \hat{s}	- stare actuală, viitoare

s^n	- vectorul stare al codorului de la tactul n
$s_m[n]$	- semnalul binar definit de ieșirea celei m
sd	- gradul de împrăștiere (spreading degree)
$S(D)$	- transformata „ D ” a semnalului (multi-non-)binar definit de succesiunea stărilor codorului
$SPTSI$	- suma ponderilor tuturor secvențelor de intrare a căror secvențe de ieșire corespunzătoare au ponderi egale
$supp$	- suportul spectrului distanțelor de întretesere
t	- variabilă (timpul)
T	- matricea de tranziție
T_b	- durata bitului
u	- secvența (de biți în cazul BTC-lor / de simboluri în cazul MBTC-lor / de cuvinte în cazul MNBTC-lor) de date de la intrarea turbo-codului
u^F	- porțiunea finală a secvenței de date afectată închiderii trellisului
$u_n, u[n]$	- bitul n în cazul BTC-lor / simbolul n în cazul MBTC-lor / cuvântul n în cazul MNBTC-lor, din secvența u
u_r^n	- bitul r din simbolul n în cazul MBTC-lor / simbolul r din cuvântul n în cazul MNBTC-lor, al secvenței u
$u_{r,q}^n$	- bitul q al simbolului r din cuvântul n din secvența u pentru MNBTC-uri
v_b	- viteza de transmisie binară (rata biților)
v_s	- viteza de semnalizare (rata simbolurilor)
$w^n, w[k]$	- eșantion din zgomot
x	- variabilă
x	- secvența (de biți în cazul BTC-lor / de simboluri în cazul MBTC-lor / de cuvinte în cazul MNBTC-lor) codată, emisă în canal (de la ieșirea turbo-codului)
$x_n, x[n]$	- bitul n în cazul BTC-lor / simbolul n în cazul MBTC-lor / cuvântul n în cazul MNBTC-lor, din secvența x
x_r^n	- bitul r din simbolul n în cazul MBTC-lor / simbolul r din cuvântul n în cazul MNBTC-lor, al secvenței x
$x_{r,q}^n$	- bitul q al simbolului r din cuvântul n din secvența x pentru MNBTC-uri
y	- secvența recepționată la ieșirea din canal, corespunzătoare secvenței x
$y_n, y[n]$	- componenta lui y corespunzătoare lui x_n
y_r^n	- componenta lui y corespunzătoare lui x_r^n
$y_{r,q}^n$	- componenta lui y corespunzătoare lui $x_{r,q}^n$
Y_1^n, Y_0^n	- termeni calculați pe baza secvenței recepționate și a dispersiei zgomotului σ^2
α	- variabilă aleatoare ce caracterizează fadingul
$\alpha(\)$	- coeficienții recurenței „înainte” (forward) în algoritmul MAP
$\beta(\)$	- coeficienții recurenței „înapoi” (backward) în algoritmul MAP
$\gamma(\)$	- metricile ramurilor în algoritmul MAP
$\Gamma(\)$	- funcția gamma
$\Gamma_k(s)$	- metricile ramurilor în algoritmul MaxLogMAP
η	- eficiență spectrală

16 Lista simbolurilor

Λ	- variabilă aleatoare dată de LLR
μ	- valoarea pragului din criteriul de stop al iterațiilor
$\pi(\cdot), \pi^{-1}(\cdot)$	- funcția de întretesere, respectiv de de-întretesere
σ^2	- dispersia / puterea zgomotului AWGN
ξ	- raportul semnal per zgomot în decibeli

1. Introducere

1.1 Motivația cercetării

Astăzi, turbo-codurile (TC) și, în general, structurile codate cu decodare iterativă au devenit o tehnologie matură și sigură, așa cum o dovedesc multiplele aplicații din sistemele de comunicații recente sau continua lor dezvoltare.

Deși fundamentele teoretice ale teoriei codării canalului au fost făcute de Claude Shannon cu aproape 60 de ani în urmă, [SHA48], introducerea TC-urilor de către Claude Berrou, Alain Glavieux și Punya Thitimajshima, [BGT93], la începutul anilor '90 este cea care a permis sistemelor de comunicații codate să funcționeze în apropierea limitelor teoretice. Astfel, în articolul de debut al TC-urilor, utilizând un cod component de memorie 5, un interleaver de 65 de mii de biți și o decodare iterativă cu 18 iterații, s-a obținut o rată de eroare $BER=10^{-5}$ la un SNR cu doar 0,7 dB mai mult decât limita indicată de C. Shannon pentru rata de codare de 1/2.

Incredibilele performanțe ale TC-lor au motivat aprofundarea cercetării lor. Au apărut astfel interleavere mai performante cum sunt S-interleaverele, [DOD95], sau interleaverul propus de colectivul de la Brest, [BSD04], noi algoritmi de decodare, precum SOVA, [HAP94], MaxLogMAP sau LogMAP, [RVH95], noi configurații de turbo-coduri, precum cele seriale și /sau hibride, [BDM97].

Trecerea în mileniul III a marcat în domeniu apariția TC-urilor cu intrări multiple (MBTC). Studiile făcute au arătat superioritatea acestei clase de TC-uri, [BJDK01]. În octombrie 2005, Catherine Douillard și Claude Berrou propun, [DOB05], două MBTC-uri concrete, de memorie 3 respectiv 4, ale căror performanțe FER se remarcă prin lipsa așanumitului fenomen „error-floor”. Practic, curba FER(SNR) a codului de memorie 4, propus în [DOB05], este paralelă cu limita Shannon pentru aceeași lungime a blocului de date. Acest rezultat reprezintă o performanță ce va fi greu de depășit deoarece „diferența” până la limita Shannon este sub 1 dB. În plus această performanță a fost obținută cu o complexitate moderată a decodării, deoarece s-au utilizat algoritmul de decodare MaxLogMAP și o cuantizare pe 4 biți a variabilelor de intrare în decodor. Ambele soluții, alese pentru ușurința implementării, diminuează performanța TC-lui.

Chiar dacă la ora actuală există TC-uri cu asemenea performanțe ca și cele amintite anterior, studiul TC-lor este departe de a fi finalizat. Motivele principale ce argumentează o astfel de afirmație sunt, pe de o parte, caracterul „aleator” impregnat TC-lui de către interleaver și, pe de alta, imensitatea de variante de TC ce se pot obține prin conjuncția codurilor componente potențiale cu inepuizabilele metode de întrețesere.

Afirmația anterioară a motivat, în parte, studiul, ale cărui rezultate sunt prezentate în această lucrare. Însă, principala motivație, pe care cu siguranță au avut-o miile de cercetători din întreaga lume, o reprezintă chiar spectaculozitatea performanțelor TC-lor: un TC este capabil să reducă rata erorii cu 4 sau chiar 5 ordine de mărime.



1.2 Scopul lucrării

- Obiectivele pe care le-am urmărit în studiul TC-lor au fost:
- construcția de modele de simulare conforme cu tipurile de canale de transmisiune reale cât și cu tipurile de modulație utilizate în practică. Aceste modele folosite în programele de simulare, sunt prezentate în Cap. 3. În acest capitol se prezintă o nouă metodă de generare a numerelor aleatoare cu distribuție Nakagami. Această nouă metodă a permis analiza, în premieră¹, a performanțelor TC-lor în canalele radio cu fading Nakagami, [BH21];
 - analiza parametrilor de care depinde funcționarea unui TC. În principal aceștia sunt configurația TC-lui, codul component, metoda de întrețesere și lungimea de întrețesere, metoda de puncturare și rata de codare, metoda de închidere a trellisului, tipul canalului de transmisie și al modulației utilizate, algoritmul de decodare, nivelul de cuantizare, numărul de iterații și criteriul de oprire(stop) utilizat. Aproape toți acești parametri au fost investigați din punctul de vedere al performanțelor BER / FER obținute în contextul TC-lui unibinar. Rezultatele obținute în urma acestor investigații sunt prezentate în capitolul 4;
 - căutarea unor noi metode de analiză a TC-lor. Între rezultatele acestor căutări se încadrează spectrul distanțelor de întrețesere (IDS), prezentat în capitolul 4, care reprezintă o nouă metodă de analiză a interleaverelor. De asemenea, în capitolul 5 se prezintă un nou criteriu de stop a iterațiilor adecvat MBTC-lor;
 - un alt scop a fost construcția RSTC-lor, TC-uri având, ca și coduri componente, codurile Reed-Solomon. Lucrarea de față prezintă o clasă de coduri construită în premieră în urma acestor căutări: MNBTC-le – turbo-codurile multi-non-binare care reprezintă o clasă de TC-uri foarte generală, care include atât MBTC-le cât și RSTC-urile. Această nouă clasă este prezentată în Capitolul 6.

Toate rezultatele experimentale prezentate în această lucrare au fost obținute prin programe de simulare proprii.

1.3 Structura lucrării

Lucrarea este structurată pe 7 capitole. În Cap. 2 sunt prezentate câteva aspecte ale problematicii codării canalului: un scurt istoric asupra metodelor de codare, limitele teoretice –deziderat al codării, motivarea denumirii de „turbo” date codurilor (convoluționale) concatenate paralel, o succintă prezentare a variantelor de TC.

În Cap. 3 se face o prezentare a modelelor urmate în simulări. Este vorba despre modelele canalului de transmisie, cât și a întregului sistem de transmisie. În modelul sistemului de transmisie s-au reținut doar blocurile componente care afectează în vreun fel (turbo-)codarea. Tot în Cap. 3 sunt date câteva detalii despre simulările efectuate (numărul de blocuri simulate ca funcție de BER, probabilitatea / rata de eroare ca și criteriu de oprire a simulărilor, relații de calcul a parametrilor distribuțiilor pe care le au secvențele de numere utilizate în simulări). În paragraful 3.4 se prezintă o nouă metodă de generare a numerelor aleatoare cu distribuție Nakagami.

Cap. 4 prezintă o analiză asupra performanțelor BER/FER a BTC-lor. Au fost analizate: configurațiile posibile de TC (paralelă, serială și hibridă), mulțimea

¹ Această investigație a fost făcută de colega mea Kovaci Maria.

codurilor convoluționale componente, recursive și sistematice, cu lungimea de constrângere mai mică de 7, algoritmi de decodare de tip MAP (MAP, Log MAP și MaxLogMAP), influența estimării canalului (mai precis a SNR-lui din canal) asupra performanțelor unui TC, modalitățile de închidere a trellisurilor pentru cele două coduri componente ale unui TC, influența puncturării și a ratei de codare asupra performanțelor BTC-lui, influența numărului de iterații precum și utilizarea unui criteriu de stop a iterațiilor asupra performanței BTC-lui. În paragraful 4.4 este tratată problematica întrețeserii utilizate în turbo-codare. Sunt prezentate câteva metode de întrețesere, o metodă originală de analiză a interleaveurilor prin intermediul spectrului distanțelor de întrețesere (IDS) și sunt calculate IDS-urile pentru interleaveurile de tip aleator și bloc.

În Cap. 5 au fost analizate turbo-codurile ce prezintă mai multe intrări binare (MBTC-le). Sunt prezentate două scheme de codare posibile pentru codurile convoluționale recursive și sistematice, având R intrări binare, de rată $R/(R+1)$,. Aceste două forme canonice ale codoarelor, cunoscute în literatură, [JOZ99], ca și „observator” și „controlor” (observer / controller), sunt analizate din punct de vedere al echivalenței și a numărului mulțimilor codoarelor generate. Deși cele două forme canonice sunt perfect echivalente în cazul BTC-lui clasic, în paragraful 5.2 este dovedită superioritatea formei canonice observator, numită G în prezenta lucrare, față de forma canonică controlor, numită H pentru o mai ușoară distincție în ecuații. În paragrafele 5.3 5.4 și 5.5 se analizează problemele specifice de construcție și decodare ale MBTC-lor. În paragraful 5.6 este prezentat un nou criteriu de stop a iterațiilor adecvat MBTC-lor. Paragraful 5.7 prezintă câteva aspecte legate de proiectarea MBTC-lor. Este propus un criteriu original de selecție a codurilor componente și a interleaveurilor. Acest criteriu se bazează pe convergența TC-lor, convergență definită ca număr mediu de iterații efectuate de TC pentru decodarea unui bloc recepționat. Pe baza acestui criteriu de selecție și utilizând forma canonică G s-au proiectat două turbo-coduri duo-binare (DBTC-uri) de memorie 3 cu performanțe FER superioare celui prezentat în [DOB05].

Cap. 6 prezintă o nouă familie de turbo-coduri: multi-non-binare. MNBTC-le reprezintă cea mai generală familie de turbo-coduri, ea incluzând atât MBTC-le care includ la rândul lor BTC-le, cât și clasa de turbo-coduri având coduri componente codurile Reed-Solomon, de asemenea propusă în premieră.

În Cap. 7 sunt prezentate, pe scurt, contribuțiile originale cuprinse în această teză.

2. Codarea canalului

2.1 Scurt istoric asupra codării canalului

Codarea canalului sau codarea FEC (Forward Error Correction) are scopul de a proteja informația transmisă în sistemele de comunicație digitale.

În 1948, în celebra sa lucrare „A mathematical theory of communication”, [SHA48], C. Shannon a demonstrat că este posibilă o comunicație sigură printr-un canal oricât de zgomotos dacă e îndeplinită condiția ca rata de transmisie să fie mai mică decât capacitatea canalului, și dacă se apelează la o codare infinită prin adăugarea de informație redundantă mesajelor transmise. Totuși C. Shannon nu a indicat soluțiile de codare a canalului. În plus, cu toate că redundanța adăugată face să crească întârzierea secvenței transmise, el nu a specificat întârzierea maximă ce poate fi tolerată pentru a fi posibilă comunicarea în apropierea limitei lui C. Shannon [HLY02].

Consecința teoremei lui C. Shannon a codării canalului zgomotos, este posibilitatea creării unui cod cu o lungime foarte mare a blocului de date pentru a se obține o performanță foarte bună. Din păcate complexitatea decodării crește exponențial cu lungimea blocului. Codurile create în următorii 50 de ani de diferiți cercetători au fost limitate ca performanță, de complexitatea decodării și de complexitatea canalului.

Unul dintre primele coduri corectoare de erori implementate practic a fost codul Hamming corector de o eroare, cod bloc propus în 1950, [HAM50]. Codurile convoluționale corectoare de erori datează din 1955, când au fost descoperite de P. Elias, [ELI50].

Mai târziu, J. Wozencraft și B. Reiffen, [WOZ57], [WOR61], ca și R. Fano și J. Massey, [FAN63], [MAS63], au propus diverși algoritmi pentru decodarea lor. Prima aplicație practică a codurilor convoluționale a fost propusă de J.A. Heller și I.M. Jacobs în anii '70, [HEJ71].

Un moment important în istoria codurilor convoluționale îl reprezintă descoperirea algoritmului de estimare a secvenței de probabilitate maximă de către A.Viterbi în 1967, [VIT67]. Un algoritm de decodare cu rata de eroare pe bit minimă a fost propus în 1974 de L. R. Bahl, J. Cocke, F. Jelinek și J. Raviv, [BCJR74], și a fost numit algoritmul „maximum a posteriori”-MAP. Algoritmul MAP, datorită complexității sale ridicate, a fost foarte rar folosit în practică până la descoperirea, în 1993, a turbocodurilor de către C. Berrou, A. Glavieux și P. Thitimajshima [BGT93].

Revenind la codurile bloc, codul Hamming corector de o eroare a avut performanțe prea scăzute pentru aplicații practice, dar se folosesc sub formă extinsă la memorii. În practică un moment important îl reprezintă descoperirea familiei de coduri bloc binare corectoare de erori multiple, cunoscute sub numele de BCH (Bose-Chaudhuri-Hocquenghem) în 1959 și 1960, [HOQ59], [BOC60]. În 1970, W. Peterson a arătat, [PET60], că aceste coduri prezintă o structură ciclică, adică toate versiunile permutate ciclic ale unui cuvânt de cod sunt de asemenea cuvinte de cod.

Prima metodă de construire a trellis-urilor, pentru coduri bloc liniare, a fost propusă de J. Wolf în 1978, [WOL78], având o mare complexitate, astfel că cercetarea / aplicarea lor s-a limitat la această categorie de coduri.

În 1988 G. Forney a arătat că unele coduri bloc au structuri trellis relativ simple [FOR88]. Motivați de această descoperire, B. Honary și G. Markarian au propus diverse metode pentru reducerea complexității asociate, [HOM97]. Algoritmul Chase este una dintre cele mai cunoscute tehnici propuse pentru decodarea codurilor bloc, algoritm propus în anul 1972, [CHA72].

În 1961 D. Gorenstein, N. Zierler, [GOZ61], au extins teoria codării binare pentru a putea fi tratate și codurile non-binare, unde simbolurile de cod sunt formate dintr-un număr de biți. Ei au propus de asemenea un algoritm, algoritmul Peterson-Gorenstein-Zierler (PGZ).

În 1960 o subcategorie foarte importantă de coduri BCH non-binare a fost descoperită de I. Reed și G. Solomon, numite coduri Reed-Solomon după numele celor care le-au creat, [RES60]. Aceste coduri au proprietăți optime, în sensul că, cuvintele de cod au distanța minimă, pentru o rată de cod dată, maxim posibilă. Aceasta nu garantează, oricum, obținerea unui BER minim. La decodarea codurilor RS binare poate fi folosit și decodorul PGZ.

Pentru codurile RS, algoritmi de decodare puternici au fost găsiți de E. Berlekamp [BER65], [BER68] și J. Massey, [MAS65], [MAS69]. În ultimii ani codurile RS și-au găsit aplicații practice la realizarea CD-playerelor și în transmisia video digitală, DVB (Digital Video Broadcasting) standardizată de ETSI.

În 1967 a fost introdusă o nouă clasă de coduri non-binare [SZT67], sistemul numerelor reziduale redundante (Redundant Residue Number System-RRNS), sistem avantajos pentru operații aritmetice rapide.

Un cod RRNS este un cod bloc ce maximizează distanța minimă și prezintă proprietăți de distanță similare cu codurile RS. R. Watson și C. Hastings, [WAH66], precum și H. Krishna ș.a., [KLS92a], [KLS92b], au exploatat proprietățile RRNS pentru detecția și corectarea unei singure erori și de asemenea pentru detecția de erori multiple.

La începutul anilor '70 codurile corectoare de erori au fost implementate în diferite sisteme de comunicații prin satelit și în anii '80 în sistemele radio mobile celulare. Totuși, pentru o lungă perioadă de timp codurile corectoare de erori și modulația au fost tratate ca subiecte diferite în sistemele de comunicații.

În 1987, prin integrarea acestor coduri și a modulației G. Ungerboeck, [UNG87a], [UNG87b], a propus modulația codată trellis (Trellis Coded Modulation), cu care se pot obține câștiguri de codare semnificative într-un mediu de transmisie de putere și bandă limitată.

Anul 1993 este un moment istoric de maximă importanță, fiind cel în care C. Berrou, A. Glavieux și P. Thitimajshima [BGT93] au propus în premieră turbo-codurile, care au permis operarea sistemelor de comunicații aproape de limita Shannon. De la recenta lor descoperire turbo-codurile au evoluat într-un ritm fără precedent și s-au consacrat în doar câțiva ani datorită eforturilor intense ale oamenilor de știință. Ca rezultat al acestei evoluții foarte rapide, turbo-codurile au fost deja introduse în sisteme standardizate ca de exemplu a III-a generație a sistemelor radio mobile 3G, [GBN01], [BDT03]. Performanțe impresionante s-au obținut cu turbo-codurile și în sistemele de transmisie video.

În schema propusă, de C. Berrou și colaboratorii săi, s-a folosit concatenarea paralelă a două coduri convoluționale sistematice recursive, având un dispozitiv de întreșesere între cele două codoare. La decodul s-a introdus o structură iterativă ce folosește o versiune modificată a algoritmului MAP clasic, [BCJR74],

pentru a decoda aceste coduri paralele concatenate. Începând cu 1993 s-au depus eforturi mari în vederea reducerii complexității decodului asociat. Un algoritm de decodare de complexitate redusă este, de exemplu, algoritmul Maximum-Logarithm-MAP propus de W. Koch și A. Baier, [KOB90], sau algoritmul Logarithm-MAP sugerat de P. Robertson, E. Willbrun și P. Hoher [RVH95].

Performanțele excelente ale turbo-codurilor au fost repede remarcate și datorită eforturilor lui S. Benedetto și G. Montorsi, [BEM96a], [BEM96b], respectiv L. Perez, J. Seghers și D. Costello, [PSC96], care au publicat lucrări în acest domeniu.

La mijlocul anilor '90 J. Hagenauer, E. Offer și L. Papke [HOP96] au extins conceptul la coduri bloc concatenate paralele. H. Nickl, J. Hagenauer și F. Burkett, [NHB97], au arătat că limita Shannon poate fi atinsă în cazul unui raport semnal/zgomot de 0,27 dB folosind un turbo-cod Hamming simplu, iar P. Jung și M. Nasshan, [JUN94], au studiat performanțele turbo-codurilor în funcție de constrângerea reprezentată de lungimea scurtă a cadrelor de transmisie, caracteristică sistemelor audio.

În 1998 S. M. Alamouti, [ALA98], a creat un cod bloc, spațiu-timp, care oferă o complexitate scăzută cu prețul unei degradări neînsemnate a performanței. Apoi, V. Tarokh, H. Jafarhani și A. Calderbank, [TJK99], au generalizat schema la un număr arbitrar de antene de emisie. Ulterior cercetările asupra codurilor spațiu-timp s-au extins de la canalele de bandă îngustă la canalele dispersive.

Trecerea în mileniul III este marcată, în domeniul codurilor protectoare, de introducerea turbo-codurilor multibinare de către colectivul de la Brest. Superioritatea noii familii de turbo-coduri este dovedită atât de performanțele BER cât și de micșorarea întârzierilor datorate întrețeserii, [DOB05].

2.2 Principiul „turbo”

Denumirea de „turbo-cod”, [BGT93], provine din faptul că ideea de bază a unui turbo-decodor este aceeași cu cea după care funcționează un motor termodinamic supra-alimentat prin turbină (motorul turbo), [HAG97]. În acest paragraf este prezentat succint principiul motorului turbo, ilustrat în Fig. 2.1, [BRW].

Pistonul 2, aflat în timpul IV (Evacuare) al ciclului termodinamic, elimină gazele arse prin supapa de evacuare, galeria de evacuare spre eșapament. Pistonul 1, aflat în timpul I (Admisie) al ciclului termodinamic absoarbe aer¹ din galeria de admisie. Volumul de aer înmagazinat în piston va influența în mod direct compresia, adică forța cu care va fi apăsător pistonul în timpul III (Ardere). Rolul turbinei atașată motorului este de a favoriza admisia unui volum de aer mai mare în piston, pe seama energiei gazelor evacuate. Astfel, gazele evacuate de pistonul 2, în drumul lor dinspre galeria de evacuare spre eșapament, antrenează turbina prin elicea ei din stânga, forțând-o să se învârtă la turații foarte mari. La rândul ei, turbina, prin elicea ei din dreapta, antrenează aerul în galeria de admisie, jucând rolul unei pompe de aer. Din galerie, aerul comprimat este forțat să intre în pistonul 1.

¹ aer (în cazul motoarelor cu aprindere prin compresie, ex: motorul diesel) sau amestec de aer cu vapori de benzină (în cazul motoarelor cu aprindere prin scânteie). În acest ultim caz amestecul provine din carburator.

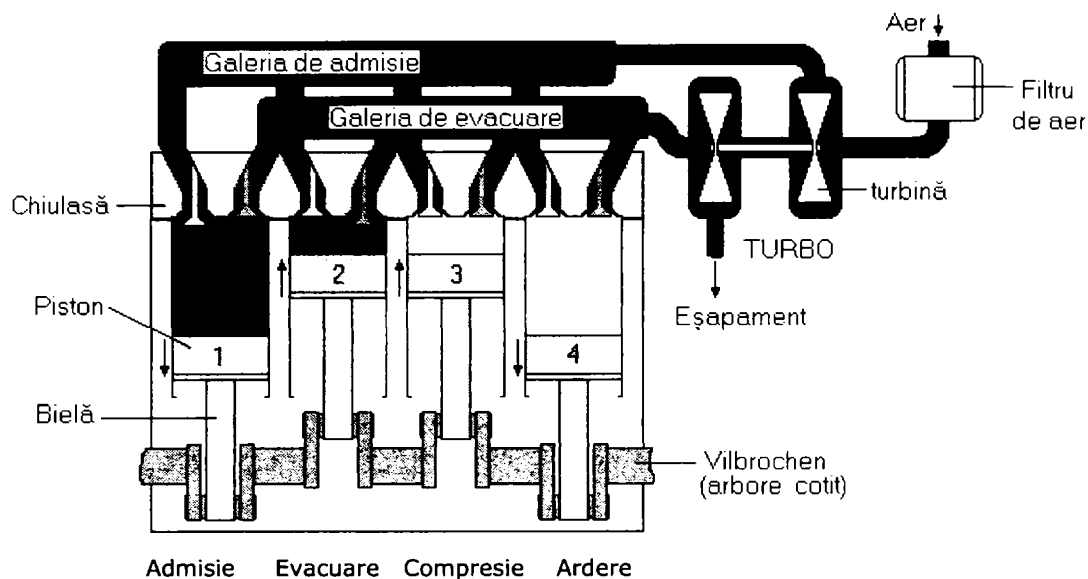


Figura 2.1 Ilustrarea principiului „turbo”

În concluzie, dacă în lipsa turbinei, pistonul (în timpul său de admisie) își „trage” aer prin depresiunea creată de el, în prezența turbinei admisia este mult amplificată de „pompa de aer” pe care o reprezintă turbina. Acest fapt conduce la o amplificare remarcabilă a puterii motorului turbo² datorită reutilizării energiei gazelor arse. La rândul său, pistonul 1, în timpul său IV, va antrena turbina, beneficiar fiind pistonul aflat în timpul de admisie (3 în cazul motorului din Fig. 2.1).

Asupra celor prezentate se impun două observații:

- 1) motorul supra-alimentat prin turbină poate fi asimilat cu un sistem cu reacție pozitivă; reacția este constituită de transferul energiei mecanice de la un piston la altul prin turbină;
- 2) întârzierea amplificării puterii, prin reacția pozitivă, întârziere ce are importanță la viteze mari³.

Atât principiul de funcționare al motorului turbo cât și cele două observații de mai sus sunt valabile și pentru decodorul turbo ce va fi prezentat în paragraful următor.

² ansamblul pieselor conținute în carcasa turbinei (turbina cu lagărele și sistemele ei de ungere) se numește „turbo”

³ această întârziere în reacție (necontrolabilă) a constituit motivația renunțării la motorul turbo în competițiile sportive.

2.3 Turbo-decodorul

În Fig. 2.2 se prezintă schematic structura unui turbo-cod, [BGT93] [BH5]. Secvența de informație, notată u , este codată de către codorul C1 rezultând secvența x_1 cu biți de paritate adăugați informației. Aceeași secvență de biți, u , este furnizată codorului C0, însă în altă ordine, obținută prin întreteserea realizată de dispozitivul de întretesere „I”. Codorul C0 generează la rândul său, secvența redundantă x_0 .

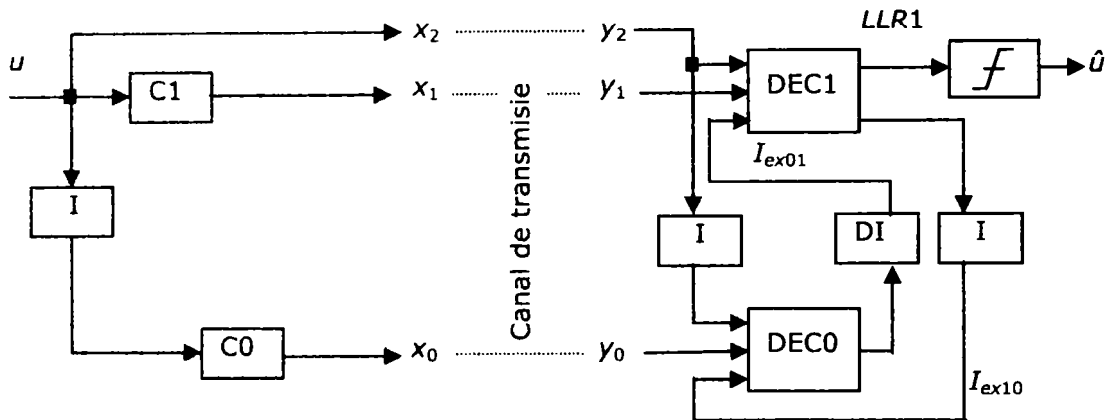


Figura 2.2 Codor / decodor pentru un turbo-cod (Cod concatenat paralel)

Codurile pe care le implementează C1 și C0 pot fi coduri bloc, [APR97], sau convoluționale, [HAG97]. Însă trebuie remarcat că, datorită necesității de a întretese secvența u (cu scopul de a decodifica secvențele x_1 și x_0), și codoarele convoluționale vor segmenta secvența de informație încât și codul convoluțional va putea fi privit ca și unul bloc.

Secvențele rezultate $x_2 = u$, x_1 și x_0 , după multiplexare și modulare (operații ce nu au fost simbolizate în Figura 2.2 pentru simplitate) constituie ieșirea turbo-codului, semnal ce va fi emis în canal. La ieșirea acestuia, prin demodulare și demultiplexare (operații de asemenea omise în Figura 2.2) rezultă secvențele recepționate corespunzătoare, y_2 , y_1 și y_0 (nebinare datorită zgomotului suprapus).

Aceste trei secvențe constituie intrarea turbo-decodorului. Turbo-decodorul cuprinde decodările convoluționale DEC1 și DEC0 corespunzătoare codoarelor C1 și C0. Prin corespondență y_2 și y_1 constituie intrarea decodorului DEC1, iar y_2 întretesută alături de y_0 , constituie intrarea decodorului DEC0.

Pe lângă aceste intrări fiecare decodor mai prezintă o a treia intrare, la care celălalt decodor îi furnizează așa numita informație extrinsecă. Conform acestor trei intrări, fiecare decodor calculează logaritmul raportului de plauzibilitate (Log Likelihood Ratio -LLR) pentru fiecare bit din secvența u :

$$LLR(u_i) = \ln \frac{p(u_i = 1 | y)}{p(u_i = 0 | y)} \quad (2.1)$$

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

Turbo-decodorul funcționează iterativ, după cum urmează:

1. se citește secvențele y_2 , y_1 și y_0 ;
2. pe baza secvențelor y_2 și y_1 , DEC1 generează informația extrinsecă care, întrețesută, constituie I_{ex10} –a treia intrare pentru DEC0;
3. pe baza secvențelor y_2 întrețesută, y_0 și I_{ex10} , DEC0 generează informația extrinsecă, care de-întrețesută constituie I_{ex01} –a treia intrare pentru DEC1;
4. din acest punct, fiecare decodor primește informație extrinsecă. Pe baza ei și a secvențelor venite din canal (y_2 și y_1 respectiv y_2 întrețesută și y_0) fiecare decodor furnizează la rândul-i informația extrinsecă. Acest proces se repetă iterativ de un anumit număr de ori (impus sau calculat, funcție de tipul turbo-codului);
5. după efectuarea tuturor iterațiilor impuse, se face o decizie hard asupra logaritmului raportului de plauzibilitate generat după ultima iterație de unul din cele două decodoare (în Figura 2.2 s-a ales $LLR1$). Secvența rezultată prin operația de decizie hard constituie ieșirea turbo-decodorului.

Observații: 1) Cele două decodoare cooperează asemănător cu pistoanele unui motor turbo: fiecare decodor furnizează celuiilalt informație extrinsecă așa cum pistoanele își furnizează reciproc energie mecanică prin turbină;

2) Dacă în cazul motorului turbo, mărimea fizică vehiculată este energia mecanică (cinetică), în cazul turbo-decodorului, decodoarele componente vehiculează între ele informație. I_{ex10} și I_{ex01} , fiind definite ca și logaritmul unui raport de probabilități (după cum se va arăta în paragraful 4.5), respectă definiția informației;

3) Decodorul turbo are nevoie de un anumit timp pentru a executa o decodare asupra secvenței recepționate, asemeni motorului turbo. Acest fapt se datorează, în principal întârzierii produse de dispozitivele de întrețesere, dar și de timpul necesar algoritmului de decodare să ruleze.

Deși denumirea de turbo-cod se referă strict la structura „paralel” propusă de inventatori, [BGT93], datorită asemănărilor conceptuale și a performanțelor, în literatură unii autori au inclus sub denumirea de turbo-cod și structurile „seriale” și cele „hibride”, cu toate că aceste structuri nu prezintă aceeași asemănare cu motorul turbo. În paragraful următor se prezintă câteva structuri de turbo-coduri analizându-se diferențele conceptuale și din punct de vedere al performanței, notând că, pentru ușurința expunerii, se va păstra denumirea de turbo-cod și pentru structurile seriale și hibride de concatenare a codurilor.

2.4 Configurații de turbo-coduri

Schema turbo-codului din Figura 2.2 poate fi generalizată, [WAD00], [BEM97], la un număr arbitrar de coduri concatenate în paralel. În Figura 2.3 se prezintă schema unui turbo-cod cu trei coduri componente concatenate paralel. Secvența de informație u este codată direct de către codorul C2 și, după o întrețesere prin I1, de către C1, respectiv după o întrețesere prin I0, de către C0. Secvența u și secvențele rezultate prin codare sunt emise în canal după multiplexare și modulare. La recepție, după operațiile inverse de demodulare și demultiplexare, secvențele recepționate sunt distribuite decodoarelor corespunzătoare. Fiecare decodor va beneficia de două seturi de secvențe de informații extrinseci, provenite

de la celelalte două decodare și la rândul-i va furniza informație extrinsecă. De remarcat că în acest caz rata de codare este $R_c=1/4$. Întârzierea produsă prin decodare este aceeași cu a turbo-decodorului din Figura 2.2, deoarece decodările lucrează în paralel. Dacă se dorește o rată de codare mai mare se poate apela la puncturare, cu prețul unei degradări în performanța ratei erorii.

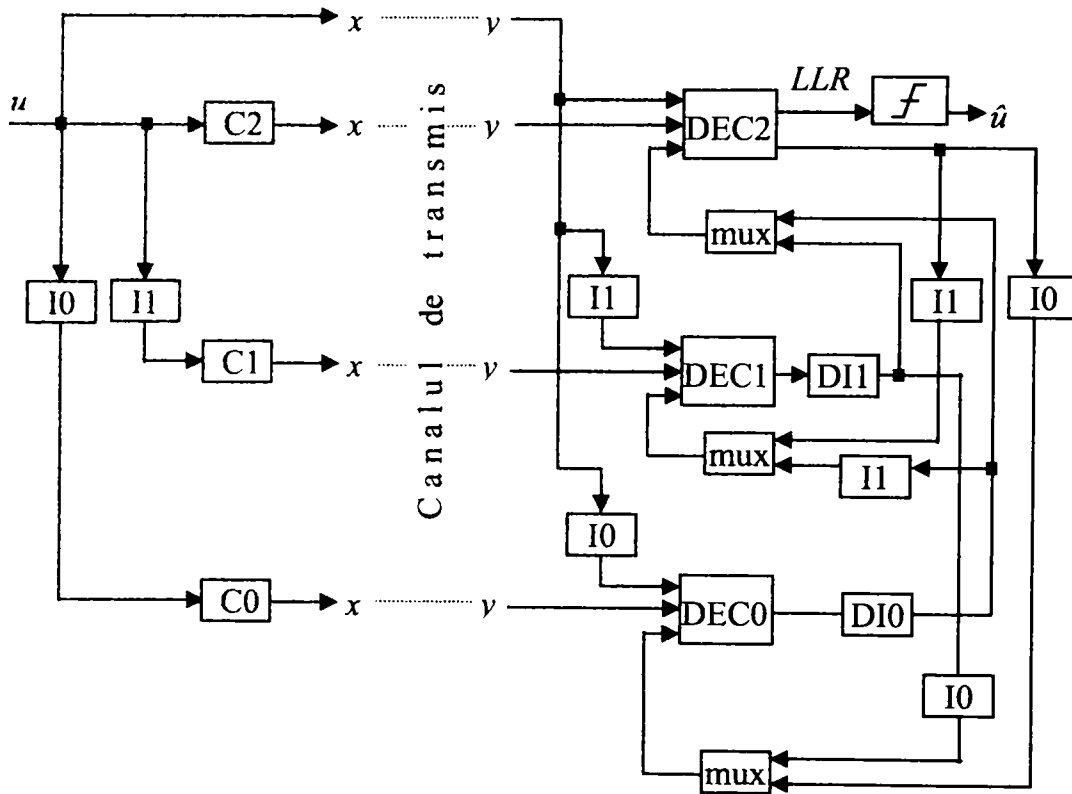


Figura 2.3 Codor/decodor pentru un turbo-cod triplu în structură paralelă

O astfel de structură de turbo-cod poartă denumirea de Cod Concatenat Paralel Multiplu.

Structura serială este o alternativă la structura paralelă a turbo-codurilor. În Figura 2.4 se prezintă un cod concatenat serial. La acesta, spre deosebire de cel paralel, secvența de informație u împreună cu cea rezultată prin codarea ei de către codorul C1, după o prealabilă multiplexare și întrețesere, se recodează cu ajutorul codorului C0. Secvențele de ieșire a turbo-codurilor sunt cele două secvențe, de la intrarea lui C0 și de la ieșirea sa. De remarcat că în acest caz rata de codare este $R_c=1/4$, iar lungimea necesară de întrețesere este dublă față de cazul paralel (deoarece secvența ce trebuie întrețesută este de lungime dublă față de cea a secvenței u). În plus codorul C0 execută o codare dublă ca volum față de codorul C1. La recepție, ambele secvențe rezultate din canal sunt preluate de DEC0. Acesta generează o informație extrinsecă asupra biților săi sistematici, care prin de-întrețesere constituie atât biții sistematici ai lui DEC1 cât și cei de paritate. Atât secvența notată „ y_1, y_2 ” cât și informația extrinsecă generată de DEC0, după

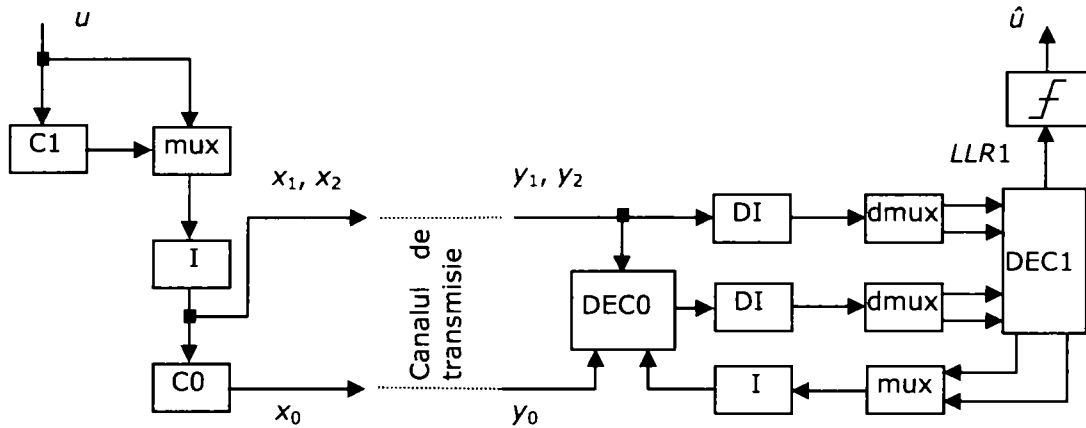
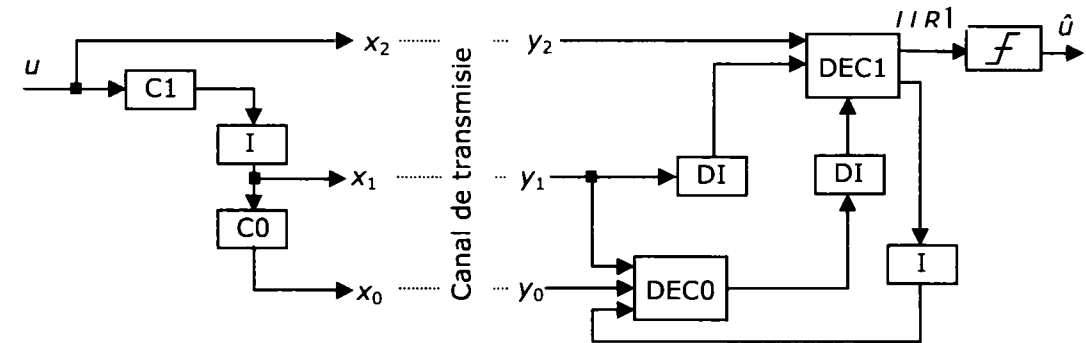
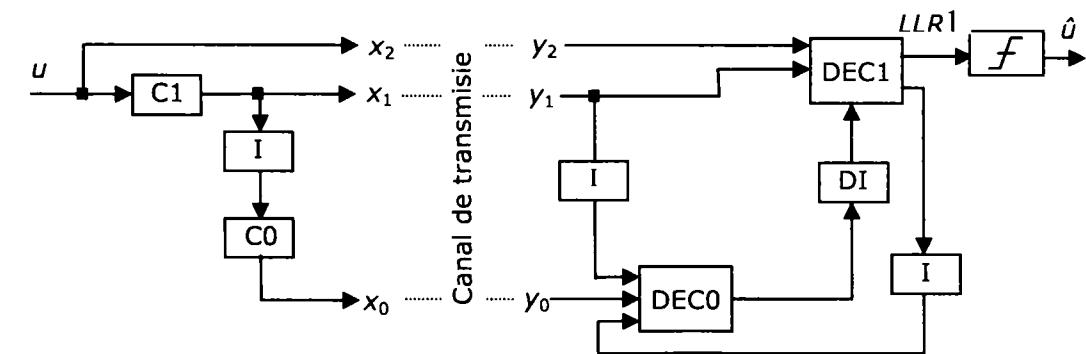


Figura 2.4 Codor/decodor pentru un turbo-cod în configurație serială



a)



b)

Figura 2.5 Două scheme de codoare pentru coduri corcâtenate hibrid

demultiplexare, sunt livrate decodoriului DEC1. Acesta, la rândul său, generează informație extrinsecă, care se referă atât la biții săi sistematici cât și la cei de paritate. Prin multiplexare și întrețesere cele două secvențe cu „informație extrinsecă” generate de DEC1, formează informația furnizată prin bucla de reacție decodoriului DEC0. Spre deosebire de configurația paralelă, unde era indiferent care decodor component genera ieșirea turbo-decodoriului, la configurația serială este mult mai comod să se aleagă DEC1. Altfel ar fi necesare adăugarea altor blocuri de demultiplexare și deîntrețesere, care ar introduce o întârziere suplimentară.

O astfel de structură poate fi și ea generalizată. Generalizarea ar consta în multiplexarea și întrețeserea secvențelor de la ieșirea turbocodului din Figura 2.4 în scopul unei a treia codări. O astfel de configurație multiplă are ca dezavantaj întârzierea, la codare și decodare, cauzată de operarea serie.

Cele două modalități de concatenare prezentate pot fi combinate într-o așa numită configurație hibridă. În Figura 2.5 se prezintă două variante de configurații hibride, ambele având la codare o structură serială, iar la decodare una paralelă. Cele două diferă doar prin faptul că cea de-a doua ieșire a turbo-codului, x_2 , este întrețesută la prima schemă iar la cea de-a doua schemă este luată direct de la ieșirea codului C1. Structura serială de la codare se reflectă în turbo-decodor prin faptul că decodorul DEC0 îi furnizează decodoriului DEC1 o informație extrinsecă despre biții de paritate și nu despre cei sistematici, ca în cazul configurației paralele. Din această cauză configurația hibridă este inferioară configurației paralele din punct de vedere al performanțelor BER, [WAD00].

În schemele prezentate anterior s-a subînțeles că secvența de intrare conține un singur bit într-o perioadă de tact. Generalizând la un număr arbitrar R de biți per tact se pot obține turbo-coduri cu o rată de codare și implicit o eficiență mai mare. Formal structurile turbo-codurilor sunt aceleași cu cele prezentate, diferența constând în faptul că legăturile dintre blocurile componente devin magistrale paralele de R biți. În plus crește complexitatea și volumul calculelor. O astfel de categorie o reprezintă turbo-codurile având $R=2$, numite „duo-binare” [JEZ03].

2.5 Limite teoretice în teoria informației

Teorema codării canalelor cu zgomot, fundamentată de către C. Shannon, [SHA48], afirmă că se poate face o transmisie cu o rată a erorii oricât de mică, dacă debitul (viteza) de transmisie este mai mică decât capacitatea canalului: $v_b \leq C$, unde v_b este viteza de transmisie binară iar C este capacitatea canalului, dată prin formula:

$$C = B \cdot \log_2 (1 + P_s/P_z) \quad (2.2)$$

unde B este banda canalului. Considerând zgomotul AWGN cu densitatea spectrală de putere $N_0/2$, rezultă pentru puterea zgomotului valoarea:

$$P_z = \sigma^2 = 2 \cdot B \cdot N_0/2 = B \cdot N_0 \quad (2.3)$$

unde σ^2 reprezintă dispersia sau puterea zgomotului AWGN. Puterea semnalului, P_s se poate calcula astfel:

$$P_s = E_b/T_b = v_b \cdot E_b \quad (2.4)$$

unde E_b sunt energia necesară per bit și T_b durata sa.

Considerând acum [VUY00] că transmisia s-ar face la capacitatea canalului, $v_b = C$, rezultă, ținând cont de relațiile (2.3) și (2.4):

$$\frac{C}{B} = \log_2 \left(1 + \frac{C}{B} \cdot \frac{E_b}{N_0} \right) \quad (2.5)$$

Din ultima relație rezultă pentru raportul E_b/N_0 expresia:

$$\frac{E_b}{N_0} = \frac{2^{C/B} - 1}{C/B}. \quad (2.6)$$

Dacă $B \rightarrow \infty$, din ecuația (2.6) rezultă pentru raportul semnal per zgomot, E_b/N_0 , valoarea minimă necesară:

$$\left(\frac{E_b}{N_0} \right)_{min} = \ln 2 = -1,59 \text{ dB} \quad (2.7)$$

Observații: 1) Mărima $\eta = v_b/B$ se numește eficiență spectrală [VUY00] și este o măsură a eficienței transmisiei. Dacă se utilizează o modulație cu 2^q niveluri, iar rata de codare este

$R_c = k_d/n_c$, atunci viteza de semnalizare este:

$$v_s = q \cdot R_c \cdot v_b \quad (2.8)$$

și, la limită (pentru respectarea teoremei lui Shannon: $v_s = B$):

$$\eta_{max} = q \cdot R_c \quad (2.9)$$

2) Trebuie remarcat că, în cazul introducerii redundanței (codării FEC), există o diferență între puterea semnalului necodat și cel codat, [WAD00]. Astfel:

$$k_c \cdot E_b = n_c \cdot P_s \cdot T_b \quad (2.10)$$

unde raportul k_c/n_c reprezintă eficiența codării, sau raportul dintre numărul de biți de informație și numărul total de biți. Atunci, pentru o densitate spectrală de putere a zgomotului $N_0/2$ și o modulație BPSK (± 1):

$$\frac{E_b}{N_0} = \frac{P_s \cdot T_b}{R_c \cdot N_0} = \frac{P_s}{2 \cdot R_c \cdot \sigma^2} = \frac{1}{2 \cdot R_c \cdot \sigma^2}. \quad (2.11)$$

3. Simularea sistemului de transmisie

3.1 Metoda de simulare Monte Carlo

Simularea funcționării diferitelor sisteme, în vederea verificării unor ipoteze asupra funcționării, prin măsurarea performanțelor sau a eficienței unor noi metode sau soluții tehnice, este o alternativă mai convenabilă, comparativ cu implementarea practică, cu condiția ca simularea să reflecte realitatea corect.

În majoritatea cazurilor, scopul implementării practice a unui sistem, sau, alternativ, al simulării funcționării sale, este de a măsura un anume parametru, y , (sau a unei funcții de acesta, $f(y)$) generat prin funcționarea respectivului sistem. Problema implementării sau, alternativ a simulării, se pune, în mod evident, deoarece funcționarea respectivului sistem depinde de parametri aleatori, x_i , cu $i \in I = \{1, 2, \dots, m\}$. Parametrii y și x_i , sunt, în general, funcții de timp, $y(t)$ și $x_i(t)$.

Simularea funcționării sistemului, în vederea măsurării (estimării) lui $y(t)$ sau a unei funcții de acesta $f(y(t))$, presupune, [JBS]:

i) crearea unui model pentru sistemul respectiv, având pe $x_i[n]$ drept variabile de intrare și pe $y[n]$ ca variabilă de ieșire (Figura 3.1).

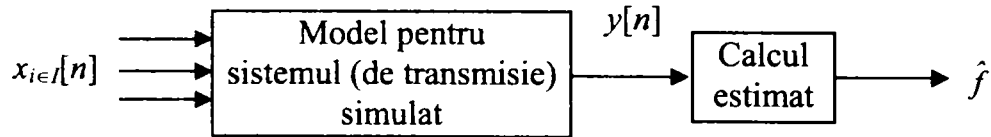


Figura 3.1 Ilustrarea metodei de simulare Monte Carlo

Elaborarea modelului înseamnă realizarea unei dependențe de forma:

$$y[n] = F(x_1[n], x_2[n], \dots, x_m[n]) = F(x_{i \in I}[n]); \quad (3.1)$$

ii) generarea unui set de m secvențe de intrare, $x_{i \in I}[n]$, cu $n=1 \div N$. Generarea secvențelor de numere, eşantioanele $x_{i \in I}[n]$, trebuie făcută după distribuții particulare, care să reproducă cât mai fidel comportarea parametrilor reali.

iii) calculul secvenței de ieșire corespunzătoare, $y[n]_{n=1 \div N}$, utilizând relația (3.1). Pe baza acestor valori se poate calcula un estimat al lui y sau a funcției dorite, $f(y)$, prin relația:

$$\hat{f} = \frac{1}{N} \cdot \sum_{n=1}^N f(y[n]) \quad (3.2)$$

Metoda de simulare descrisă anterior, [JBS], este cunoscută sub denumirea de *metoda Monte Carlo*, după numele celebrului oraș, sediu al celor mai renumite cazinouri din lume, ea fiind aplicată, la început, pentru simularea jocurilor de noroc.

3.2 Modele și metode de simulare a sistemelor de transmisie ce utilizează turbo-codare

3.2.1 Introducere

Datorită scopului de a simula sisteme de transmisie ce utilizează turbo-coduri pentru protecție împotriva erorilor, modelul de sistem din Figura 3.1 trebuie modificat în sensul adăugării blocurilor de codare și decodare turbo. Se obține astfel modelul din Figura 3.2.

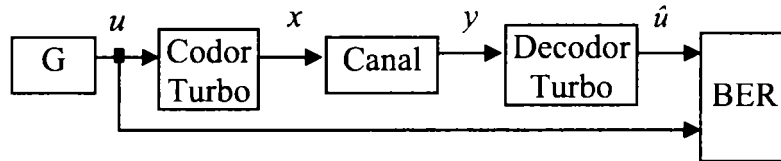


Figura 3.2 Model pentru simularea unui sistem de transmisie turbo-codat

Astfel, un program de simulare pentru analiza performanțelor unui turbo-cod trebuie să:

- genereze date binare, u , în forma adecvată turbo-codului (blocuri de biți sau caractere, după cum turbo-codul este uni- sau multi-binar, de lungime egală cu lungimea de întrețesere);
- turbo-codeze aceste secvențe binare;
- simuleze trecerea secvențelor turbo-codate, x , prin canalul de transmisie ales;
- execute turbo-decodarea conform algoritmului dorit;
- calculeze rata erorii de bit, BER (Bit Error Rate) și/sau a altor parametri, de exemplu FER (Frame Error Rate).

În programele de simulare a căror rezultate sunt prezentate în lucrarea de față datele binare se aleg aleator, după procedeul ce este descris în continuare. Se generează o secvență u de N numere u^n , $n=1, 2, \dots, N$, cu distribuție uniformă în intervalul $[0, 1]$. Prin aplicarea transformării:

$$u \rightarrow [1.99 \cdot u] \quad (3.3)$$

unde $[]$ semnifică operația de trunchiere la partea întreagă, se obține o secvență binară de N biți (N fiind fixat prin lungimea dispozitivului de întrețesere și a metodei de turbo-codare).

Secvența de date se turbo-codează, obținându-se secvența x , conținând N/R_c biți, unde R_c reprezintă rata de turbo-codare binară ($R_c < 1$). Această secvență corespunde unui singur bloc turbo-codat.

Pentru a studia performanțele turbo-codurilor, ceea ce interesează relativ la canalul de transmisie utilizat este doar secvența de ieșire, y . Paragrafele următoare prezintă modelele de canal utilizate în programele aferente prezentei lucrări. De notat că secvența de intrare în turbo-decodor, y , trebuie să fie în forma „soft”, altfel spus decizia în vederea reconstrucției secvenței binare aparține turbo-decodorului și nu demodulatorului din componența canalului binar. Din această cauză, modelele de canal binar nu vor conține și blocul de decizie.

Turbo-decodorul conține cel puțin două decodare componente și are o structură paralelă sau serială. După decodarea fiecărui bloc recepționat este contorizat numărul de erori.

Volumul simulărilor necesare pentru calculul BER depinde de SNR. Pentru SNR-uri mici (sub 0.7 dB), unde numărul de erori per bloc este mare, trebuie transmise puține blocuri. Pe măsură ce SNR-ul crește este necesar a se simula transmisia unui număr mai mare de blocuri, astfel încât să se obțină suficiente erori pentru a avea o precizie bună a curbei BER, [BH6].

S-au utilizat două criterii de oprire ale simulărilor. Într-un prim criteriu, se efectuează transmisia până când se depășește un anumit număr de erori ce depinde de BER ca și Tab. 3.1, unde N semnifică lungimea de întrețesere.

Tabelul 3.1 Valorile parametrilor ce indică (orientativ) volumul simulărilor

BER impus	Nr. erori		Nr. blocuri		SNR(dB)	
	necesar	ales	necesar	ales	$N=1784,$ $M=2$	$N=1784,$ $M=3$
1.0	N	1000×N	1	1.000	-	-
0.5	N	1000×N	2	2.000	-	-
0.2	N	600×N	5	3.000	-	0.0
0.1	N	400×N	10	4.000	0.0 0.1	0.1
0.05	N	250×N	20	5.000	0.2	0.2
0.02	N	120×N	50	6.000	0.3	0.3
0.01	N	70×N	100	7.000	0.4	0.4
5E-3	N	40×N	200	8.000	0.5	0.5
2E-3	N	18×N	500	9.000	0.6	0.6
1E-3	N	10×N	1.000	10.000	0.7	0.7
5E-4	N	6×N	2.000	13.000	0.8	-
2E-4	N	3,2×N	5.000	16.000	-	-
1E-4	N	2×N	10.000	20.000	0.9	-
5E-5	N	1,5×N	20.000	30.000	-	0.8
2E-5	N	N	50.000	50.000	1.0	-
1E-5	N	N	100.000	100.000	-	0.9
5E-6	N	N	200.000	200.000	1.1	-
2E-6	N	N	500.000	500.000	-	-
1E-6	N	N	1.000.000	1.000.000	1.2	1.0
5E-7	N	N	2.000.000	2.000.000	-	-
2E-7	N	N	5.000.000	5.000.000	-	-
1E-7	N	N	10.000.000	10.000.000	-	-

O altă cale este să se fixeze apriori numărul de blocuri. Această cale s-a abordat atunci când, spre exemplu s-au comparat rezultatele BER după fiecare iterație. Bineînțeles că BER-ul nu este apriori cunoscut, dar fie este estimat, fie, după primele simulări, se deduce volumul necesar al simulărilor. Spre exemplu, în coloanele șase și șapte ale tabelului sunt indicate valorile SNR-ului corespunzător pentru un turbo-cod ce are coduri componente de memorie $M=2$, respectiv $M=3$, la o lungime de întrețesere $N=1784$, [CCS]. Astfel, la memorie 2, pentru SNR = 0.0 dB

și 0.1 dB, s-a simulat transmiterea a 4000 de blocuri, pentru 0.2 dB s-a simulat transmiterea a 5000 de blocuri, ... iar la 1.2 dB un milion de blocuri. Peste această valoare a SNR-ului simulările nu-și găsesc rostul întrucât BER-ul este deja suficient de mic.

Menționez faptul că, dacă foloseam eșantionarea de tip „importance sampling”, volumul de simulări ar fi fost mult mai mic.

Indiferent de criteriul de stop al simulărilor, curbele BER s-au trasat cu o rezoluție de 0,1 dB, cu pornire de la 0 dB. Valoarea maximă a SNR-ului a fost stabilită funcție de BER-ul obținut (în general până sub 10^{-6}).

Exceptând cazul curbelor BER construite după fiecare iterație, în toate celelalte cazuri (când parametrul investigat este altul decât numărul de iterații) s-a utilizat un criteriu de oprire al iterațiilor. Astfel, atunci când toate valorile vectorului LLR (Log Likelihood Ratio) aferent unui bloc (în curs de decodare) sunt în modul sub un anumit prag, nu se mai efectuează restul de iterații rămase.

În continuare se vor prezenta modele pentru blocul „canal” din schema din Fig. 3.2 în conformitate cu modulația utilizată. Sunt prezentate câteva dintre cele mai uzuale tipuri de modulații: BPSK/QPSK, DQPSK defazat cu $\pi/4$, 16QAM. De asemenea, s-a avut în vedere atât canalul AWGN cât și canalul cu fading plat (Rayleigh, Rice sau Nakagami).

3.2.2 Modulație BPSK și canal AWGN / canal fading plat

În Fig. 3.3 a) este prezentat un model pentru simularea funcționării unui sistem de transmisie cu modulație BPSK printr-un canal cu fading plat și zgomot AWGN fără codare iar în Fig. 3.3 b) modelul pentru simularea funcționării sistemului de transmisie cu modulație BPSK printr-un canal cu fading plat și zgomot AWGN turbo codat.

Practic trebuie simulate următoarele operații:

– conversia secvenței binare unipolară într-una bipolară. Această operație este sugerată de blocul notat „±” în Fig. 3.3a), iar relația aferentă este:

$$x_k = 2 \cdot d_k - 1, \quad k = 1 \div N/R_c \quad (3.4)$$

unde: d_k este amplitudinea/nivelul de tensiune înainte de conversie și x_k este secvența turbo-codată, având lungimea de $1/R_c$ ori mai mare (R_c –rata de turbo-codare) decât lungimea de întrețesere, N .

– efectul trecerii prin canal a semnalului util. Acest efect este sintetizat de relația:

$$y_k = \alpha_k \cdot x_k \cdot \sqrt{E_{bc}} + w_k \quad (3.5)$$

în care E_{bc} este energia per bit transmis din secvența codată iar $\alpha_k = 1$ în cazul canalului AWGN, respectiv o variabilă aleatoare cu $\overline{\alpha^2} = 1$, în cazul canalului cu fading plat (Rayleigh, Nakagami sau Rice), [JBS00], iar w_k reprezintă un eșantion din zgomotul AWGN.

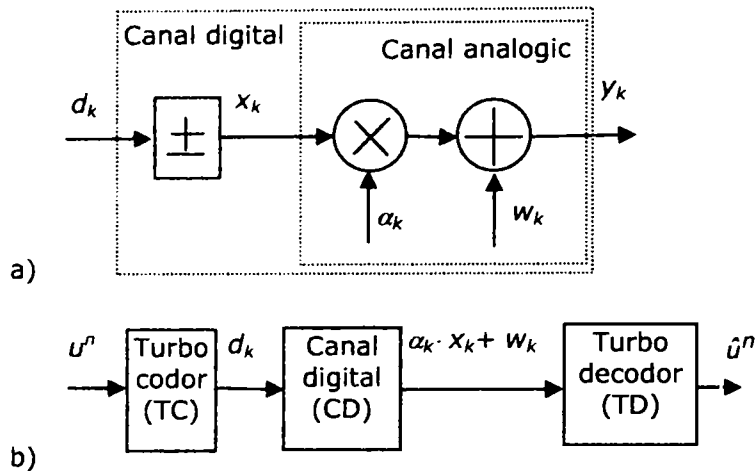


Figura 3.3 Modele de simulare: a) sistem de transmisie necodat cu modulație BPSK și canal cu fading plat și zgomot AWGN; b) sistemul de transmisie turbo-codat.

Alegând:

$$E_{bc} = 1 = R_c \cdot E_{bi} \quad \text{și} \quad T_b = 1, \quad (3.6)$$

unde E_{bi} reprezintă energia per bit din secvența de informație iar R_c rata de codare, rezultă că „puterea” conținută în secvența de numere $\alpha_k \cdot x_k \cdot \sqrt{E_{bc}}$ este unitară:

$$P_s = \frac{1}{N_t} \cdot \sum_{k=1}^{N_t} a_k^2 \cdot x_k^2 \cdot E_{bc} = \frac{1}{N_t} \cdot \sum_{k=1}^{N_t} a_k^2 = \overline{a_k^2} = 1, \quad (3.7)$$

unde $N_t = n_b \cdot N / R_c$ reprezintă numărul de biți transmiși, adică egal cu produsul dintre numărul de blocuri transmise, n_b , și numărul de biți dintr-un bloc, N/R_c . Ceea ce rămâne de aflat este dispersia sau puterea zgomotului:

$$\overline{w^2} = \frac{1}{N_t} \cdot \sum_{k=1}^{N_t} w_k^2. \quad (3.8)$$

Aceasta se află impunând un anume raport semnal per zgomot în decibeli:

$$\xi_0 = SNR_{dB} \stackrel{def}{=} 10 \cdot \lg \left(\frac{E_{bi}}{N_0} \right). \quad (3.9)$$

unde N_0 este dublul densității spectrale de putere a zgomotului AWGN considerat. Raportul semnal per zgomot de la ieșirea canalului simulat este:

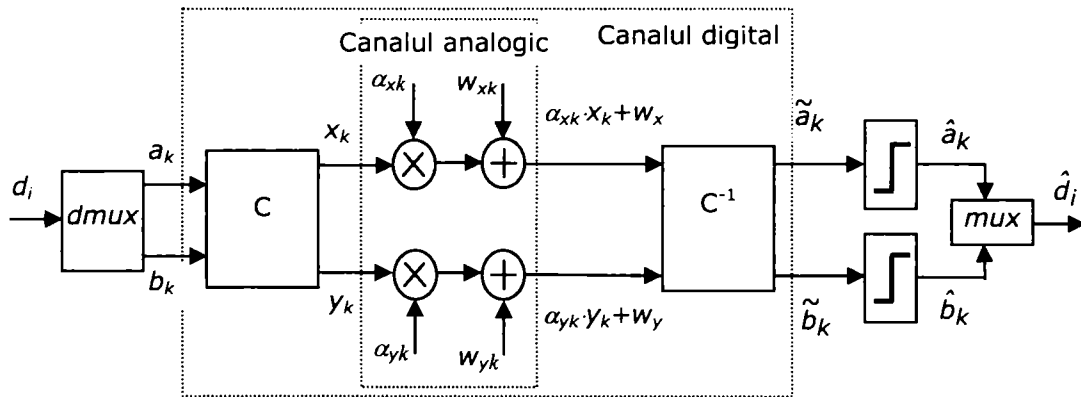
$$SNR = \frac{\overline{a^2} \cdot \overline{x^2} \cdot E_{bc} / T_b}{\overline{w^2} / T_b} = \frac{1}{\overline{w^2}} = \frac{R_c \cdot E_{bi}}{N_0 / 2} = 2 \cdot R_c \cdot 10^{\xi_0/10}. \quad (3.10)$$

În concluzie, w_k este o secvență de numere aleatoare cu distribuție gaussiană, de medie nulă și dispersie σ^2 , dată prin relația:

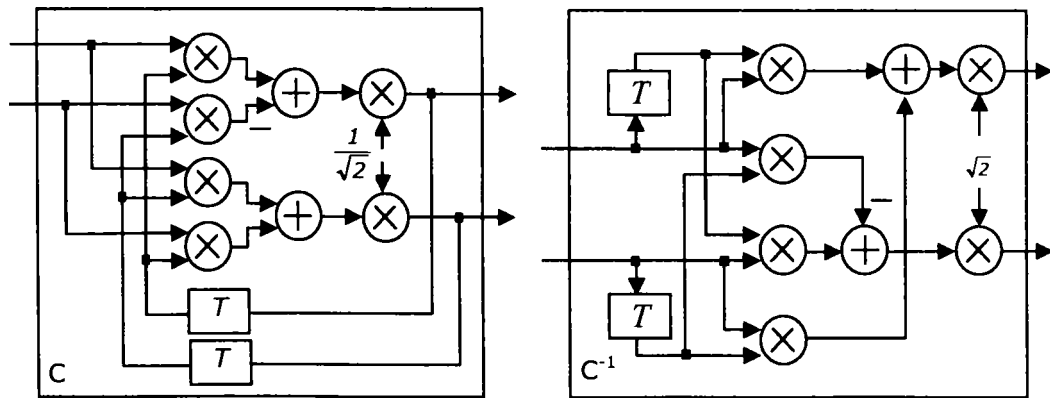
$$\sigma^2 = \overline{w^2} = \frac{1}{2 \cdot R_c \cdot 10^{\xi_0 / 10}} \quad (3.11)$$

3.2.3 Modulație DQPSK defazat cu $\pi/4$ și canal AWGN / canal fading plat

Modelul „canalului digital” AWGN / fading plat cu modulație DQPSK deplasat cu $\pi/4$ este prezentat în Fig. 3.4a. Acest tip de modulație este des utilizat în transmisiile prin canale cu fading, unde de asemenea se utilizează turbo-codarea. O prezentare mai detaliată a acestui tip de modulație a fost făcută în [BH2]. Pentru modelare, sunt esențiale relațiile de calcul ale amplitudinilor componentelor în fază (x_k) și în cuadratură (y_k), pornind de la valorile dibitului de intrare (a_k, b_k):



a) schema bloc;



b) blocurile C și C⁻¹ –detaliu constructiv;

Figura 3.4 Modelul canalului digital cu fading plat și modulație DQPSK deplasat cu $\pi/4$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \frac{\sqrt{2}}{2} \begin{bmatrix} a_k \cdot x_{k-1} - b_k \cdot y_{k-1} \\ a_k \cdot y_{k-1} + b_k \cdot x_{k-1} \end{bmatrix} \quad (3.12)$$

și, respectiv, cele de reconstrucție ale valorilor binare:

$$\begin{bmatrix} a_k \\ b_k \end{bmatrix} = \sqrt{2} \cdot \begin{bmatrix} x_k \cdot x_{k-1} + y_k \cdot y_{k-1} \\ -x_k \cdot y_{k-1} + y_k \cdot x_{k-1} \end{bmatrix}. \quad (3.13)$$

Considerând valorile ± 1 pentru a_k și b_k , puterea secvențelor x_k și y_k este $1/2$. Pentru calculul puterii secvențelor de zgomot w_{xk} și w_{yk} ce trebuie simulate, în ipoteza cunoașterii ratei de turbo-codare și a raportului semnal per zgomot $\xi_0(\text{dB})=10 \cdot \lg(E_b/N_0)$, rezultă:

$$\overline{w_x^2} = \frac{N_0}{2 \cdot E_{bc}} \cdot \overline{a^2} \cdot \overline{x^2} = \frac{N_0}{4 \cdot R_c \cdot E_{bi}} = \frac{1}{4 \cdot R_c \cdot 10^{\xi_0/10}}. \quad (3.14)$$

La fel ca și în cazul paragrafului precedent, $\alpha_k=1$ pentru canal AWGN sau α_k este o variabilă aleatoare Rayleigh, Nakagami sau Rice cu putere unitară ($\overline{a^2} = 1$) pentru canal cu fading plat.

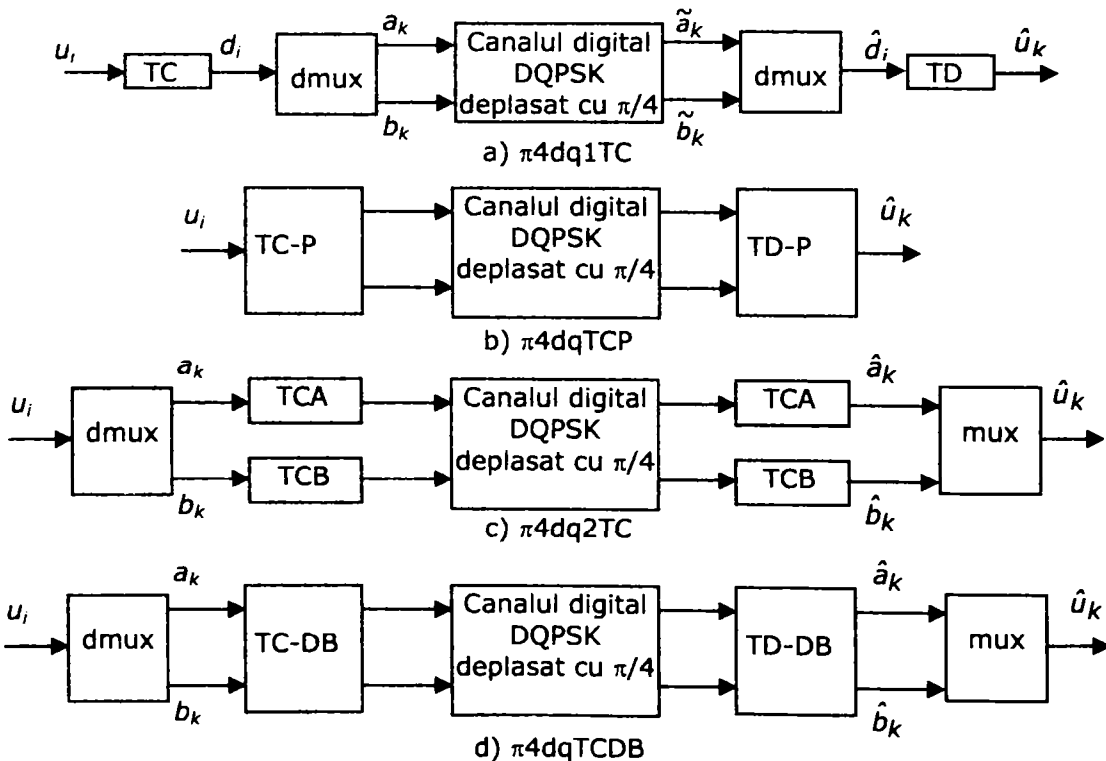


Figura 3.5 Modele de simulare pentru sisteme de transmisie cu modulație DQPSK deplasat cu $\pi/4$

Spre deosebire de cazul modulației BPSK, însă, simularea sistemului turbo-codat cu modulație DQPSK deplasat cu $\pi/4$ implică câteva probleme, [BH11]. În primul rând turbo-codul, în varianta sa clasică, acceptă la intrare o singură secvență și nu două. Această secvență nu poate fi d_i decât dacă blocurile de decizie apar după multiplexor (la decodare), fapt impus de necesitatea ca turbo-decodorul să opereze cu secvențe soft. Se obține astfel schema din Fig. 3.5a).

Pentru identificare această configurație va fi notată în continuare cu acronimul $\pi 4dq1TC$. Avantajul acestei scheme este că permite utilizarea oricărui turbo-cod. Un dezavantaj ar fi prezența demultiplexorului cât și, mai ales, a multiplexorului (care ar trebui, într-o eventuală implementare practică, să opereze cu caractere și nu cu biți). Această complicație ar putea să fie evitată dacă s-ar utiliza un turbo-cod cu o intrare și două ieșiri. O astfel de configurație poate fi obținută prin puncturare (turbo-cod de rată 1/2). Se obține astfel schema din Fig. 3.6.b), $\pi 4dqTCP$. Alte variante de turbo-codare sunt prezentate în Fig. 3.6c) și d). Modelul din Fig. 3.6c), notat $\pi 4dq2TC$, permite utilizarea oricărui TC, dublează viteza de calcul dar și complexitatea. Se evită totuși multiplexarea de caractere. Fig. 3.6d) sugerează utilizarea unui TC duo-binar ($\pi 4dqTCDB$) în locul celor două turbo-coduri clasice din Fig. 3.6c). Turbo-codul duo-binar prezintă o serie de avantaje care vor fi discutate în Cap. 5.

3.2.4 Modulația 16QAM și canal AWGN / fading plat

Modulațiile de tip QAM sunt utilizate în conjuncție cu turbo-codarea atunci când canalul de transmisie este puțin zgomotos (SNR-ul este relativ mare). În cazul sistemului cu modulație 16QAM, considerând secvențele a_k, b_k, c_k, e_k , bipolare (± 1) și $E_o=1$ (E_o reprezintă jumătate din energia semnalului emisibil cu cea mai mică energie din constelația 16QAM, [BH2]), în ipoteza cunoașterii ratei de turbo-codare și a raportului semnal per zgomot $\xi_0(dB) = 10 \cdot \lg(E_{bi}/N_0)$ rezultă, pentru puterea zgomotului (secvențele w_{kx} și w_{ky}), valoarea:

$$\overline{w_x^2} = \frac{N_0}{4 \cdot E_{bc}} \cdot \overline{a^2} \cdot \overline{x^2} = \frac{5 \cdot N_0}{4 \cdot R_c \cdot E_{bi}} = \frac{5}{4 \cdot R_c \cdot 10^{\xi_0/10}}. \quad (3.15)$$

În Fig. 3.6 este prezentat modelul canalului de transmisie cu modulație 16QAM, cu aceleași observații asupra coeficienților α_k , $\alpha_k \equiv 1$ pentru canal AWGN sau α_k este o variabilă aleatoare Rayleigh, Nakagami sau Rice cu putere unitară ($\overline{\sigma^2} = 1$) pentru canal cu fading plat. Relațiile de calcul aferente modelului din Fig. 3.6 și în ipotezele date anterior sunt după cum urmează. La emisie, după multiplexare, blocul de calcul C, execută operațiile:

$$\begin{aligned} x_k &= a_k \cdot (c_k + 2) \cdot \sqrt{E_o} = a_k \cdot (c_k + 2) \\ y_k &= b_k \cdot (e_k + 2) \cdot \sqrt{E_o} = b_k \cdot (e_k + 2). \end{aligned} \quad (3.16)$$

Canalul s-a simulat prin înmulțirea coeficienților x_k și y_k cu coeficienții α_{kx} și α_{ky} și prin însumarea coeficienților w_{kx} și w_{ky} . La recepție se calculează estimății biților emiși:

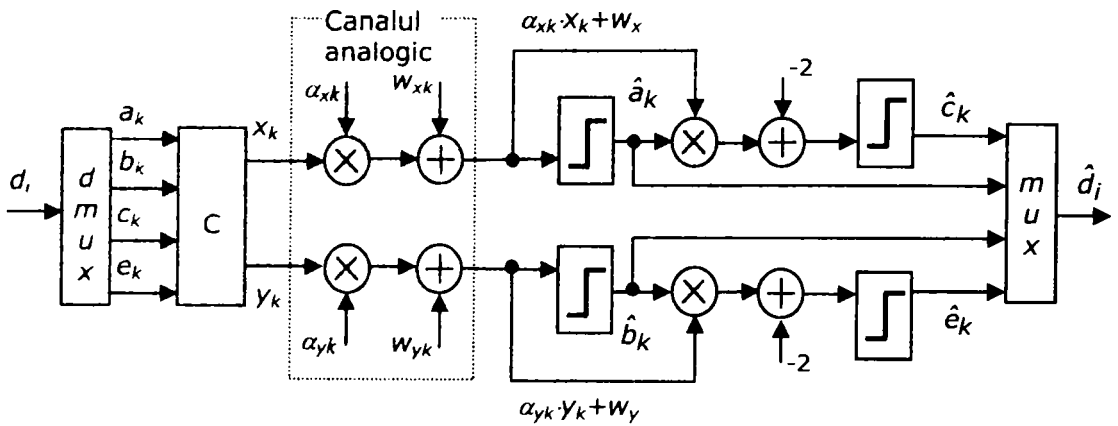
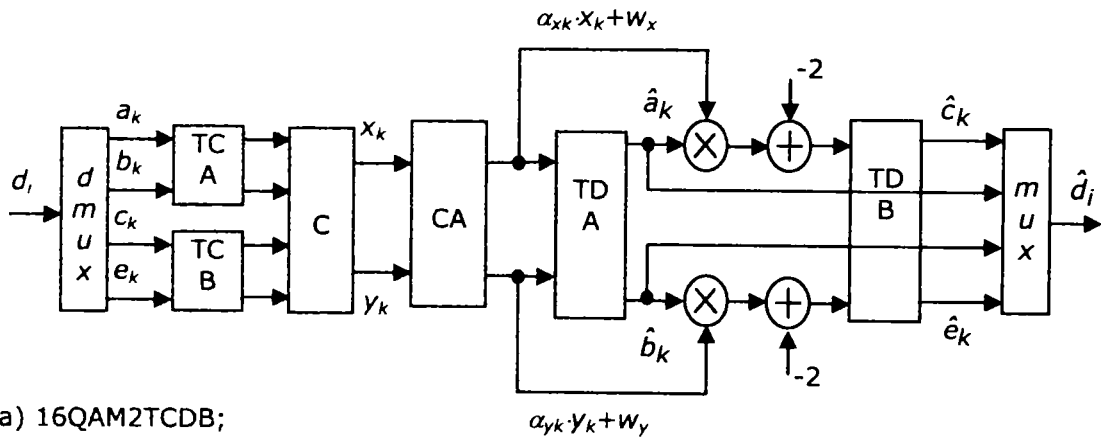
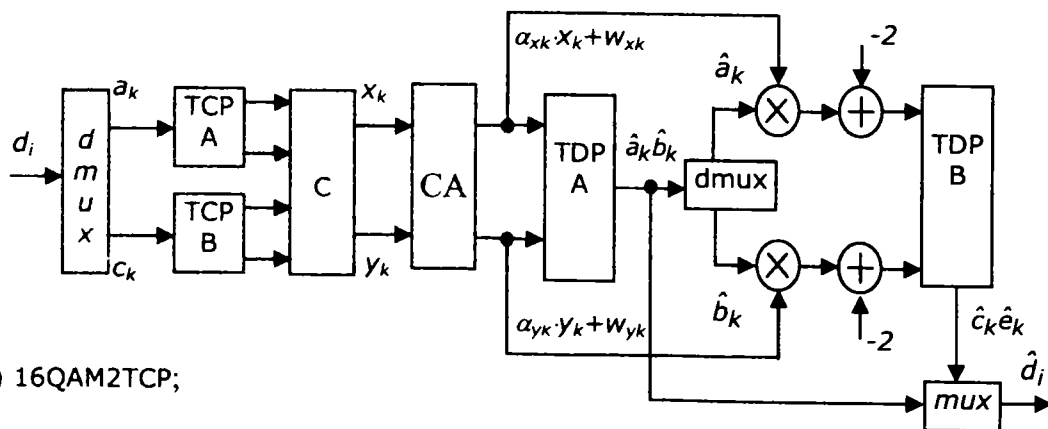


Figura 3.6 Modelul canalului digital cu fading plat și modulație 16QAM



a) 16QAM2TCDB;



b) 16QAM2TCP;

Figura 3.7 Modele de simulare pentru sisteme de transmisie turbo-codate cu modulație 16QAM

$$\begin{aligned}\hat{a}_k &= \text{sign}(a_{kx} \cdot x_k + w_{xk}) & \hat{b}_k &= \text{sign}(a_{ky} \cdot y_k + w_{ky}) \\ \hat{c}_k &= \text{sign}((a_{kx} \cdot x_k + w_{xk}) \cdot \hat{a}_k - 2) & \hat{e}_k &= \text{sign}((a_{ky} \cdot y_k + w_{ky}) \cdot \hat{b}_k - 2)\end{aligned}\quad (3.17)$$

La fel ca și în cazul modulației DQPSK defazat cu $\pi/4$ și în cazul modulației 16QAM, turbo-codarea se poate face în mai multe variante. Spre exemplu este posibil a aplica câte un turbo-cod separat pe fiecare ieșire a demultiplexorului. Însă volumul de calcul va fi enorm la decodare.

Nu este posibilă varianta cu un singur turbo-cod cu patru intrări deoarece la decodare cele 4 ieșiri se obțin recursiv în doi pași. Astfel, o variantă de compromis este aplicarea a două turbo-coduri, fiecare cu câte două intrări și două ieșiri, Fig. 3.7a). Utilizarea turbo-codurilor clasice (cu o intrare) este posibilă prin modificarea adecvată a demultiplexării, respectiv a multiplexării. Se obține astfel schema din Fig. 3.7b).

3.3 Performanțele sistemelor de transmisie cu modulație BPSK, DQPSK deplasat cu $\pi/4$ și 16QAM, necodate

În diagramele din Fig. 3.8 sunt prezentate curbele BER obținute prin simularea canalelor AWGN și cu fading plat de tip Rayleigh, [BH12], în conjuncție cu modulația BPSK, DQPSK deplasat cu $\pi/4$ și 16QAM, iar în diagrama din Fig. 3.9 sunt

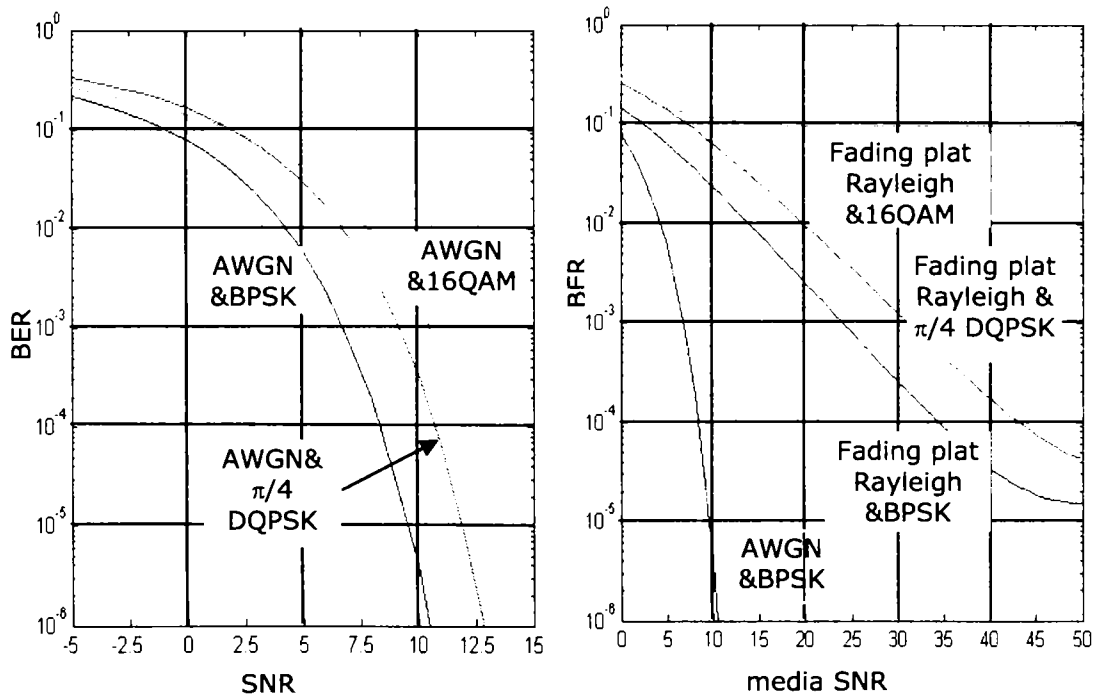


Figura 3.8 Performanțele BER obținute prin simularea sistemelor de transmisie cu modulație BPSK și $\pi/4$ DQPSK, în canale AWGN și cu fading plat de tip Rayleigh

prezentate curbele BER obținute prin simularea canalelor cu fading plat de tip Nakagami pentru patru valori ale parametrului m , în conjuncție cu modulația BPSK.

Evident, cele mai bune rezultate sunt date de canalul fără fading (AWGN). Modulația BPSK oferă un câștig de 2,5dB față de DQPSK deplasat cu $\pi/4$ și de 4dB față de 16QAM, pentru o plajă destul de largă de valori ale BER-ului. Indiferent de tipul modulației, dependența BER(SNR) este exponențial descrescătoare.

Canalele cu fading de tip Rayleigh sunt mult mai slabe în performanța BER. Se observă o dependență liniară descrescătoare a ratei erorii funcție de raportul semnal per zgomot, BER(SNR), pentru modulațiile BPSK și DQPSK deplasat cu $\pi/4$, cu o limitare a BER-ului la valori peste 10^{-5} . Modulația 16QAM este impracticabilă în canalele cu fading plat de tip Rayleigh.

Canalele cu fading plat de tip Nakagami, cu valori supraunitare ale parametrului m , au performanțe cuprinse între cele cu fading plat de tip Rayleigh (ce corespund, de fapt, unui canal cu fading plat de tip Nakagami cu $m=1$) și canalele fără fading (ce corespund unui canal cu fading plat de tip Nakagami cu $m=\infty$).

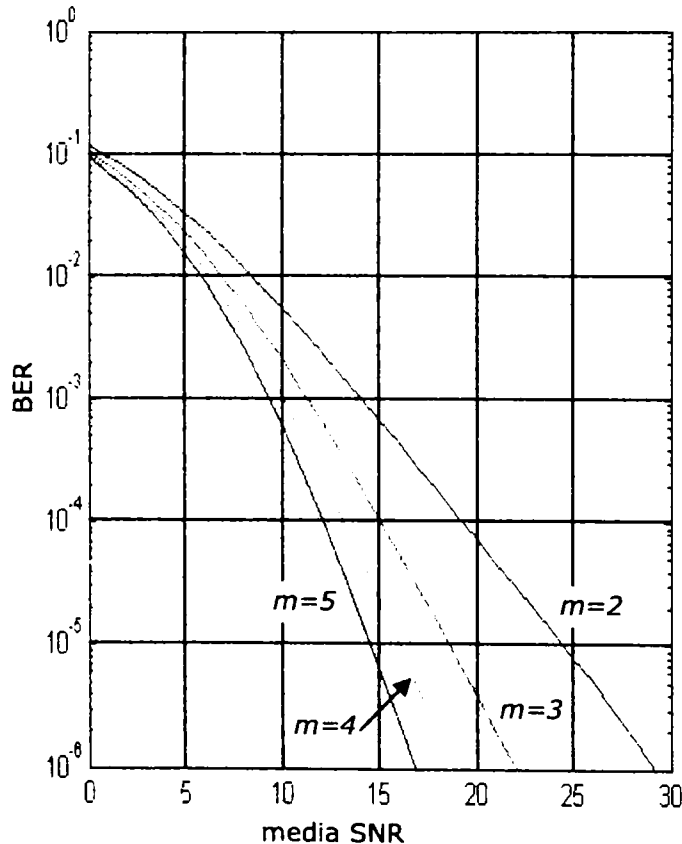


Figura 3.9 Performanțele BER obținute prin simularea unui sistem de transmisie cu modulație BPSK în canale cu fading plat de tip Nakagami cu $m=2, 3, 4$ și 5 , [PRO00]

3.4 Generarea numerelor aleatoare cu distribuție Nakagami

În acest paragraf se prezintă o metodă originală, [BH19] [BH21] [BH29], de generare a numerelor aleatoare cu distribuție Nakagami. Această metodă a fost aplicată pentru simularea canalului cu fading plat de tip Nakagami și implicit pentru determinarea performanțelor TC-ilor în canale de acest tip.

3.4.1 Distribuția Nakagami

Densitatea de probabilitate Nakagami este, [PRO]:

$$p_N(a) = \frac{2 \cdot m^m}{\Gamma(m)} \cdot \frac{a^{2m-1}}{(2\sigma^2)^m} \cdot \exp\left(-m \frac{a^2}{2\sigma^2}\right) \quad (3.18)$$

unde $m \geq 1/2$ este un parametru ce caracterizează anvergura fadingului canalului de transmisie. Valoarea medie \bar{a} și respectiv valoarea medie pătratică $\overline{a^2}$, pentru $m \geq 1$, întreg, sunt:

$$\bar{a} = \sqrt{2\sigma^2} \cdot \sqrt{\frac{n}{4m}} \cdot K, \quad \overline{a^2} = 2\sigma^2 \quad (3.19)$$

unde:

$$K=1 \text{ pentru } m=1 \text{ și } K = \prod_{i=1}^{m-1} \frac{2i+1}{2i} \text{ pentru } m \geq 2.$$

Fără a pierde din generalitate, vom presupune că $\overline{a^2} = 1$. Rezultă că: $2\sigma^2 = 1$ și $\bar{a} = \sqrt{n/4m} \cdot K$. În (3.18), $\Gamma(m)$ este funcția gamma dată prin relația:

$$\Gamma(m) = \int_0^{\infty} t^{m-1} \cdot e^{-t} \cdot dt. \quad (3.20)$$

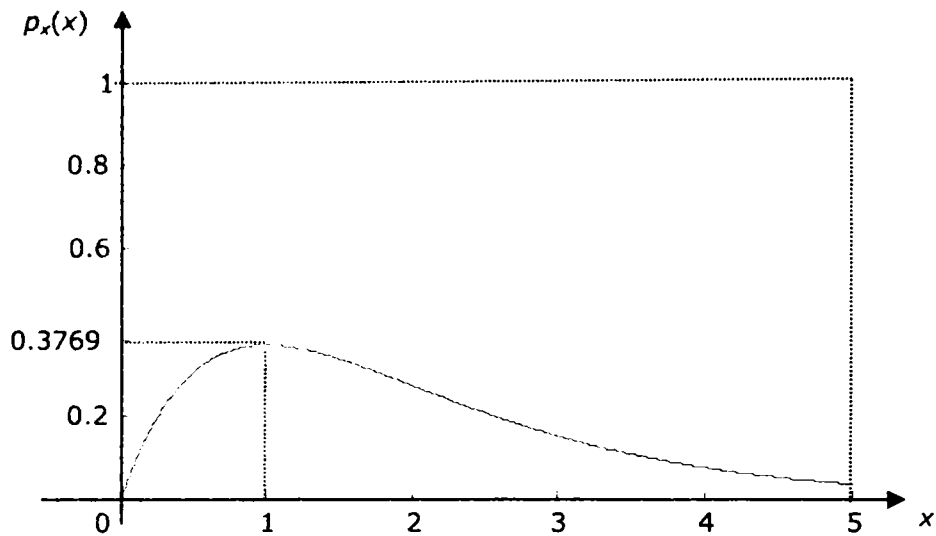
Pentru generarea numerelor aleatoare cu distribuție Nakagami, se face schimbarea de variabilă:

$$x = m \frac{a^2}{2\sigma^2} = ma^2. \quad (3.21)$$

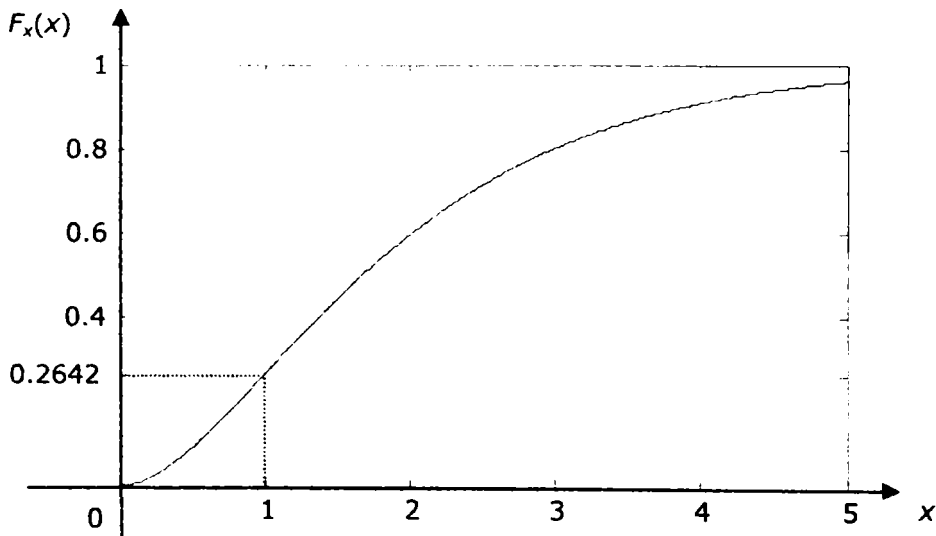
Înlocuind (3.21) în (3.18), densitatea de probabilitate Nakagami devine:

$$p_x(x) = \frac{p_N(a)}{\frac{dx}{da}} = \frac{x^{m-1} \cdot e^{-x}}{\Gamma(m)}. \quad (3.22)$$

Funcția de repartiție a probabilității variabilei aleatoare x se obține integrând relația (3.22):



a) densitatea de probabilitate;



b) funcția de repartiție;

Figura 3.10 Densitatea de probabilitate și funcția de repartiție a probabilității pentru variabila aleatoare x , pentru $m=2$

$$F_x(x) = \int_0^x p_x(t) \cdot dt = \frac{1}{\Gamma(m)} \cdot \int_0^x t^{m-1} \cdot e^{-t} \cdot dt = \frac{\Gamma_x(m)}{\Gamma(m)} \quad (3.23)$$

Dacă m este un număr natural atunci integrala:

$$I_{m-1}(x) = \int_0^x \frac{t^{m-1}}{(m-1)!} \cdot e^{-t} \cdot dt \quad (3.24)$$

se poate calcula prin părți:

$$\begin{aligned} I_{m-1}(x) &= \frac{t^{m-1}}{(m-1)!} \cdot e^{-t} \Big|_x^0 + \int_0^x \frac{t^{m-2}}{(m-2)!} \cdot e^{-t} \cdot dt \\ &= -\frac{x^{m-1}}{(m-1)!} \cdot e^{-x} + I_{m-2}(x). \end{aligned} \quad (3.25)$$

Utilizând recurența obținută găsim:

$$F_x(x) = I_{m-1}(x) = -e^{-x} \cdot \sum_{k=1}^{m-1} \frac{x^k}{k!} + I_0(x), \quad (3.26)$$

cu:

$$I_0(x) = \int_0^x e^{-t} \cdot dt = 1 - e^{-x}. \quad (3.27)$$

Așadar:

$$F_x(x) = 1 - e^{-x} \cdot \sum_{k=0}^{m-1} \frac{x^k}{k!}, \quad x \geq 0, \quad m = \text{întreg}. \quad (3.28)$$

Observații:

- 1) Dacă $m=1$ atunci $F_x(x) = 1 - e^{-a \cdot x}$ și se obține distribuția Rayleigh.
- 2) Relația (3.28) poate fi pusă sub forma:

$$F_x(x) = 1 - Q(x) \cdot e^{-x},$$

unde $Q(x)$ este un polinom de gradul $m-1$ în x .

În continuare vom determina punctul de inflexiune al funcției $F_x(x)$, acest punct fiind esențial pentru algoritmul de generare ce va fi descris în secțiunea următoare. $F_x(x)$ prezintă un punct de inflexiune x_f obținut prin anularea derivatei de ordin doi [BEC]:

$$\begin{aligned} \frac{d^2 F_x(x)}{dx^2} &= \frac{dp_x(x)}{dx} = \\ &= \frac{1}{\Gamma(m)} \cdot \left[(m-1) \cdot x^{m-2} - x^{m-1} \right] \cdot e^{-x}, \quad x \geq 0, \quad m \neq 1, \end{aligned} \quad (3.29)$$

de unde:

$$x_f = m-1. \quad (3.30)$$

Graficele funcțiilor $p_x(x)$ și $F_x(x)$, sunt prezentate în Fig.3.10, pentru $m=2$.

3.4.2 Algoritm pentru generarea numerelor aleatoare cu distribuție Nakagami

În cele ce urmează se prezintă un algoritm de generare a numerelor aleatoare cu distribuție Nakagami, bazat pe schimbarea de variabilă dată în Ec. (3.21) și pe următoarea teoremă, a cărei demonstrație poate fi găsită în [BEC05]:

Teorema 3.1: Prin transformarea unei variabile u , cu distribuție uniformă pe intervalul $[0, 1)$, prin inversa unei funcții de repartiție a probabilității, $F_x^{-1}(u)$, se obține o variabilă aleatoare având ca și funcție de repartiție pe $F_x(x)$.

Deoarece, pentru $m \neq 1$, funcția $F_x(x)$ dată de (3.22) sau de (3.25) nu este analitic inversabilă, o relație de forma $x = F_x^{-1}(u)$ nu se poate găsi decât prin aproximări. Un algoritm, original, de obținere a unei aproximări a relației $x = F_x^{-1}(u)$, cu $u \in [0, 1)$, este descris în continuare:

Pasul 1. Se generează $u \in [0, 1)$, o realizare a unei variabile aleatoare cu distribuție uniformă;

Pasul 2. Se calculează x_F , $p_x(x_F)$ și $F_x(x_F)$, cunoscând m ;

Pasul 3. Se calculează x_1 cu relația:

$$(x_1 - x_F) \cdot p_x(x_F) = u - F_x(x_F), \quad (3.31)$$

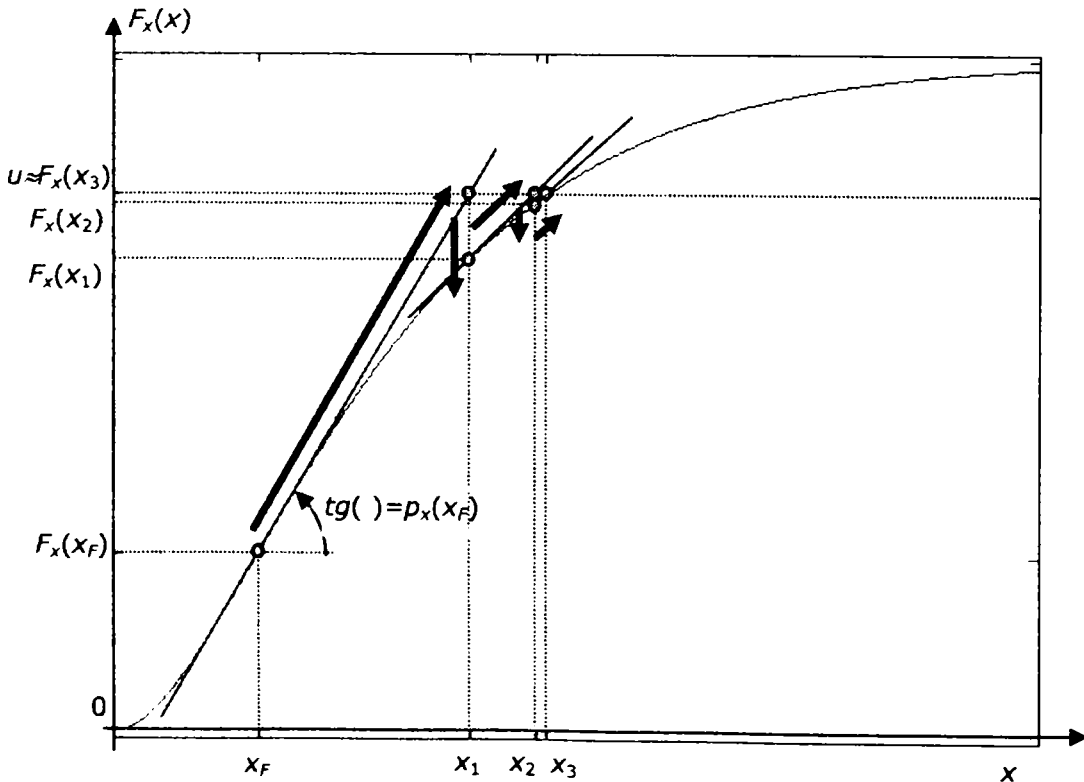


Figura 3.11 Ilustrarea algoritmului de generare a unei variabile aleatoare cu distribuție Nakagami

obținându-se:

$$x_1 = x_F + \frac{u - F_x(x_F)}{p_x(x_F)}; \quad (3.32)$$

Pasul 4. Se calculează $F_x(x_1)$ și se compară cu u . Dacă:

$$|F_x(x_1) - u| < c, \quad (3.33)$$

atunci se trece la pasul 5, unde c este precizia dorită în aproximare. Dacă relația (3.33) nu este satisfăcută, atunci se repetă pașii 3 și 4 înlocuind x_1 cu x_2 și x_F cu x_1 .

Pasul 5. Se revine la variabila α (Nakagami):

$$a = \sqrt{\frac{x}{m} \cdot 2\sigma^2} = \sqrt{\frac{x}{m}}. \quad (3.34)$$

În Fig. 3.11 este ilustrat procesul de generare descris mai sus. S-au considerat valorile $m=2$, $u=0.8$, $c=10^{-5}$, rezultând succesiv $x_F=1.0$, $x_1=2.456344$, $x_2=2.913904$, $x_3=2.992212$, $x_4=2.994307$, $F_x(x_3)=0.8$.

Evident, seria $\{x_k\}$ converge, printr-un număr finit de iterații, către $F_x^{-1}(u)$, crescător sau descrescător funcție de semnul diferenței $u - F_x(x_F)$.

Observație: Algoritmul prezentat anterior, în cazul în care m nu este natural, impune calculul a două integrale, a căror valori sunt $\Gamma_x(m)$ și $\Gamma(m)$. Acestea se pot face de asemenea printr-o aproximare numerică (integrale Riemann).

Vom prezenta în continuare o verificare a acurateții algoritmului de generare a numerelor aleatoare cu distribuție Nakagami, prezentat anterior. În acest scop s-au generat secvențe de N numere cu „putere” unitară ($\sigma^2 = 1$), pentru $m = 2, 3, 4$ și 5. Pe baza acestor secvențe s-au construit histograme ce sunt prezentate în anexa D. Fiecare histogramă s-a construit pentru 100 de intervale cu lățimi egale și s-a comparat cu densitatea de probabilitate teoretică corespunzătoare. Compararea s-a făcut atât grafic (anexa D) cât și prin relația:

$$\varepsilon = \frac{\sum_{k=1}^{100} (h_N(k) - p_N(k))^2}{\sum_{i=1}^{100} p_N^2(k)}, \quad (3.35)$$

unde $h_N(k) = n(k)/N$ reprezintă proporția $n(k)$ a numerelor cu valori în intervalul k , din totalul de N numere generate.

În figura 3.12, diagramele 1 și 3 prezintă valorile erorii ε în procente, definite prin relația (3.31), funcție de exponentul preciziei c , pentru diferite valori ale parametrului m . Aceste valori au fost obținute pentru secvențe de $N=10^5$

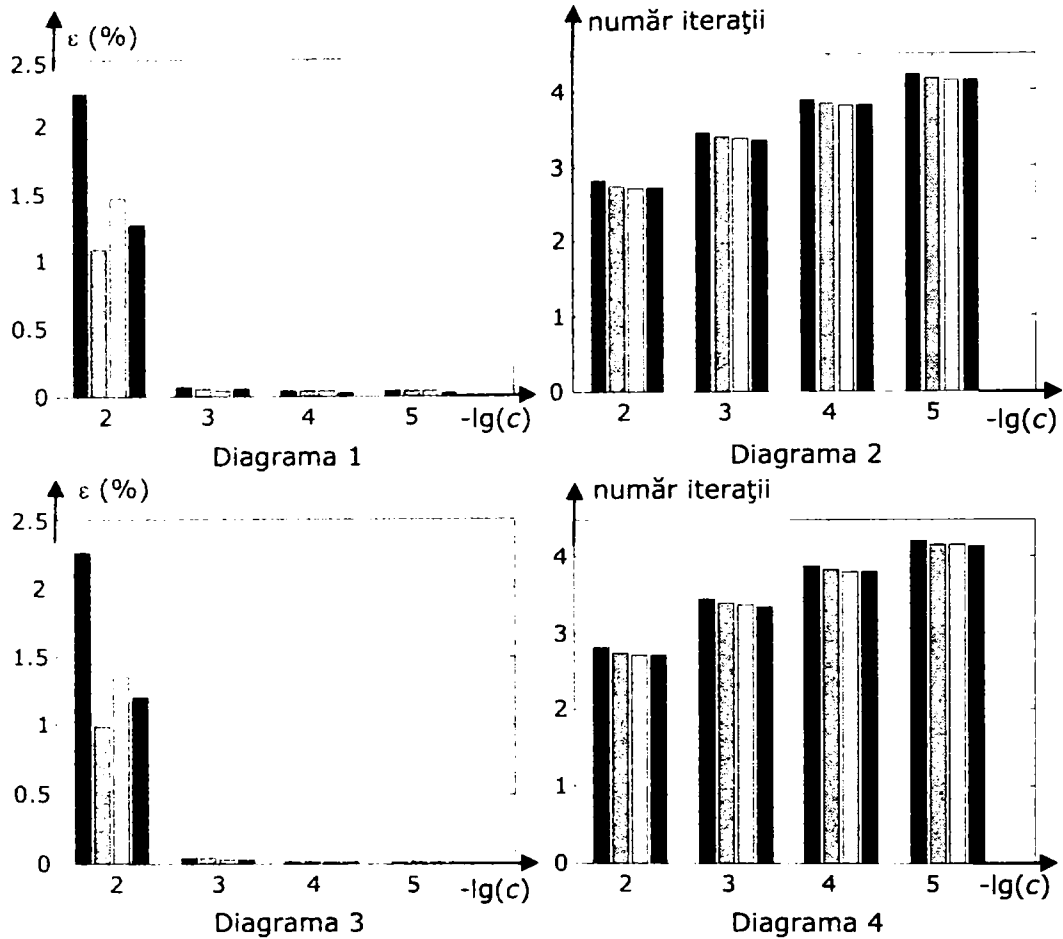


Figura 3.12 Variația erorii ε în procente și a numărului de iterații funcție de exponentul preciziei c , pentru secvențe de 10^5 numere (diagramele 1 și 2) respectiv de 10^6 numere (diagramele 3 și 4) și pentru 4 valori ale parametrului m : 2 – albastru, 3 – cian, 4 – galben și 5 – roșu.

numere (diagrama 1), respectiv $N=10^6$ numere (diagrama 3). În diagramele 2 și 4 se prezintă numărul mediu de iterații, necesar algoritmului descris anterior, pentru a genera un număr, funcție de precizia c aleasă.

Atât diagramele 1 și 3 din figura 3.12 cât și cele prezentate în Anexa D arată că o precizie a algoritmului fixată sub valoarea 10^{-3} este suficientă pentru a obține o eroare medie pătratică sub valoarea de 0,1%. Dacă precizia c ia valori de 10^{-5} , eroarea ε , indiferent de parametrul m , coboară mult sub pragul de 0,01%. Prețul obținerii unei asemenea performanțe este mărirea numărului de iterații cu o unitate, așa cum arată diagramele 2 și 4 din figura 3.12. În concluzie, algoritmul propus pentru generarea numerelor aleatoare cu distribuție Nakagami este foarte performant în ceea ce privește precizia, însă necesită în medie circa 4 pași algoritmici per număr generat. În plus este relativ simplu și poate fi utilizat pentru orice valoare a lui m .

4. Turbo-Coduri Uni-binare

4.1 Structura unui turbo-cod

Acest capitol descrie componența unui TC și parametrii care influențează performanța sa. Acești parametri sunt sugerați de Figura 4.1. Este descrisă influența fiecăruia dintre acești parametri, diferitele variante sau posibilități de alegere a lor, precum și modul în care aceste soluții alese influențează performanțele BER și FER. Analiza fiecărui parametru a fost finalizată printr-o evaluare a avantajelor și a dezavantajelor oferite de fiecare variantă, analiză ce include atât performanțele BER și FER cât și complexitatea realizării practice.

Toate variantele de turbo-cod analizate au fost implementate prin programe de simulare proprii. Printre variantele de turbo-cod descrise există și soluții originale. De asemenea sunt propuse și metode originale de analiză (ex: spectrul distanțelor de întrețesere).

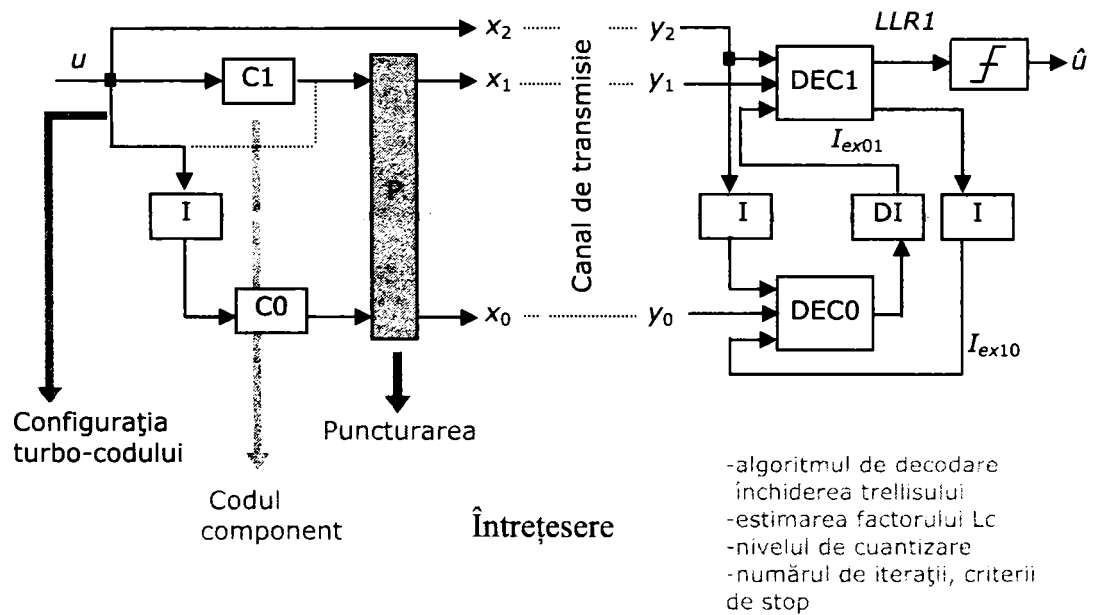


Figura 4.1 Schema generică corespunzătoare unui turbo-cod și parametrii ce intervin în definirea sa

4.2 Codurile convoluționale componente

Codurile ce sunt concatenate în formarea unui TC, pe baza cărora se construiesc codoarele C1 respectiv C0 din Fig.4.1, pot fi coduri convoluționale sau coduri bloc (BCH în speță), [GOP] [HOP]. În cele ce urmează se vor avea în vedere doar codurile convoluționale ca și coduri componente.

4.2.1 Codorul recursiv, sistematic –polinoamele atașate

În Figura 4.2 este prezentată schema generală a unui codor convoluțional recursiv și sistematic, de rată 1/2, schemă cunoscută în literatura de specialitate sub numele de „controller” [FOR70]. Cu notațiile din figură rezultă ecuațiile:

$$c[n] = \sum_{k=0}^M h_{1,k} w[n-k] \quad (4.1)$$

$$w[n] = \sum_{k=1}^M h_{0,k} \cdot w[n-k] + u[n] \quad (4.2)$$

Considerând că $h_{0,0}=1$ și aplicând transformata „D”, [JOZ] [WAD] [VUY], celor două ecuații (4.1) și (4.2) obținem:

$$C(D) = \left(\sum_{k=0}^M h_{1,k} \cdot D^k \right) \cdot W(D) \quad (4.3)$$

$$W(D) \cdot \sum_{k=0}^M h_{0,k} D^k = U(D) \quad (4.4)$$

Eliminând pe $W(D)$ rezultă:

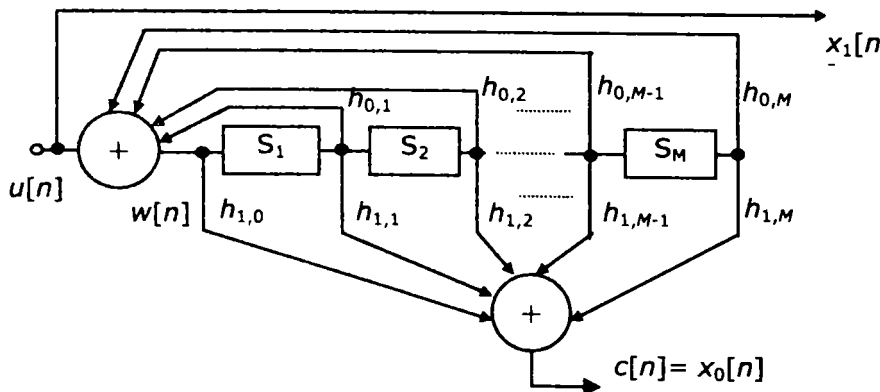


Figura 4.2. Codor convoluțional recursiv și sistematic cu sumatoare externe (forma canonică "controller")

$$\frac{C(D)}{U(D)} = \frac{\sum_{k=0}^M h_{1,k} \cdot D^k}{\sum_{k=0}^M h_{0,k} \cdot D^k} \quad (4.5)$$

ecuație care definește funcția de transfer a codorului. Matricea generatoare a codorului este:

$$H(D) = \begin{bmatrix} 1, & h_1(D) \\ & h_0(D) \end{bmatrix}. \quad (4.6)$$

Polinoamele $h_1(D)$ - de ieșire și $h_0(D)$ - de reacție trebuie să fie prime între ele. Lungimea de constrângere este dată de gradul maxim al polinoamelor $h_1(D)$ și $h_0(D)$ plus 1. În Tabelul 4.1 sunt indicate polinoamele cu coeficienți binari, până la gradul 5 ce pot fi utilizate pentru $h_1(D)$ sau $h_0(D)$. Polinoamele ce au cel puțin un divizor comun nu sunt compatibile. În prezenta lucrare au fost analizate turbo-codurile construite cu toate combinațiile posibile de coduri prezentate în Tabelul 4.1.

O alternativă a circuitului din Figura 4.2 este codorul cu sumatoare interioare, prezentat în Figura 4.3, schemă cunoscută în literatura de specialitate sub numele de „observer”, [JOW]. Cu notațiile din figură rezultă:

$$\begin{aligned} s_M[n] &= g_{1,M} \cdot u[n-1] + g_{0,M} \cdot c[n-1] \\ s_{M-1}[n] &= g_{1,M-1} \cdot u[n-1] + g_{0,M-1} \cdot c[n-1] + s_M[n-1] \\ &\dots \\ s_1[n] &= g_{1,1} \cdot u[n-1] + g_{0,1} \cdot c[n-1] + s_2[n-1] \\ c[n] &= g_{1,0} \cdot u[n] + s_1[n] \end{aligned} \quad (4.7)$$

Aplicând transformata „D” ecuațiilor (4.7) rezultă:

$$\begin{aligned} S_M(D) &= g_{1,M} \cdot U(D) \cdot D + g_{0,M} \cdot C(D) \cdot D \\ S_{M-1}(D) &= g_{1,M-1} \cdot U(D) \cdot D + g_{0,M-1} \cdot C(D) \cdot D + S_M(D) \cdot D \\ &\dots \\ S_1(D) &= g_{1,1} \cdot U(D) \cdot D + g_{0,1} \cdot C(D) \cdot D + S_2(D) \cdot D \\ C(D) &= g_{1,0} \cdot U(D) + S_1(D). \end{aligned} \quad (4.8)$$

Tabelul 4.1 Polinoamele utilizabile în TC-uri

Grad	Ireductibil	Primitiv	Polinoamele în octal și divizorii lor										
0	*		1	1	0	0	0	0	0	0	0	0	0
1	*	*	3	1	0	0	0	0	0	0	0	0	0
2			5	1	3	0	0	0	0	0	0	0	0
	*	*	7	1	0	0	0	0	0	0	0	0	0
3			11	1	3	7	0	0	0	0	0	0	0
	*	*	13	1	0	0	0	0	0	0	0	0	0
	*	*	15	1	0	0	0	0	0	0	0	0	0
			17	1	3	0	0	0	0	0	0	0	0
4			21	1	3	0	0	0	0	0	0	0	0
			23	1	0	0	0	0	0	0	0	0	0
			25	1	0	7	0	0	0	0	0	0	0
			27	1	3	0	0	15	0	0	0	0	0
	*	*	31	1	0	0	0	0	0	0	0	0	0
			33	1	3	7	0	0	0	0	0	0	0
			35	1	3	0	13	0	0	0	0	0	0
	*	*	37	1	0	0	0	0	0	0	0	0	0
5			41	1	3	0	0	0	0	0	0	37	0
			43	1	0	7	0	15	0	0	0	0	0
	*	*	45	1	0	0	0	0	0	0	0	0	0
			47	1	3	0	13	0	0	0	0	0	0
	*	*	51	1	0	0	0	0	0	0	0	0	0
			53	1	3	0	0	0	0	0	31	0	0
			55	1	3	7	0	0	0	0	0	0	0
	*	*	57	1	0	0	0	0	0	0	0	0	0
			61	1	0	7	13	0	0	0	0	0	0
			63	1	3	0	0	0	0	0	0	0	0
			65	1	3	0	0	0	23	0	0	0	0
	*	*	67	1	0	0	0	0	0	0	0	0	0
			71	1	3	0	0	15	0	0	0	0	0
*	*	73	1	0	0	0	0	0	0	0	0	0	
*	*	75	1	0	0	0	0	0	0	0	0	0	
		77	1	3	7	0	0	0	0	0	0	0	

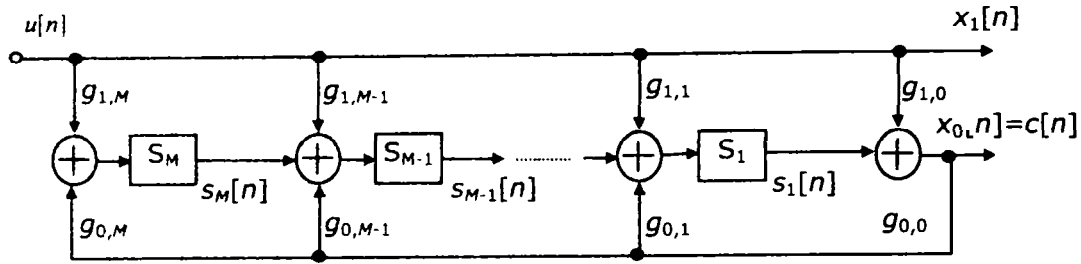


Figura 4.3 Codor convoluțional recursiv și sistematic cu sumatoare interne (forma canonică „observer”)

Eliminând variabilele $S_m(D)$ rezultă:

$$C(D) = \sum_{m=0}^M g_{1,m} \cdot D^m \cdot U(D) + \sum_{m=1}^M g_{0,m} \cdot D^m \cdot C(D) \quad (4.9)$$

sau:

$$C(D) \cdot \left(\sum_{m=0}^M g_{0,m} \cdot D^m \right) = U(D) \cdot \left(\sum_{m=0}^M g_{1,m} \cdot D^m \right), \quad (4.10)$$

ecuație care este identică cu (4.5), fapt ce atestă echivalența funcțională a circuitelor din Figurile 4.2 și 4.3.

4.2.2 Performanțele codurilor convoluționale. Concluzii

În Tabelul 4.2 sunt indicați parametrii utilizați în simulările a căror rezultate sunt prezentate în acest paragraf.

Turbo-codul utilizat are asociată schema dată în Figura 4.1 cu o concatenare paralelă. Nu s-a utilizat puncturare încât rata de turbo-codare rezultată a fost aproximativ 1/3. Diferența se datorează procedurii de închidere a trellisului pentru primul cod, C1 prin inserarea a M biți redundanți în secvența de date. În acest fel rata de turbo-codare este $(N-M)/(3 \cdot N)$. Dispozitivul de întrețesere (interleaver-ul) a fost de tip S. Simulările au fost făcute, pentru toate perechile de coduri, la lungimea de întrețesere de 1784 aleasă conform cu recomandările „Consultative Committee for Space Data Systems” prezentate în [CCS]. Decodarea componente DEC1 și DEC2 implementează algoritmul MAP cu 15 iterații și un criteriu de stop a iterațiilor bazat pe valorile LLR-ului. Valoarea pragului a fost aleasă în toate cazurile egală cu 10.

În Tabelul A.1 și Tabelul A.2 din Anexa A sunt prezentate rezultatele simulărilor. Numerele din prima coloană indică, în octal, valorile polinoamului $g_0(D)$, respectiv numerele din prima linie indică, în octal, polinomul $g_1(D)$. Perechile de polinoame incompatibile (ce au divizori comuni) sunt marcate printr-un zero. Rezultatele furnizate de ambele tabele (pentru perechile de polinoame compatibile) reprezintă rata erorii de bit (BER) înmulțită cu 10^8 . Toate simulările au fost făcute la un raport semnal per zgomot $SNR=1\text{dB}$.

Tabelul 4.2 Parametrii BTC-lui utilizați în simulările pentru analiza codurilor convoluționale componente

	Parametrul BTC-lui	Varianta aleasă pentru simulări
1	Codul component	analizat în acest paragraf
2	Configurația TC-ului	paralelă
3	Puncturarea Rata de codare	fără $(N-M)/(3 \cdot N) \approx 1/3$
4	Întreșeserea Lungimea blocurilor	S-ilv. ($S=29$) sau CCSDS 1784 biți, conform cu recomandarea CCSDS
5	Canal	AWGN, $SNR=1\text{dB}$
6	Modulație	BPSK
7	Algoritm de decodare	MAP
8	Închiderea trellisului	C1-închis, C0-neînchis coeficienții $\beta(\)$ la start au fost setați echiprobabili
9	Cuantizarea	infinită
10	Număr de iterații Criteriu de oprire	15 $\text{prag LLR} = 10$

Este de remarcat faptul că toate polinoamele primitive din Tabelul 4.2 au rezultate bune, atunci când sunt utilizate ca polinoame de reacție. Dintre acestea polinoamele 13 și 15 oferă cele mai bune rezultate în asociație cu multe dintre coduri, cum sunt 17, 21, 25, 33, 37, 43, 47, 45, 51, 57, 61, 63, 67, 71 și 75. Concluzia este valabilă și pentru polinoamele primitive de grad 4, 23 și 31. Situația nu este la fel de evidentă pentru gradul 5, acest fapt putându-se explica prin comparația făcută doar la 1dB. Este posibil și probabil ca peste această valoare, codurile având o memorie M mai mare să fie mai performante.

O explicație a acestui fapt cât și, totodată, o concluzie finală este că performanțele turbo-codului depind foarte mult de compatibilitatea *cod component* – *interleaver* și nu separat de fiecare în parte. Această concluzie este foarte importantă pentru proiectarea TC-ilor, pentru că, conform acesteia, trebuie căutate perechi cod-interleaver, misiune mult mai dificilă decât aceea de a căuta separat codul și separat interleaverul. O astfel de strategie v-a fi prezentată în capitolul următor. Deși concluziile afirmate anterior se regăsesc în multe referințe bibliografice ([HLZ], [WAD], [VUY]), totuși nu am găsit în literatura de specialitate rezultate ale unor investigații asemănătoare.

4.3 Configurația turbo-codului

Prin termenul de configurație a TC-ului, în acest context, se înțelege modul de concatenare a codoarelor și a decodoarelor componente. Atât codoarele cât și decodoarele componente pot fi concatenate paralel sau serial. Dacă atât codoarele cât și decodoarele sunt concatenate paralel atunci vorbim de un cod (convoluțional)

concatenat paralel (Parallel Concatenated convolutional Code) sau turbo-cod¹. Dacă concatenarea este serială atât la emisie cât și la recepție atunci vorbim despre un cod (convoluțional) concatenat serial (Serial Concatenated convolutional Code). Există posibilitatea decodării paralele a unei secvențe turbo-codate printr-o structură serială la emisie. Se obține astfel, per ansamblu emisie-recepție, o structură hibridă (Hibrid Concatenated Convolutional Code). Rezultă așadar trei moduri distincte de concatenare, respectiv trei configurații posibile: paralelă, serială și hibridă. Cu toate că noțiunea de turbo-cod invocă concatenarea paralelă, prin turbo-cod (TC) se va înțelege oricare din cele trei configurații, urmând a se specifica explicit care este configurația utilizată. În paragrafele următoare vor fi descrise cele trei configurații definite anterior și vor fi prezentate performanțele obținute în fiecare caz.

4.3.1 Cod convoluțional concatenat paralel (PCCC)

Turbo-codul sau codul convoluțional concatenat paralel, are structura prezentată în Figura 4.4. Descrierea funcționării sale poate fi găsită în [BGT93], [HLY00] sau [BH3]. Secvența de date este codată de ambele codoare, C1 și C0, cu diferența că pentru cel de-al doilea codor C0 secvența de date se reordonează. Operația de reordonare sau de întrețesere (interleaving) este efectuată de către dispozitivul de întrețesere „I”, numit în literatura engleză interleaver.

Deoarece prin codările efectuate de codoarele C1 și C0 rezultă două secvențe de biți redundanți, x_0 și x_1 , de aceeași lungime cu a secvenței originale x_2 , rezultă că TC-ul paralel are o rată de codare naturală (excluzând procedeele de puncturare) de 1/3. Această valoare este posibil să se micșoreze cu câteva procente dacă se utilizează închiderea trellisului.

În cazul TC-ului paralel, ambele decodoare calculează LLR (Log Likelihood Ratio – logaritmul raportului de plauzibilitate) și generează secvență extrinsecă I_{ex01} , respectiv I_{ex02} referitoare la secvența de biți de informație. Nu se mai calculează LLR pentru secvența de biți de paritate (redundanța), așa cum trebuie să facă decodoarele componente în cazul TC-urilor seriale sau hibride.

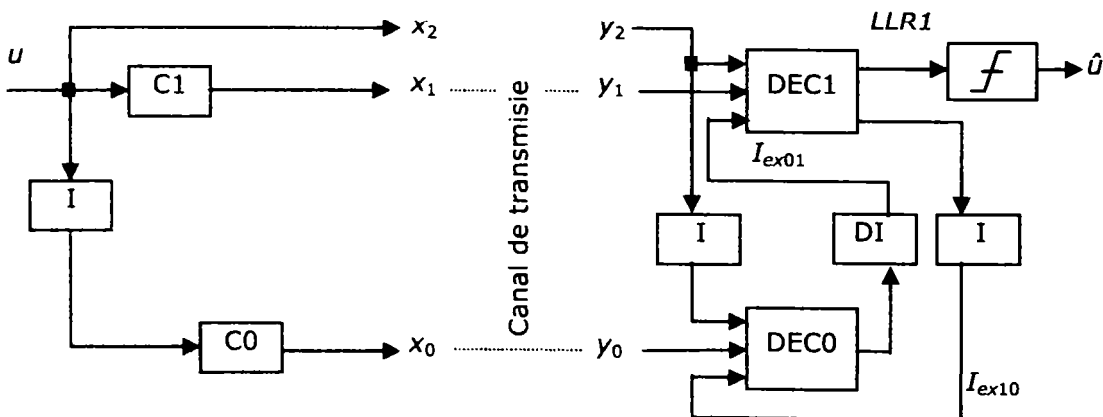


Figura 4.4 Schema corespunzătoare unui cod convoluțional concatenat paralel (PCCC)

¹ În fapt, noțiunea de turbo-cod se identifică cu cea de cod (convoluțional) concatenat paralel, și doar într-o accepțiune formală ea include și modurile de concatenare serial și hibrid (vezi paragraful 2.3).

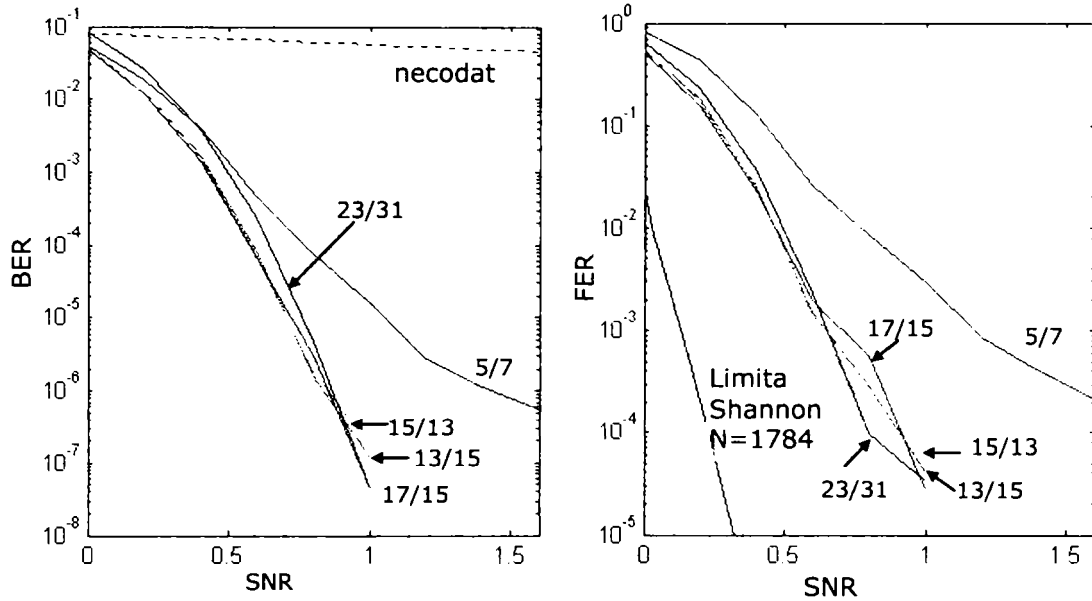


Figura 4.5 Performanțele unor TC concatenate paralel și interleaver de tip S

În Figura 4.5 sunt prezentate performanțele BER și FER obținute cu un TC paralel și cinci coduri convoluționale reprezentative, dintre cele mai performante indicate în paragraful anterior. Ceilalți parametri ai TC-urilor utilizați în simulări sunt definiți în Tabelul 4.2, cu precizarea că interleaverul utilizat a fost de tip S.

4.3.2 Cod convoluțional concatenat serial (SCCC)

În cazul codului convoluțional concatenat serial (Figura 4.6), secvența de date, având lungimea de N biți, este codată de codorul C1. Secvența de date este apoi multiplexată cu secvența rezultată prin codarea efectuată de C1, obținându-se o secvență de lungime $2 \cdot N$ care, după o prealabilă întrețesere, este codată de către

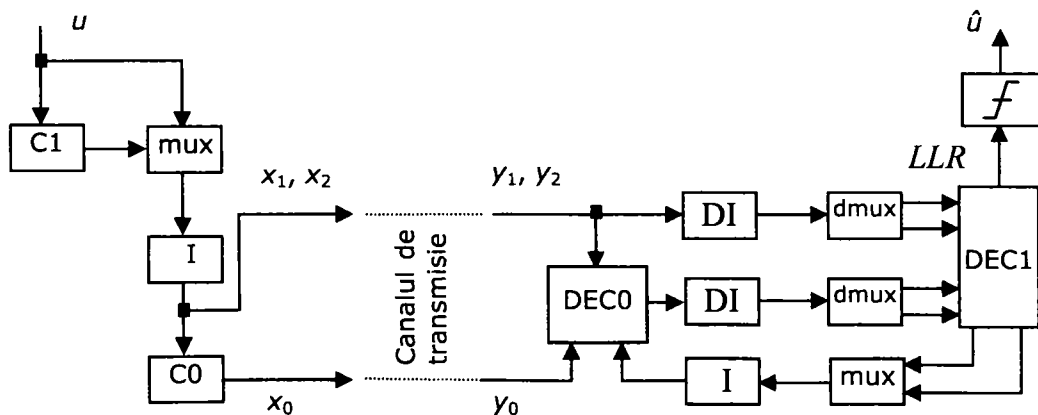


Figura 4.6 Schema corespunzătoare unui cod convoluțional concatenat serial (SCCC)

C0. Cele două secvențe (de la intrarea lui C0 și de la ieșirea sa) constituie secvența transmisă. Rezultă așadar pentru SCCC o rată de codare normală intrinsecă de valoare $1/4$, [DIP].

Tabelul 4.3 Parametrii SCCC-lui utilizați în simulări

	Parametrul BTC-lui	Varianta aleasă pentru simulări
1	Codul component	analizat în acest paragraf
2	Configurația TC-ului	serială
3	Puncturarea Rata de codare	fără $(N-M)/(4N) \approx 1/4$
4	Întreșeserea Lungimea blocurilor	S-ilv. ($S=29$) $892 = (1784 / 2)$ conform cu recomandarea (CCSDS)
5	Canal	AWGN, $SNR=1dB$
6	Modulație	BPSK
7	Algoritm de decodare	MAP
8	Închiderea trellisului	C1-închis, C0-neînchis coeficienții $\beta()$ la start au fost setați echiprobabili
9	Cuantizarea	infinită
10	Număr de iterații Criteriu de oprire	15 $prag LLR = 10$

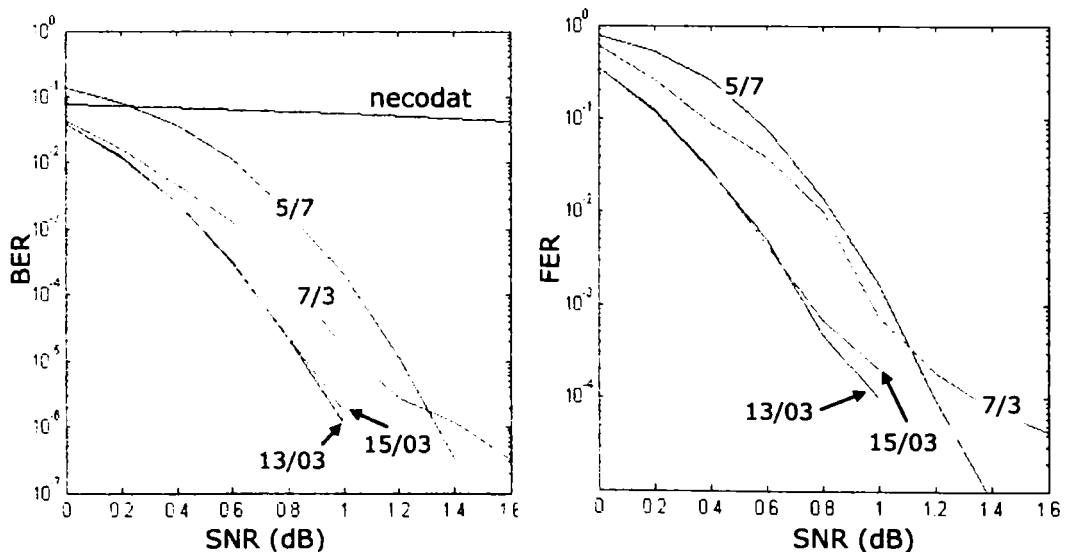


Figura 4.7 Performanțele SCCC-urilor

Tabelul 4.4 Performanțele codurilor convoluționale în concatenare serială
 Numerele din căsuțe indică: BER $\times 10^9$; FER $\times 10^6$; număr de erori
 Căsuțele marcate cu „x” corespund unor perechi de polinoame incompatibile

$g_0(D)$ $g_1(D)$	1	3	5	7	11	13	15	17
1	x	18913131 1000000 168516	20285393 1000000 180540	3615168 783200 32175	18549493 999900 164905	18549493 589700 14533	1637682 592900 14559	638132 214000 5673
3	7556004 225800 67324	x	x	8876 500 79	x	0 0 0	0 0 0	x
5	7619775 227900 67816	x	x	17191 100 153	x	19685 300 175	11698 200 104	x
7	128939 2200 1144	17303 400 154	90337 800 804	x	x	70078 700 623	115298 1000 1025	3937 200 35
11	7779190 237400 69157	x	x	x	x	96625 700 859	51518 400 458	x
13	272440 2400 2422	70753 800 629	138020 1100 1227	152080 1400 1352	2209448 14700 19642	x	3430933 22900 30501	3148143 20200 27987
15	280427 2600 2493	51181 800 455	211136 1900 1877	157930 1200 1404	2163217 15000 19231	3400899 22000 30234	x	4797187 30700 42647
17	880089 149000 7824	x	x	701912 130000 6240	x	2117997 16400 18829	2056242 16600 18280	x

La recepție, decodorul component DEC0 va opera în mod corespunzător cu secvențe de lungime $2 \cdot N$. Informația apriori pe care o utilizează DEC0 se referă exclusiv la secvența pe care acesta o tratează ca secvență de informație (x_1 , x_2 sau corespunzător y_1 , y_2). DEC0 generează o secvență de informație extrinsecă, cu referire la secvența compusă x_1 , x_2 . Însă după demultiplexare această secvență extrinsecă (de lungime $2 \cdot N$) se va regăsi la intrarea în DEC1 sub forma a două secvențe apriori ce îi furnizează informație (apriori) lui DEC1 atât pentru secvența sa de biți de informație (x_1) cât și pentru secvența sa de biți de control (x_2). DEC1 va genera informație extrinsecă atât despre biții săi de informație cât și pentru cei de control. Aceste două secvențe, după multiplexare și întreșere constituie secvența apriori a lui DEC0.

Tab. 4.4 prezintă performanțele obținute prin simularea unui SCCC. Parametrii utilizați au fost sintetizați în Tab. 4.3. În simulări, pentru fiecare cod, s-au transmis un număr de blocuri egal cu 10000. Este important de subliniat aici că, deși lungimea de întreșere a rămas aceeași, 1784 biți, lungimea blocului de date este doar 892, jumătate din cea utilizată în cazul paralel. De remarcat că ierarhia codurilor s-a modificat esențial față de cazul paralel. Astfel, cele mai bune performanțe, în cazul concatenării seriale, se obțin prin codurile 13/3, 15/3 și 17/7. De asemenea, pentru lungimea de constrângere $K=3$ cel mai performant cod, până la un $SNR=1,2$ dB, este 7/3. Însă, așa cum prezintă Fig. 4.7, pentru valori ale SNR -ului peste 1,2 dB codul 5/7 are o comportare asimptotică mai bună decât 7/3.

4.3.3 Cod convoluțional concatenat hibrid (HCCC)

Codul convoluțional concatenat hibrid (Fig.4.8) prezintă o structură serială la emisie (codare) și una paralelă la recepție (decodare). Secvența de date, x_2 , este codată de către C1, obținându-se o secvență care după o prealabilă întrețesere este codată de către C0. Secvențele emise sunt: cea de date, x_2 , secvența codată de C1 (după întrețesere, Figura 4.8a, sau dinainte de întrețesere, Figura 4.8 b) și secvența codată de C0. Cu toate că schema decodorului HCCC-ului este identică cu cea a PCCC-ului, funcțional există deosebiri esențiale. Astfel, în cazul HCCC-ului, DEC2 îi furnizează lui DEC1 informație extrinsecă despre secvența de biți de control (a lui DEC1) și nu despre secvența de date, ca în decodorul PCCC. Asemenea PCCC-ului și HCCC are o rată de codare $R=1/3$.

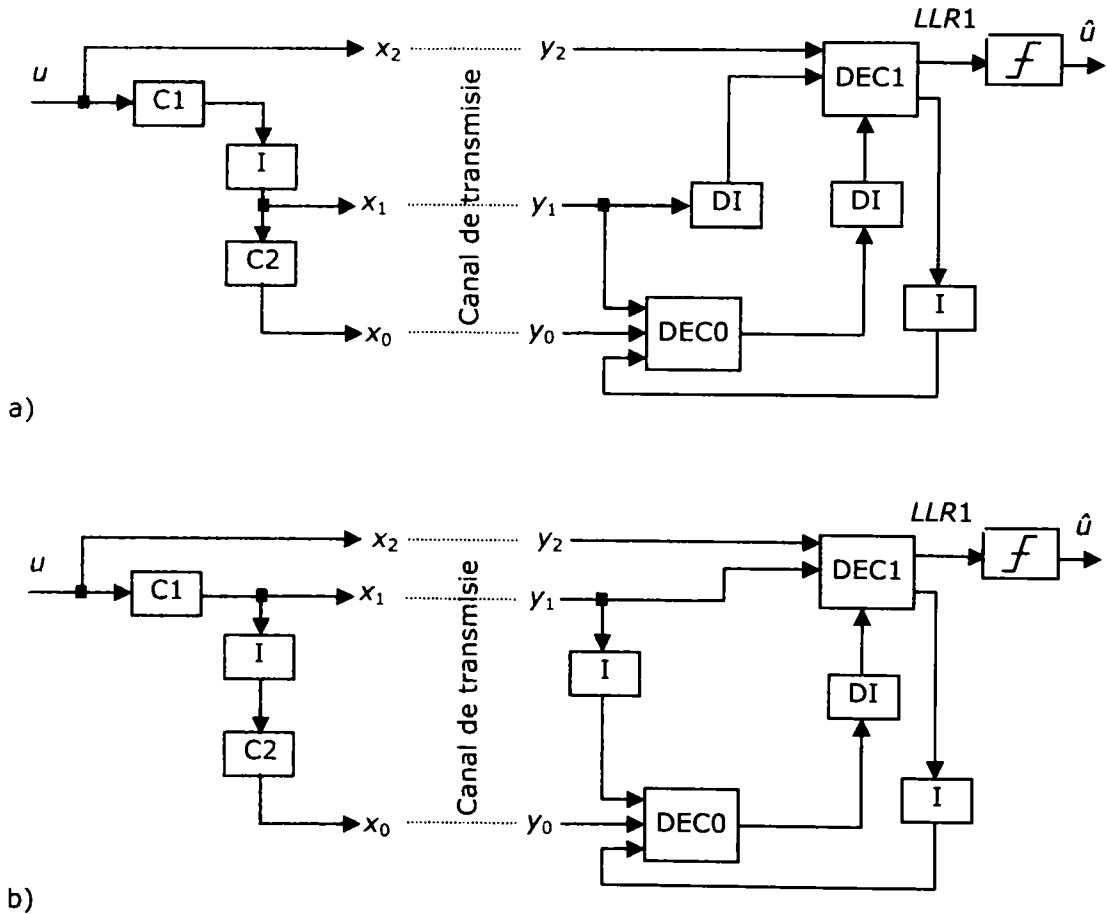


Figura 4.8 Două scheme de coduri convoluționale concatenate hibrid

Tabelul 4.5 Parametrii HCCC-lui utilizați în simulări

	Parametrul BTC-lui	Varianta aleasă pentru simulări
1	Codul component	analizat în acest paragraf
2	Configurația TC-ului	hibridă
3	Puncturarea Rata de codare	fără $(N-M)/(3N) \approx 1/3$
4	Întreșeserea Lungimea blocurilor	S-ilv. ($S=29$) 1784 biți, conform cu recomandarea CCSDS
5	Canal	AWGN, $SNR=1dB$
6	Modulație	BPSK
7	Algoritm de decodare	MAP
8	Închiderea trellisului	C1-închis, C2-neînchis coeficienții $\beta(\)$ la start au fost setați echiprobabili
9	Cuantizare	infinită
10	Număr de iterații Criteriu de oprire	15 $prag LLR = 10$

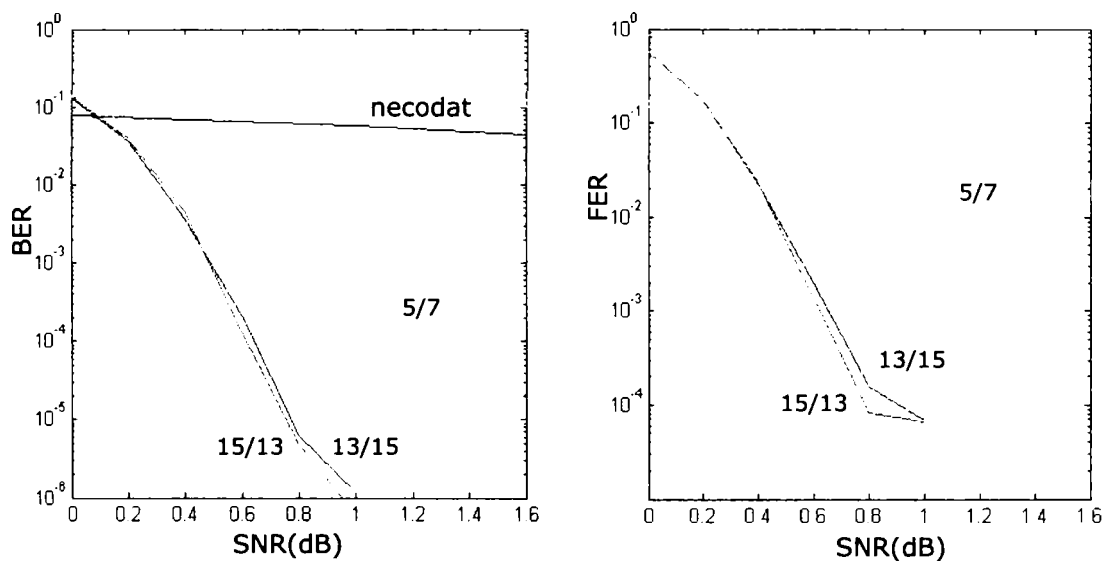


Figura 4.9 Performanțele HCCC-urilor

În Tabelul 4.5 sunt prezentați parametrii HCCC-ului a cărei performanțe sunt prezentate în Tabelul 4.6 și în Figura 4.9. În simulările pentru construcția Tabelul 4.6 s-au transmis 10000 de blocuri pentru fiecare cod. Rezultatele prezentate de Tabelul 4.6 afirmă o ierarhie a codurilor convoluționale asemănătoare cazului paralel. Astfel, pentru $K=3$ cel mai performant cod este 5/7 iar pentru $K=4$ codurile 15/13 și 13/15.

Tabelul 4.6 Performanțele codurilor convoluționale în concatenare hibridă
 Numerele din fiecare căsuță indică, în ordine: $BER \times 10^9$; $FER \times 10^6$; număr de erori
 Căsuțele marcate cu „x” corespund unor perechi de polinoame incompatibile

$g_0(D)$ $g_1(D)$	1	3	5	7	11	13	15	17
1	x	68735465 1000000 612433	69003370 1000000 614130	41198089 999800 366663	68793363 1000000 611573	21902699 982000 194715	21590326 980900 191938	20153993 952800 179169
3	108952749 1000000 970769	x	x	29146516 783800 259400	x	1738695 69400 15457	1892125 71800 16821	x
5	108995280 1000000 970058	x	x	268426 14600 2389	x	210236 11100 1869	345556 27000 3072	x
7	68349213 1000000 608308	32455505 776800 288854	613932 37000 5464	x	x	47581 2500 423	16872 900 150	5035658 223100 44767
11	109288638 1000000 971576	x	x	x	x	25421 1900 226	43982 2400 391	x
13	40483352 988700 359897	2025984 69800 18011	257592 11400 2290	23172 1300 206	144994 11000 1289	x	0 0 0	7424 600 66
15	4035601 988800 358765	2987626 127700 26560	966254 89900 8590	23847 1500 212	140944 10500 1253	1574 100 14	x	7424 400 66
17	36046681 967100 320455	x	x	4201799 169900 37354	x	2699 200 24	53318 4800 474	x

4.3.4 Concluzii

În subcapitolul 4.3 au fost prezentate modalitățile de configurare (concatenare) a turbo-codurilor precum și performanțele BER(SNR) oferite de fiecare. În diagramele din Figura 4.10 sunt comparate cele mai bune rezultate obținute cu fiecare din cele trei configurații prezentate anterior. Trebuie însă subliniat faptul că cele trei configurații oferă rate de codare naturale diferite, 1/3 pentru configurațiile paralel și hibrid și 1/4 pentru configurația serială.

Figura 4.10 arată performanțe asemănătoare pentru cele trei tipuri de configurații. Cu toate acestea, comparația nu este într-un totu obiectivă ținând cont de rata de codare mai mică oferită de configurația serială și de lungimile blocurilor de date diferite (1784 pentru paralel și hibrid și 892 pentru serial). Rata de codare mai mică ar trebui să conducă, în mod firesc la performanțe superioare pentru configurația serială, însă lungimea blocului de date mai mică contrabalansează acest fapt. Totuși, trebuie remarcat aici că lungimea de întrețesere este și în cazul serial de 1784 biți, astfel că, din acest punct de vedere există egalitate. În fapt, datorită diferențelor structurale dintre configurațiile paralelă și hibridă, pe de o parte, și serială, pe de altă parte, este dificil de a găsi acei parametri care să conducă la

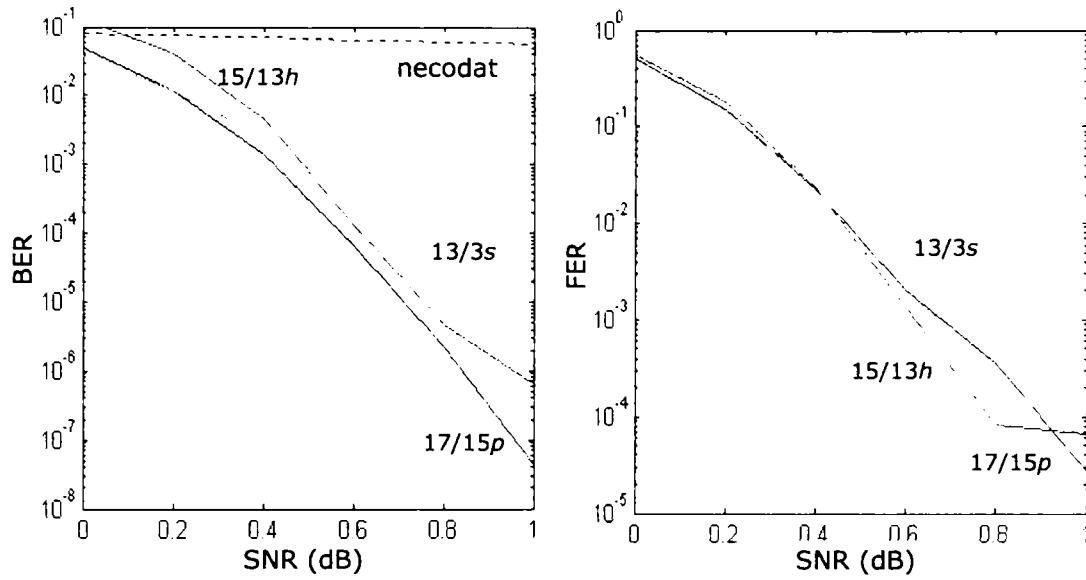


Figura 4.10 Comparație între performanțele celor mai bune TC-uri în configurație: paralelă (17/15), hibridă (15/13) și serială (13/3)

comparații obiective: indiferent de modul de alegere a acestor parametri, vor fi diferențe între ratele de codare, lungimile blocurilor de date sau lungimile de întretesere.

O comparație mai obiectivă între configurația serială și celelalte două ar trebui să implice puncturarea. Puncturarea ar putea permite ajustarea ratei de codare a configurației seriale la valoarea $1/3$. În plus va trebui să se modifice și lungimea de întretesere, pentru ca lungimea blocului de date să fie egală cu cea din configurațiile paralelă și hibridă.

Dacă comparația dintre configurația serială și celelalte două nu este foarte obiectivă-datorită ratelor de codare diferite, între configurațiile paralelă și hibridă aceasta este edificatoare. În mod evident configurația paralelă este superioară celei hibride.

Cu toate că Figura 4.10 sugerează superioritatea configurației paralele, trebuie subliniat încă un fapt, poate mai puțin evident. Anume interleaverul utilizat a fost în toate cazurile un S interleaver. Acest tip de interleaver s-a dovedit a fi cel mai performant pentru configurația paralelă. Rămâne de verificat dacă afirmația este valabilă și pentru configurațiile hibridă și serială. Această verificare nu este cuprinsă în cadrul acestei lucrări.

4.4 Întreșeserea

4.4.1 Introducere

Dispozitivul de întreșesere (interleaver) este o componentă indispensabilă a oricărui turbo-cod. 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, [AHK] [BAK]. Altfel spus, un dispozitiv de întreșesere implementează o funcție bijectivă de forma:

$$\pi : I \rightarrow I, \text{ cu } I = \{1, 2, \dots, N\}, \quad (4.11)$$

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

$$\pi^{-1} : I \rightarrow I, \text{ cu } \pi^{-1}(\pi(i)) = i, \forall i \in I. \quad (4.12)$$

Un bun interleaver, [BH10] [DAM99] [HOK97], trebuie să împrăștie cât mai bine secvența de biți originală. Încercările cercetătorilor din domeniu de a găsi permutarea ideală sunt departe de finalitate. Mai mult, practica dovedește că interleaverul trebuie ales și funcție de codul (convoluțional) component. Cu toate acestea, așa cum se va arăta în continuare în acest paragraf, este posibil să se facă o apreciere asupra calităților intrinseci ale interleaverului (fără a ține cont de codul component). Până în prezent, în literatura de specialitate, ca și parametru ce cuantifică „împrăștierea” produsă de interleaver s-a considerat distanța minimă de întreșesere. În acest paragraf am definit „spectrul distanțelor de întreșesere” (IDS – Interleaving Distances Spectrum) pe baza căruia am introdus un parametru, gradul de împrăștiere, ce completează analiza care se poate face asupra unui interleaver. În aceste condiții, printr-o analiză (ce va fi prezentată în continuare) asupra câtorva dintre cele mai cunoscute tipuri de interleavere am arătat că un bun interleaver trebuie:

- să aibă o distanță minimă de întreșesere de valoare cât mai mare și
- să aibă un grad de împrăștiere cât mai bun.

Fiind dată funcția interleaver, π , se definește, [CRO00], distanța de întreșesere dintre pozițiile i și j ca:

$$d(i, j) = |i - j| + |\pi(i) - \pi(j)|, \forall i, j \in I, i \neq j. \quad (4.13)$$

Atunci distanța minimă de întreșesere este dată prin:

$$d_{min} = \min_{\substack{i, j \in I \\ i \neq j}} d(i, j) \quad (4.14)$$

Mulțimea valorilor funcției $d(\cdot)$, dată prin (4.14), este mărginită inferior de valoarea 2 iar superior de $2 \cdot N - 2$.

În subparagraful următor sunt prezentate câteva dintre cele mai cunoscute tipuri de interleavere cunoscute în prezent. În subparagraful 4.4.3 am introdus

spectrul distanțelor de întrețesere (IDS). În subparagrafele 4.4.4 și 4.4.5 am calculat IDS-ul pentru întrețeserile de tip S și rectangular, iar în subparagraful 4.4.6 sunt prezentate câteva rezultate ale simulărilor făcute, urmând ca în subparagraful 4.4.7 să se prezinte concluziile asupra analizei întrețeserilor.

4.4.2 Proiectarea dispozitivelor de întrețesere

Dispozitivul de întrețesere aleator, sau pur aleator, este relativ simplu de realizat, [BAP1] [BAP2], oferă o bună împrăștiere a secvenței originale, însă are în general $d_{min} = 2$, adică cea mai mică valoare posibilă. În Figura 4.11 este ilustrată operația de întrețesere (permutarea) de tip aleator pentru un bloc de date de lungime $N=10$.

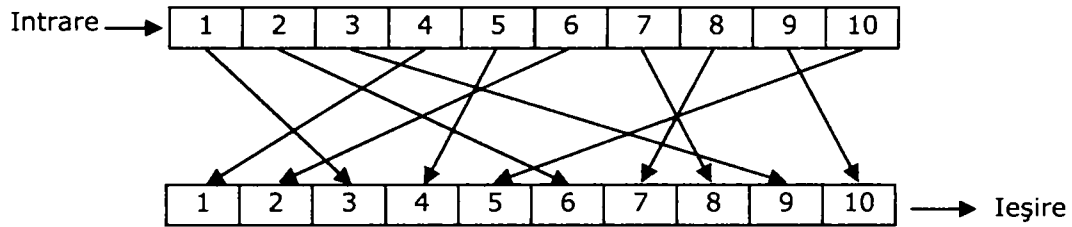


Figura 4.11 Dispozitiv de întrețesere aleator

Procedeul de construcție al unui dispozitiv de întrețesere aleator este următorul. Cunoscând lungimea de întrețesere, N , se construiește mulțimea $I = \{1, 2, \dots, N\}$. Se alege în mod aleator un număr, $n_1 \in I$. Se atribuie $\pi(1) = n_1$ și se elimină această valoare din I . Procedeul se repetă până la epuizarea mulțimii I .

Un dezavantaj major al întrețeserii aleatoare este nereproductibilitatea procedurii de generare al funcției π : o dată generată funcția de tip aleator ea trebuie memorată pentru a putea fi reprodusă.

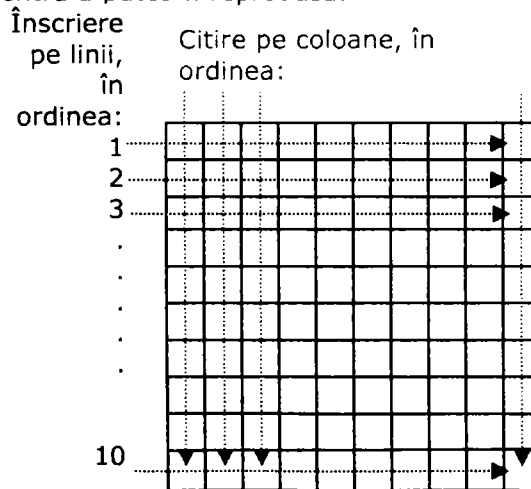


Figura 4.12 Întrețesere bloc

Dispozitivul de întrețesere rectangular sau bloc prezintă cea mai simplă structură. În Figura 4.12 este prezentat un dispozitiv de întrețesere bloc

(rectangular) având lungimea $N=10 \times 10$. Datele de intrare sunt introduse în dispozitiv linie cu linie. Citirea se va face pe coloane, schimbându-se astfel ordinea biților. După întrețesere secvența devine: $n_{11}, n_{111}, n_{21}, \dots, n_{91}, n_{2}, n_{22}, \dots, n_{90}, n_{100}$. Oricare doi biți aflați inițial la mai puțin de 10 (numărul de coloane) poziții unul de celălalt vor fi depărtați la cel puțin 10 (numărul de linii) poziții.

Dispozitivul de întrețesere pseudo-aleator face parte din recomandarea CCSDS (Consultative Committee for Space Data Systems) denumită "CCSDS Recommendation for Telemetry Channel Coding", [CCS].

Denumirea dispozitivului provine de la faptul că permutările nu sunt aleatoare în adevăratul sens al cuvântului (avem de-a face cu o dezordine controlată). Este un dispozitiv de întrețesere performant, care îmbină avantajele furnizate de interleaverale aleator și bloc, adică prezintă o bună împrăștiere la o distanță minimă suficient de mare.

Permutarea pentru fiecare secvență de lungime N a blocului este dată de o reordonare particulară a biților $1, 2, \dots, N$, generată de următorul algoritm¹:

- $N = k_1 \cdot k_2$ unde k_1 este un parametru fix, iar k_2 variază în funcție de lungimea dispozitivului de întrețesere;

- pentru s de la 1 la N (poziția curentă înainte de interschimbare) se efectuează următoarele operații și se obțin permutările $\pi(s)$:

- $m = (s - 1) \bmod 2$

- $i = \left\lfloor \frac{s - 1}{2k_2} \right\rfloor$

- $j = \left\lfloor \frac{s - 1}{2} \right\rfloor - ik_2$

- $t = (19i + 1) \bmod \frac{k_1}{2}$

- $q = t \bmod 8 + 1$

- $c = (p_q \cdot j + 21m) \bmod k_2$

- $\pi(s) = 2(t + c \frac{N+1}{2} + 1) - m$

în care $[]$ - parte întreagă

mod - operație modulo (clase de resturi)

$$q = 1 \div 8 \text{ și } p_1 = 31, p_2 = 37, p_3 = 43, p_4 = 47, p_5 = 53, p_6 = 59, \\ p_7 = 61, p_8 = 67.$$

Dispozitivul de întrețesere de tip "S" este de tip aleator, însă, spre deosebire de cel pur aleator, prin construcție se forțează o distanță minimă de întrețesere egală cu S , [BH27] [PHA]. Algoritm de construcție al funcției de întrețesere este următorul: se selectează o posibilă poziție viitoare pentru bitul curent. Această poziție este comparată cu a celor S biți selectați anterior în aceeași manieră aleatoare. Dacă se îndeplinește condiția ca:

¹ Notățiile utilizate în acest algoritm sunt preluate intact din referința [CCS02] și nu au fost menționate în lista de simboluri de la pagina viii

$$|n(n) - n(j)| > S \text{ pentru } n \text{ și } j \text{ cu proprietatea } |n - j| < S \quad (4.15)$$

adică poziția după interschimbare a bitului curent diferă cu S poziții față de a celor S alese anterior, atunci se trece mai departe. Dacă condiția nu va fi îndeplinită, se alege o altă poziție a bitului curent, care va fi la rândul său testată. Procesul se va repeta până când s-au găsit pozițiile tuturor celor N biți. Simulările pe computer au

demonstrat că, dacă $S \leq \sqrt{\frac{N}{2}}$, atunci procesul va converge. Proiectarea acestor

echipamente este dificilă pentru că, după ce o mare parte a algoritmului a fost parcursă, este din ce în ce mai greu din punct de vedere probabilistic să se genereze numere aleatoare din cele rămase în secvență care să îndeplinească cerința (4.15).

4.4.3 Spectrul distanțelor de întreșesere

În acest subparagraf este prezentat spectrul distanțelor de întreșesere, care reprezintă o contribuție originală. Spectrul distanțelor de întreșesere (IDS) constituie un aparat matematic pe baza căruia se poate face o apreciere mai obiectivă asupra proprietăților unui interleaver. De asemenea, IDS-ul oferă o imagine a interleaverului, în sensul că îi conferă o identitate fiecărui tip de interleaver. Prin IDS se poate recunoaște ușor metoda de întreșesere pe care o implementează o anumită permutare.

În [BH28] am introdus funcția „spectru” a distanțelor, $ids : J \rightarrow N$,

$$ids(k) = \text{număr perechi } (i, j) \in I \times I \text{ pentru care } d(i, j) = k \quad (4.16)$$

unde, pentru simplitate $J = \{1, 2, \dots, N\}$. Cu N s-a notat mulțimea numerelor naturale.

Se poate defini atunci gradul de împrăștiere, sd –spreading degree, ca și:

$$sd = \text{supp}\{ids(k)\} / \max_{k \in J} ids(k) \quad (4.17)$$

adică raportul dintre suportul funcției ids și valoarea maximă a spectrului distanțelor. Prin suportul funcției ids se înțelege numărul distanțelor $d \in J$ pentru care $ids(d) > 0$.

În Figura 4.13 sunt prezentate spectrele distanțelor de întreșesere pentru dispozitivele de întreșesere rectangular cu $N=1024$ (IR32), pseudo-aleator cu $k_1=10$ și $N=1000$ (IP6), pseudo-aleator cu $k_1=8$ și $N=1024$ (IP12), pseudo-aleator cu $k_1=8$ și $N=1000$ (IP10), aleator (IS1A) și S-interleaver-ele cu $S=5$ (IS5A), cu $S=10$ (IS10A), cu $S=20$ (IS20A), cu $S=23$ (IS23A). În Tabelul 4.7 sunt indicate, pentru fiecare dintre aceste interleavere, distanța minimă de întreșesere (d_{min}), distanța maximă de întreșesere (D_{max}), valoarea maximă a spectrului distanțelor (ids_{max}), suportul spectrului distanțelor ($\text{supp}(ids)$) cât și gradul de împrăștiere, sd .

Din cele prezentate se observă că IDS-urile dispozitivelor de întreșesere de tip aleator sunt foarte asemănătoare. Diferența dintre acestea o constituie „zgomotul” suprapus peste componenta continuă cât și valoarea de start (d_{min}).

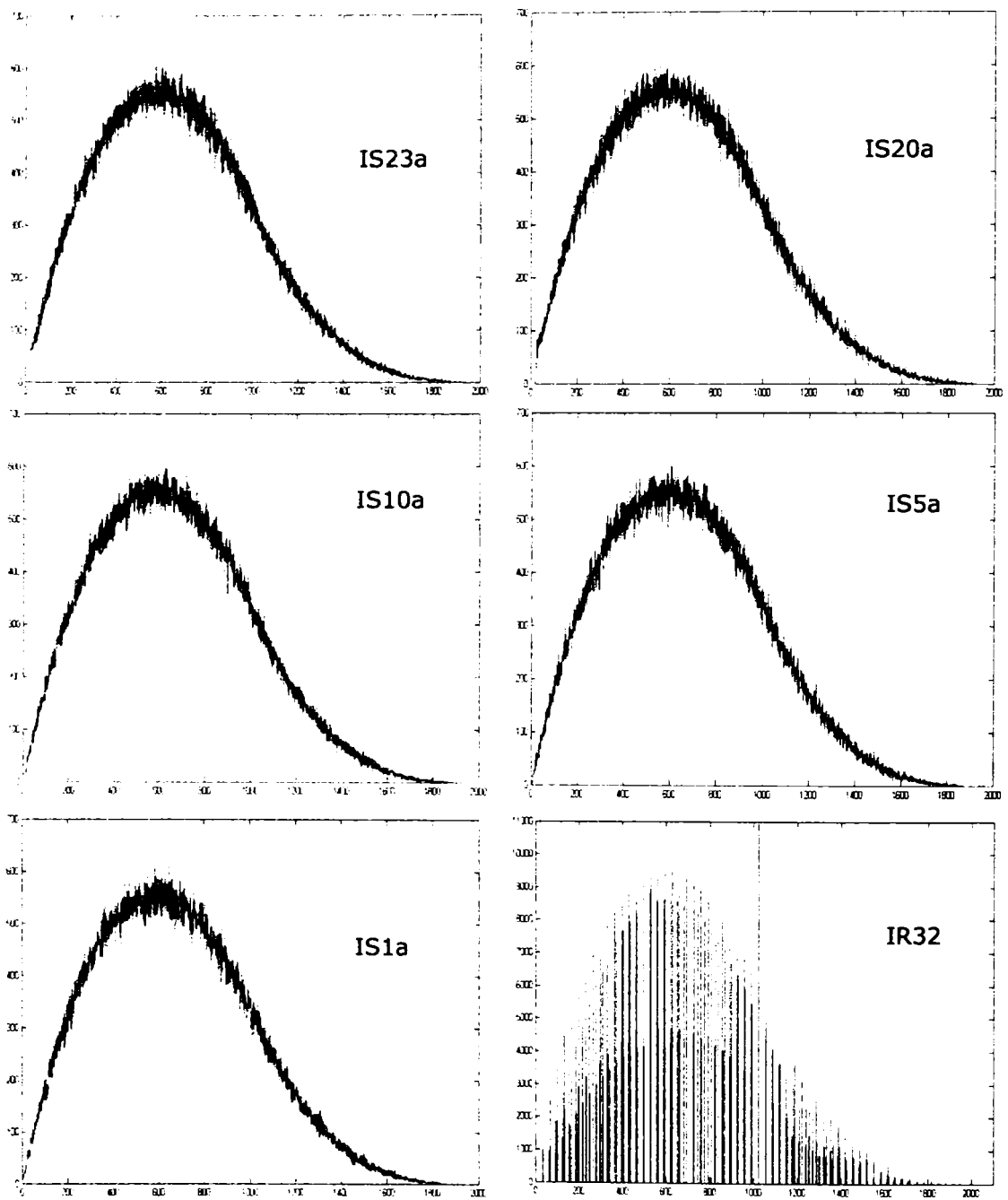


Figura 4.13 Spectrele distanțelor de întrețesere: număr de perechi de poziții i și j ce au distanța d , funcție de distanța d , pentru interleaveurile rectangular cu $N=1024$ (IR32), pseudo-aleator cu $k_1=10$ și $N=1000$ (IP6), pseudo-aleator cu $k_1=8$ și $N=1024$ (IP12), pseudo-aleator cu $k_1=8$ și $N=1000$ (IP10), aleator (IS1A) și S-interleaver-ele cu $S=5$ (IS5A), cu $S=10$ (IS10A), cu $S=20$ (IS20A), cu $S=23$ (IS23A)

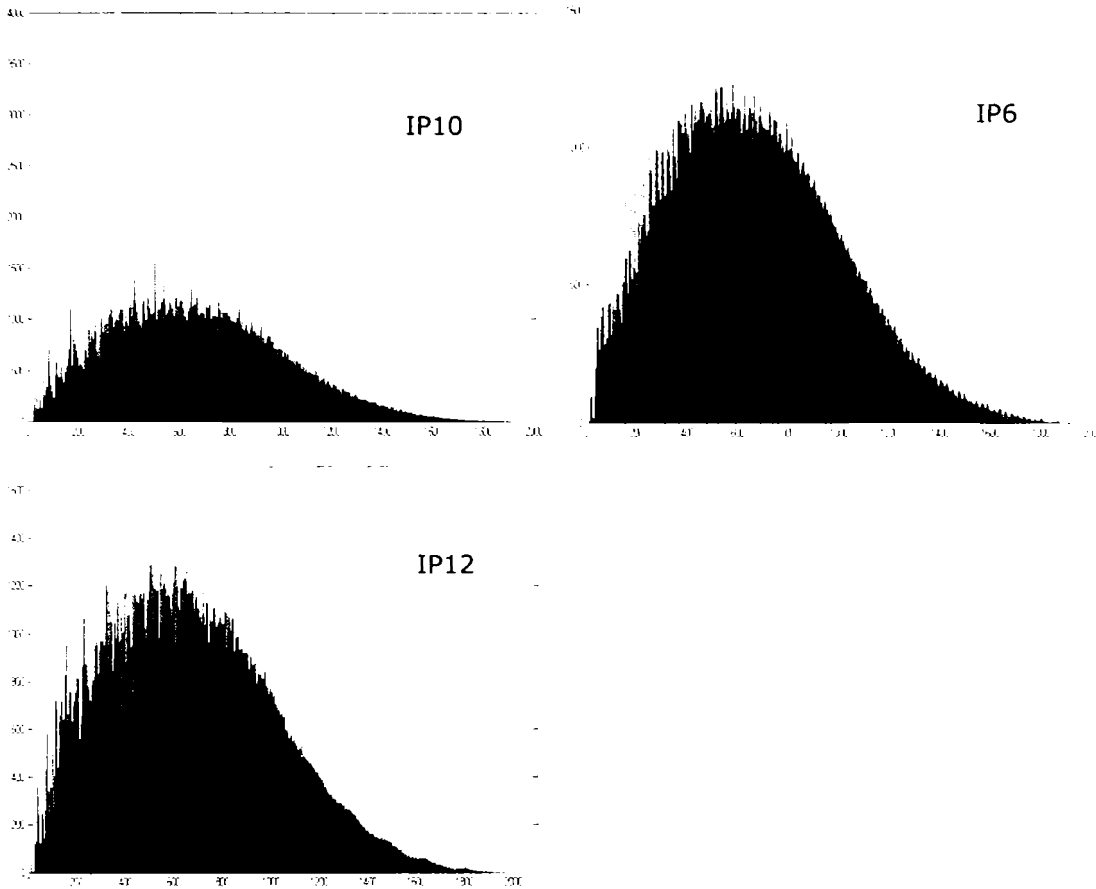


Figura 4.13 (continuare) Spectrele distanțelor de întrețesere: număr de perechi de poziții i și j ce au distanța d , funcție de distanța d , pentru interleavele rectangulare cu $N=1024$ (IR32), pseudo-aleator cu $k_1=10$ și $N=1000$ (IP6), pseudo-aleator cu $k_1=8$ și $N=1024$ (IP12), pseudo-aleator cu $k_1=8$ și $N=1000$ (IP10), aleator (IS1A) și S-interleaver-ele cu $S=5$ (IS5A), cu $S=10$ (IS10A), cu $S=20$ (IS20A), cu $S=23$ (IS23A)

Tabelul 4.7 Parametrii interleavelor investigate

	IS23A	IS20A	IS10A	IS5A	IS1A	IR32	IP10	IP6	IP12
d_{min}	24	21	11	6	2	33	10	6	12
D_{max}	1945	1935	1937	1972	1909	2046	1964	1978	2036
S_{max}	619	614	600	601	610	10944	3769	1424	1679
$supp$	1839	1855	1845	1846	1842	122	947	939	963
sd	2.9709	3.0212	3.0750	3.0715	3.0197	0.0111	0.2513	0.6594	0.5736
N	1000	1000	1000	1000	1000	1024	1000	1000	1024

Identitatea IDS -lor dispozitivelor de întrețesere de tip S o confirmă și datele din Tabelul 4.7. Practic, exceptând d_{min} , acestea diferă cu sub 2%.

Pentru a compara S-interleaveurile din punct de vedere al spectrului, acesta poate fi privit ca și un semnal determinist având zgomot suprapus:

$$ids(k) = ids_m(k) + w(k) \quad (4.18)$$

și, utilizând o metodă de filtrare (mediere), se poate face comparația între spectrele mediate $ids_m[k]$ ale dispozitivelor de întrețesere aleatoare.

Algoritmul de mediere este următorul¹:

1. inițializare

$d_f = 0$; diferența dintre adevărat și prezis

D_{min} ; punctul de start D_{max} ; punctul de final

$Lm = 30$; lungimea de mediere

$Ids(k)$, $k = D_{min} \div D_{max}$; spectrul nemediat

$$c_p(j) = \frac{2 \cdot j}{Lm \cdot (Lm + 1)}, j = 1 \div Lm; \text{coeficienții de ponderare}$$

2. pentru i de la $n+1$ la $N-1$:

$med = 2 \cdot d_f / (Lm + 2) + ids(i:i+Lm-1) * c_p^T$; media următoarelor M valori ale lui $ids(.)^2$

$p(i) = (2 \cdot med + Lm \cdot ids_m(i-1)) / (Lm + 2)$; valoarea prezisă (număr real) a spectrului mediat

$ids_m(i) = [p(i)]$; se rotunjește la cel mai apropiat întreg

$d_f = d_f + ids(i) - ids_m(i)$; se reactualizează diferența

În Fig. 4.14 se prezintă spectrele filtrate ale dispozitivelor de întrețesere aleatoare (IS1A, IS5A, IS10A, IS20A și IS23A). Singura diferență dintre spectrele filtrate o reprezintă d_{min} .

De asemenea și dispozitivele de întrețesere de tip pseudo-aleator au spectrele distanțelor asemănătoare între ele. Însă, spectrul distanțelor pentru acestea prezintă multe valori nule (umplerea figurii indică o revenire la zero frecvență, fapt nerealizat la IDS-le interleaverelor de tip S) și de asemenea linii singulare mai pronunțate. Aceste linii singulare sunt linii spectrale cu concentrare de distanțe.

Dacă valoarea maximă a spectrului mediat pentru interleaveurile aleatoare are valoarea aproximativă 550, același maxim, pentru IDS-le interleaverelor de tip pseudo-aleator se situează undeva în apropierea valorii 1200.

Privind IDS-ul interleaverului rectangular, concluzia imediată este că acesta este un spectru de linii, acestea fiind situate la multipli de 33.

IDS-le interleaverelor pseudo-aleatoare au un suport de valoare jumătate din cel al IDS-lor interleaverelor aleatoare iar cel rectangular este cu peste un ordin de mărime mai mic. Diferența este și mai pronunțată în ce privește gradul de împrăștiere. Este de așteptat așadar ca interleaveurile de tip S (având un grad de împrăștiere mai bun) să aibă o performanță BER mai bună decât celelalte tipuri de

¹ Notațiile utilizate în algoritmul de mediere nu sunt adăugate în lista de simboluri.

² Definirea sub-vectorului $ids()$ și produsul acestuia cu vectorul c_p transpus au fost simbolizate ca și în limbajul MATLAB.

interleavere, chiar dacă, spre exemplu, interleaverul rectangular are cea mai mare distanță de întreșesere. Acest fapt este confirmat de simulările prezentate în subparagraful 4.4.6.

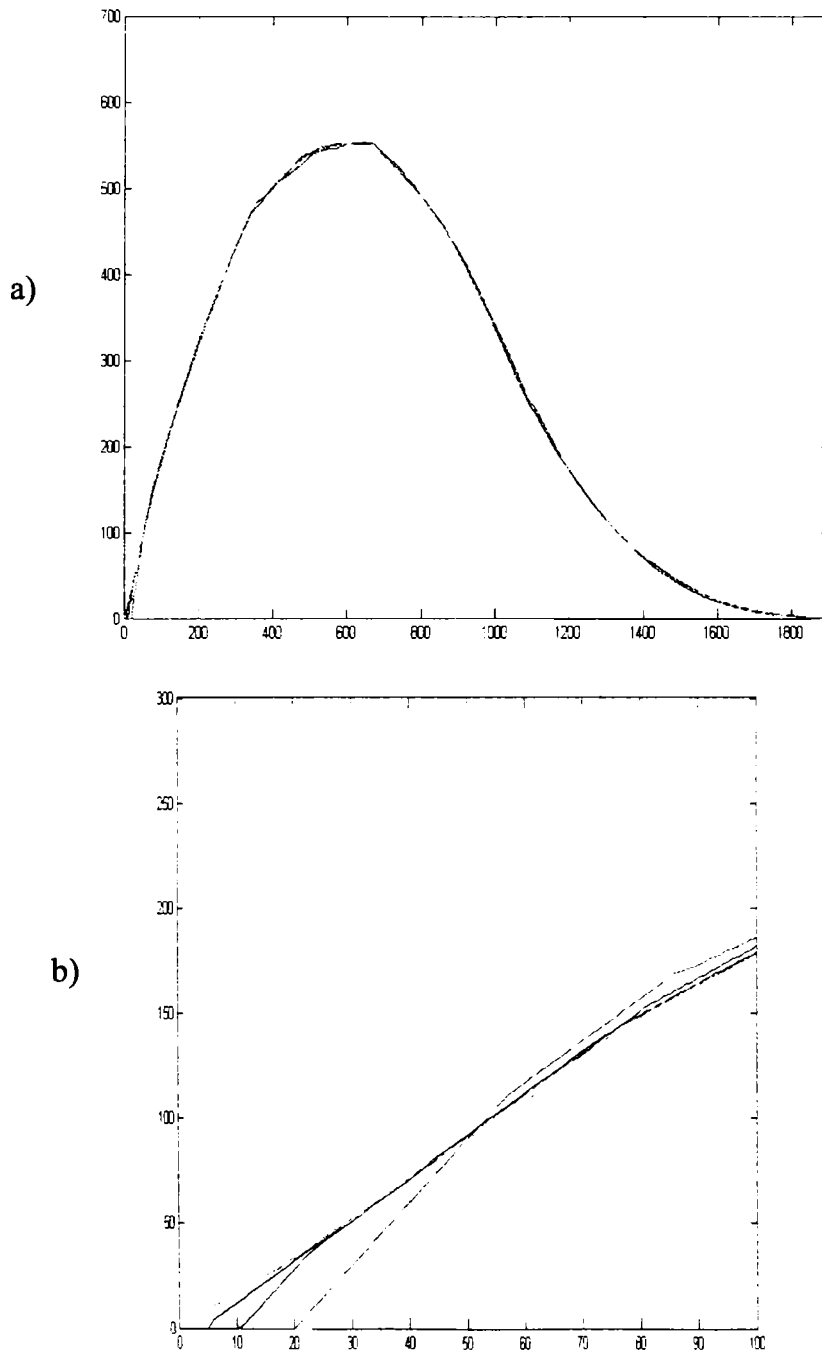


Figura 4.14 a) Spectrele distanțelor de întreșesere mediate pentru interleaver-e de tip S; b) Detaliu.

4.4.4 Calculul spectrului distanțelor pentru un S interleaver

Conform definiției:

$$ids(k) = nr[d(i,j) = k], \quad k = 2 \div (2 \cdot N - 2) \quad (4.19)$$

unde $nr[d(i,j)=k]$ reprezintă numărul de perechi (i, j) pentru care distanța de întrețesere este k . Pe de altă parte probabilitatea ca $d(i, j)=k$ este:

$$P\{d(i, j)=k\} = \frac{nr[d(i, j) = k]}{C_N^2}, \quad (4.20)$$

în care $C_N^2 = (N-1) \cdot N/2$ reprezintă numărul total de perechi (i, j) cu $1 \leq i < j \leq N$.

Probabilitatea ca $d(i,j)=k$ se poate scrie ca:

$$p\{d(i,j)=k\} = \sum_{x=1}^{k-1} P\{j-i = x\} \cdot P\{\pi(j) - \pi(i) = k-x \mid j-i = x\}. \quad (4.21)$$

Probabilitatea ca $j-i=x$, este dată de raportul dintre numărul de valori pe care poate să le ia i astfel încât să poată fi realizabile condițiile:

$$x + 1 < i + x = j \leq N, \quad (4.22)$$

și numărul total de perechi (i, j) , care este C_N^2 . Astfel:

$$P\{d(i,j)=k\} = \begin{cases} \frac{N-x}{C_N^2}, & x \leq N-1 \\ 0, & x \geq N \end{cases} \quad (4.23)$$

sau, utilizând funcția treaptă unitate, $\sigma(k)$, definită ca:

$$\sigma(k) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (4.24)$$

rezultă:

$$P\{d(i, j)=k\} = \frac{N-x}{C_N^2} (1 - \sigma(x - N)). \quad (4.25)$$

Urmărind același raționament și ținând cont de caracterul aleator al întrețeserii, fapt ce conduce la independența pozițiilor de după întrețesere $\pi(j)$ și $\pi(i)$ de pozițiile inițiale j , respectiv i , rezultă că:

$$\begin{aligned}
 P\{|\pi(j) - \pi(i)|=k-x / j-i=x\} &= P\{|\pi(j) - \pi(i)|=k-x\} = \\
 &= \frac{N-k+x}{C_N^2} (1 - \sigma(k-x-N)). \quad (4.26)
 \end{aligned}$$

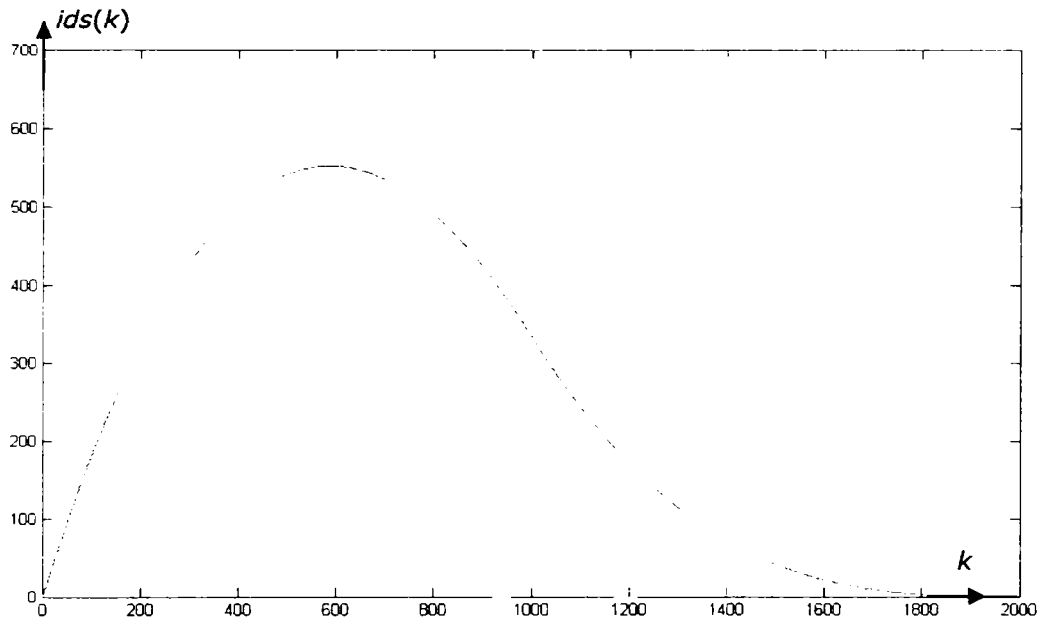


Figura 4.15 Spectrul distanțelor de întreșesere teoretic pentru interleaverul aleator cu $N=1000$

Din relațiile (4.19), (4.20), (4.21), (4.25) și (4.26), rezultă că:

$$\begin{aligned}
 ids[k] &= C_N^2 \cdot P\{d(i,j)=k\} = \\
 &= \sum_{x=1}^{k-1} \frac{(N-x) \cdot (N-k+x)}{C_N^2} (1 - \sigma(x-N)) \cdot (1 - \sigma(k-x-N)) \quad (4.27)
 \end{aligned}$$

În Figura 4.15 se prezintă graficul spectrului teoretic al unui interleaver aleator cu lungimea de 1000 de biți. Funcția definită prin relația (4.27) se poate explicita, considerând separat cazurile $k \leq N$ și respectiv $k > N$:

$$ids(k) = \begin{cases} ids_1(k), & k \leq N \\ ids_2(k), & k > N \end{cases} \quad (4.28)$$

unde:

$$ids_1(k) = \sum_{x=1}^{k-1} \frac{(N-x) \cdot (N-k+x)}{C_N^2} =$$

$$= \frac{k-1}{C_N^2} \cdot \left[\frac{k^2}{6} - k \left(N - \frac{1}{6} \right) + N^2 \right], \quad 2 \leq k \leq N \quad (4.29)$$

și:

$$ids_2(k) = \sum_{x=k-N+1}^{N-1} \frac{(N-x) \cdot (N-k+x)}{C_N^2} =$$

$$= \frac{(2N-k)^3 - (2N-k)}{6 \cdot C_N^2}, \quad N < k \leq N-2. \quad (4.30)$$

Funcția $ids_1(k)$ prezintă un maxim pentru:

$$k = k_M = 2 \cdot N - \sqrt{2 \cdot \left(N^2 - N - \frac{1}{6} \right)} \approx (2 - \sqrt{2}) \cdot N \approx 0,586 \cdot N. \quad (4.31)$$

Valoarea acestui maxim este $ids_1(k_M) \approx 0,552 \cdot N$. Pentru $N=1000$, aceste valori sunt: $k_M \approx 586$ și, respectiv, $ids_1(k_M) \approx 552$. Suportul funcției $ids(k_M)$ –spectrul distanțelor de întrețesere, are valoarea:

$$supp_o = (2 \cdot N - 2) - 1 = 2 \cdot N - 3 \quad (4.32)$$

Astfel încât gradul de împrăștiere va avea valoarea:

$$sd = supp_o / ids_1(k_M) \approx 3,62. \quad (4.33)$$

4.4.5 Spectrul distanțelor de întrețesere pentru un interleaver rectangular pătratic

În acest subparagraf este calculat IDS-ul pentru un interleaver rectangular cu lungimea $N = M \times M$. Blocul de date (biți) ce trebuie întrețesut, așa cum se arată în Figura 4.12, se organizează într-o matrice cu dimensiunile $M \times M$. Considerând că înscrierea datelor în matrice se face pe linie, de la stânga la dreapta, poziția unui bit din blocul de date se poate exprima funcție de coordonatele blocului prin relația:

$$i = p_i \cdot M + q_i + 1, \quad 0 \leq q_i < M, \quad 0 \leq p_i < M, \quad (4.34)$$

astfel că funcția interleaver este:

$$\pi(i = p_i \cdot M + q_i + 1) = q_i \cdot M + p_i + 1 \quad (4.35)$$

(citire de pe coloană), iar distanța de întrețesere va avea expresia:

$$d(i, j) = (p_j - p_i) \cdot M + q_j - q_i + |(q_j - q_i) \cdot M + p_j - p_i|. \quad (4.36)$$

Calculul IDS-lui, adică al numărului de perechi (i, j) care au aceeași distanță d va fi făcut în trei cazuri distincte:

A. $p_j = p_i, q_j > q_i$.

Acesta este cazul în care i și j sunt în aceeași linie. Datorită simetriei, situația este identică cu cea în care i și j sunt în aceeași coloană, astfel că valorile IDS-lui găsite în acest caz A trebuie dublate. În cazul A relația (4.36) devine:

$$d(i, j) = (q_j - q_i) \cdot (M + 1) = k \cdot (M + 1). \quad (4.37)$$

Numărul de perechi (i, j) având distanța $d = k \cdot (M + 1)$ este dat de numărul de posibilități ca $p_j = p_i$ și de numărul de posibilități ca $q_j = q_i + k$, astfel că:

$$ids_A(k \cdot (M + 1)) = 2 \cdot nr[p_j = p_i] \cdot nr[q_j = q_i + k] = 2 \cdot M \cdot (M - k), \quad 1 \leq k \leq M, \quad (4.38)$$

unde $nr[condiție]$ reprezintă numărul de cazuri în care *condiție* este adevărată.

B. $p_j > p_i, q_j > q_i$.

În cazul în care bitul j este situat la dreapta și mai jos față de bitul i , distanța dintre cei doi biți va fi:

$$d(i, j) = (p_j - p_i + q_j - q_i) \cdot (M + 1) = k \cdot (M + 1), \quad 2 \leq k \leq 2 \cdot M - 2. \quad (4.39)$$

Este ușor de văzut că distanța d poate lua aceleași valori ca și în cazul A . Astfel, valorile funcției IDS trebuie sumate. Calculul IDS-lui trebuie făcut separate pentru $k \leq M$, respective pentru $M + 1 \leq k$, dar utilizând funcția treaptă unitate, obținem:

$$ids_B(k \cdot (M + 1)) = \sum_{x=1}^{k-1} (M - x) \cdot \sigma(M - x) \cdot (M - k + x) \cdot \sigma(M - k + x). \quad (4.40)$$

C. $p_j > p_i, q_j < q_i$.

În cazul în care bitul j se află poziționat la stânga și mai jos de bitul i :

$$d(i, j) = (p_j - p_i + q_i - q_j) \cdot (M - 1) = k \cdot (M - 1), \quad 2 \leq k \leq 2 \cdot M - 2. \quad (4.41)$$

Tabelul 4.8 Valorile IDS pentru un interleaver rectangular $M \times M$

Distanța d	Parametrul k	Number de perechi $s(d)$
$k \cdot (M - 1)$	$1 \leq k \leq M$	$(k - 1) \cdot (M - k) \cdot M + (k - 1) \cdot k \cdot (k + 1) / 6$
$k \cdot (M - 1)$	$M + 1 \leq k \leq 2 \cdot M - 2$	$(2 \cdot M - k - 1) \cdot (2 \cdot M - k) \cdot (2 \cdot M - k + 1) / 6$
$k \cdot (M + 1)$	$1 \leq k \leq M$	$2 \cdot M \cdot (M - k) + (k - 1) \cdot (M - k) \cdot M + (k - 1) \cdot k \cdot (k + 1) / 6$
$k \cdot (M + 1)$	$M + 1 \leq k \leq 2 \cdot M - 2$	$(2 \cdot M - k - 1) \cdot (2 \cdot M - k) \cdot (2 \cdot M - k + 1) / 6$
în rest	-	0

Calculul IDS-lui este identic cu cel din cazul *B*. Singura diferență constă în faptul că distanța *d* poate lua, în acest caz, valori egale cu multipli de $M-1$ și nu de $M+1$.

În Tabelul 4.8 sunt prezentate valorile sumate ale IDS-lui pentru toate cazurile. Deoarece interleaverul rectangular este unul de tip determinist nu există diferențe între spectrele distanțelor de întretesere determinate practic și teoretic, astfel că funcția *ids* definită prin valorile Tabelului 4.8 are graficul prezentat în Figura 4.13 pentru cazul particular în care $M=32$ ($N=1024$ biți). Este vorba despre interleaverul notat IR32.

4.4.6 Rezultatele simulărilor

În Figura 4.16 a,b sunt prezentate curbele BER obținute prin simulări pe calculator cu interleaverul definite anterior. S-a simulat un turbo-cod convoluțional având două coduri componente identice, RSC, nepuncturate cu $G = [1, 5/7]$. Zgomotul din canal s-a presupus AWGN iar modulația utilizată a fost BPSK. Lungimea blocurilor de date a fost $N=1024$ pentru IR32 și IP12, iar pentru restul $N=1000$. S-au efectuat un număr de 8 iterații. Volumul simulărilor a fost funcție de raportul semnal per zgomot (SNR), astfel încât să se obțină un număr minim de erori conform Tabelul 4.9.

Tabelul 4.9 Numărul minim de erori funcție de SNR

SNR[dB]	0	0,3	0,5	0,7	0,9	1,1	1,3	1,5	1.7
Nr.er.	200000	140000	100000	60000	30000	15000	9000	3000	1000

În Figura 4.16c se prezintă simulări cu un set de alte interleavere de tip S având exact același mod de construcție cu primele. Acestea, pentru distincție, au fost notate cu „B”. Și aceste simulări confirmă concluziile rezultate din analiza primului set de interleavere de tip S (notate cu „A”): performanța BER se îmbunătățește pe vreme ce *S* crește (și implicit d_{min}), dar există posibilitatea ca

4.4.7 Concluzii

pentru *S*-uri apropiate să se obțină rezultate mai bune cu interleaverul cu grad de împrăștiere mai bun chiar dacă *S*-ul său e mai mic.

Utilizând același cod component și o lungime de întretesere de $31 \times 31 = 961$ biți, apropiată de cea utilizată în aceste simulări, în [WHO00] (pag. 2229) sunt prezentate performanțele BER/SNR ale câtorva tipuri de interleavere utilizate și în prezenta lucrare. Performanțele interleaverelor rectangular IR32 și ale interleaverelor pur aleatoare (IS1A și IS1B) sunt prezentate în Figura 4.16 din prezenta lucrare sunt practic identice cu cele prezentate în Figura 23 din [WHO00]. În mod evident alegerea unui interleaver de tip S, cu *S* de valori cât mai mari, conduce la un câștig de codare de aproximativ 0.2 dB la un BER = 10^{-4} .

În prezentul subcapitol s-a definit spectrul distanțelor pentru un interleaver și s-a construit acest spectru pentru trei tipuri de interleavere: aleator, pseudo-aleator și rectangular. S-a făcut o comparație, între interleavere de tipuri diferite și de același tip, pe baza acestor spectre și pe baza rezultatelor simulărilor făcute cu un TC ce înglobează aceste interleavere. Concluzia ce rezultă este că factorii esențiali, ce indică „calitatea” unui interleaver, constituie gradul de împrăștiere al spectrului distanțelor și distanța minimă de împrăștiere, cu o oarecare importanță mai mare a primului. Am definit un factor de calitate al interleaverului de forma:

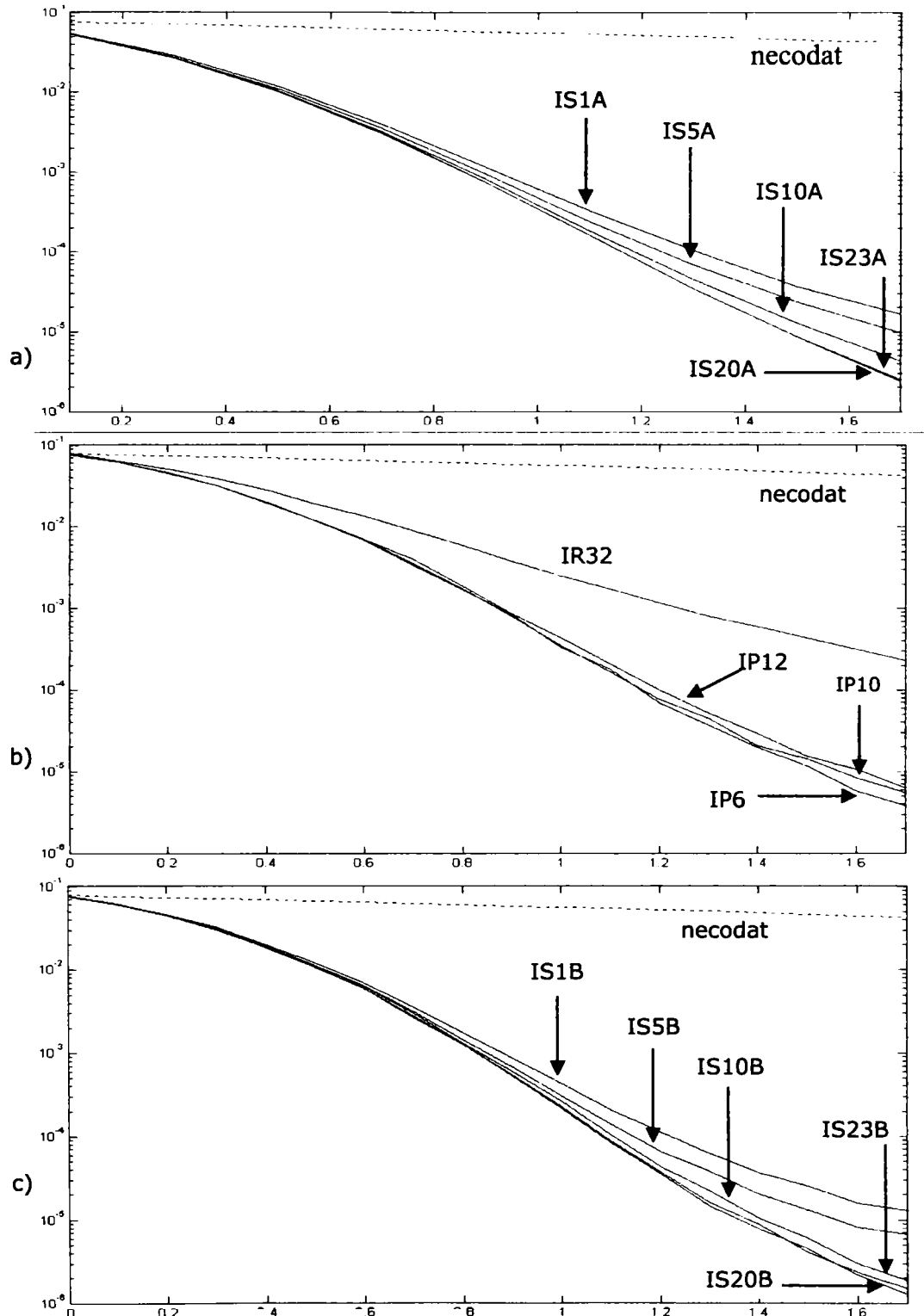


Figura 4.16 Rata erorii (BER) funcție de raportul semnal per zgomot (SNR) în dB

$$F_c = \frac{\text{supp}}{N} \cdot d_{\min}^a, \text{ cu } 0 < a < 1. \quad (4.42)$$

Pentru $a = 0,5$, în Tabelul 4.10 s-au calculat valorile factorului de calitate pentru interleavele avute în discuție. Acești factori de calitate indică că interleavele pseudo-aleatoare sunt echivalente cu S-interleavele având S de valoare 3 sau 4. Acest fapt este confirmat și de curbele BER din Figura 4.16.

Tabelul 4.10 Valorile factorului de calitate pentru interleavele investigate

	IS23A	IS20A	IS10A	IS5A	IS1A	IR32	IP10	IP6	IP12
F_c	9,00	8,50	6,12	4,52	2,60	0,68	2,99	2,30	3.25

Un bun interleaver trebuie așadar să aibă (așa cum s-a specificat la începutul paragrafului) atât o distanță minimă de întrețesere mare dar și un grad de împrăștiere mare. Cea mai bună argumentare a acestei afirmații o constituie interleaverul rectangular. El posedă cea mai mare distanță minimă de întrețesere dar cel mai mic grad de împrăștiere. Performanța sa este una foarte slabă, fapt explicabil prin gradul său mic de împrăștiere.

IDS-ul și parametrii pe care-i definește devine astfel un instrument foarte util în analiza interleaverelor, în predicția performanțelor acestora și, nu în ultimul rând, în identificarea lor.

4.5 Algoritm de decodare

4.5.1 Algoritm Maximum A-Posteriori (MAP)

Algoritm Maximum A-Posteriori este cel mai performant algoritm de decodare utilizat în TCC. El a fost propus de Bahl, Cocke, Jelinek și Raviv [BCJR74]. Algoritm MAP calculează logaritmul raportului de plauzibilitate în forma [HLY02]:

$$\begin{aligned}
 LLR(u^n|y) &= \ln \left(\frac{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = +1}} p\{s^{n-1} = \hat{s} \wedge s^n = s \wedge y\}}{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = -1}} p\{s^{n-1} = \hat{s} \wedge s^n = s \wedge y\}} \right) = \\
 &= \ln \left(\frac{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = +1}} \alpha_{n-1}(\hat{s}) \cdot \gamma_n(\hat{s}, s) \cdot \beta_n(s)}{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = -1}} \alpha_{n-1}(\hat{s}) \cdot \gamma_n(\hat{s}, s) \cdot \beta_n(s)} \right) \quad (4.35)
 \end{aligned}$$

unde:

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

$$\alpha_n(s) = \sum_{\text{toti } \hat{s}} \gamma_n(\hat{s}, s) \alpha_{n-1}(\hat{s}) \quad (4.36)$$

$\beta_n(s) = P\{y^{j>n} | s^n = s\}$ este probabilitatea ca, dată fiind starea trellisului s la momentul k , secvența recepționată din canal, după acest moment, să fie $y^{j>n}$,

$$\beta_{n-1}(\hat{s}) = \sum_{\text{toti } \hat{s}} \gamma_n(\hat{s}, s) \cdot \beta_n(s) \quad (4.37)$$

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

$$\begin{aligned} \gamma_n(\hat{s}, s) &= C \cdot e^{(u^n \cdot L(u^n) / 2)} \cdot \exp\left(\frac{E_b}{2 \cdot \sigma^2} \cdot 2 \cdot a_F \cdot \sum_{r=1}^R y_r^n \cdot x_r^n\right) = \\ &= C \cdot e^{(u^n \cdot L(u^n) / 2)} \cdot \exp\left(\frac{L_c}{2} \cdot \sum_{r=1}^R y_r^n \cdot x_r^n\right), \end{aligned} \quad (4.38)$$

în care:

$$L_c = 2 \cdot a_F \cdot E_b / \sigma^2 \quad (4.39)$$

este coeficientul de încredere al canalului, $L(u^n)$ este informația extrinsecă reordonată (întreșută), numită și apriori iar a_F este amplitudinea fadingului (dacă nu există fading atunci $a_F = 1$).

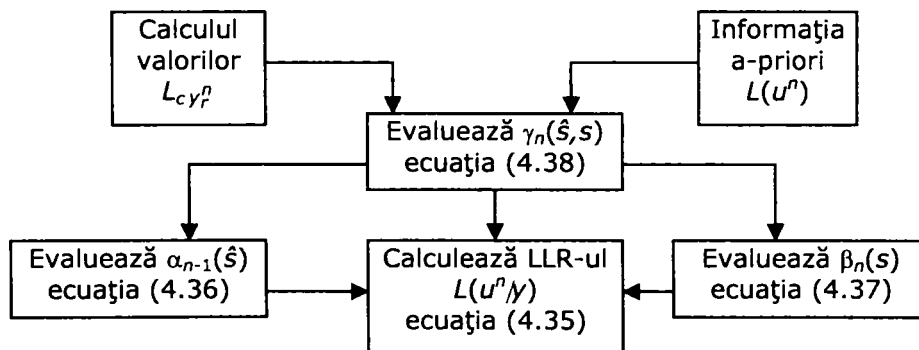


Figura 4.17 Organigrama algoritmului MAP

Rezumatul algoritmului MAP

Se calculează $\gamma_n(\hat{s}, s)$ cu Ecuația (4.38) utilizând valorile canalului y_r^n și LLR-ul a-priori $L(u^n)$ (ce este furnizat într-un decodor iterativ turbo de un alt decodor component). Constanta C poate fi omisă din calculul lui $\gamma_n(\hat{s}, s)$ deoarece se va simplifica în raportul din Ecuația (4.35). Pentru calculul valorilor $\alpha_n(\hat{s}, s)$ se utilizează recurența înainte dată de Ecuația (4.36), iar pentru calculul valorilor $\beta_n(\hat{s}, s)$ recurența înapoi din Ecuația (4.37). În final, toate valorile calculate pentru $\alpha_n(\hat{s}, s)$, $\beta_n(\hat{s}, s)$ și $\gamma_n(\hat{s}, s)$ sunt utilizate în Ecuația (4.35) pentru calculul valorilor pentru $L(u^n/\underline{y})$. Aceste operații sunt rezumate în organigrama din Figura 4.17. Pentru a evita probleme de calcul numeric în calculul recursiv pentru $\alpha_n(\hat{s}, s)$ și $\beta_n(\hat{s}, s)$, se poate proceda la o normalizare a acestor valori. Această normalizare se șterge în raportul din Ecuația (4.35) și de aceea nu cauzează schimbări în LLR-ul produs de algoritm.

Algoritmul MAP este, în forma descrisă în această secțiune, extrem de complex datorită multiplicărilor, a operatorilor exponențiali și a operatorilor logaritmi naturali utilizați. În paragrafele următoare se prezintă două versiuni ale algoritmului MAP care, cu prețul unei oarecare degradări a performanței, sunt mai puțin complexe.

Rezultate experimentale

Deoarece în majoritatea simulărilor efectuate în această lucrare s-a utilizat algoritmul MAP, în acest paragraf se prezintă un studiu asupra influenței estimării canalului asupra performanțelor algoritmului MAP, [BH17]. În acest context, prin estimarea canalului se înțelege aprecierea cantitativă a SNR-ului canalului utilizat în construcția valorii coeficientului de încredere L_c dat de (4.39). Nu am găsit în literatură o analiză similară.

Tabelul 4.11 Parametrii TC-lui pentru analiza algoritmului MAP

	Parametrul BTC-lui	Varianta aleasă pentru simulări
1	Codul component	5/7, 17/15 și 23/31
2	Configurația TC-ului	paralelă
3	Puncturarea Rata de codare	fără $(N-K+1)/(3 \cdot N) \approx 1/3$
4	Întreșeserea Lungimea blocurilor	S-ilv. ($N=400$ $S=13$ și $N=1784$ $S=29$)
5	Canal	AWGN
6	Modulație	BPSK
7	Algoritm de decodare	MAP cu L_c analizat în acest paragraf
8	Închiderea trellisului	C1-închis, C2-neînchis coeficienții $\beta(\)$ la start au fost setați echiprobabili
9	Cuantizarea	infinită
10	Număr de iterații Criteriu de oprire	15 <i>prag LLR</i> = 10

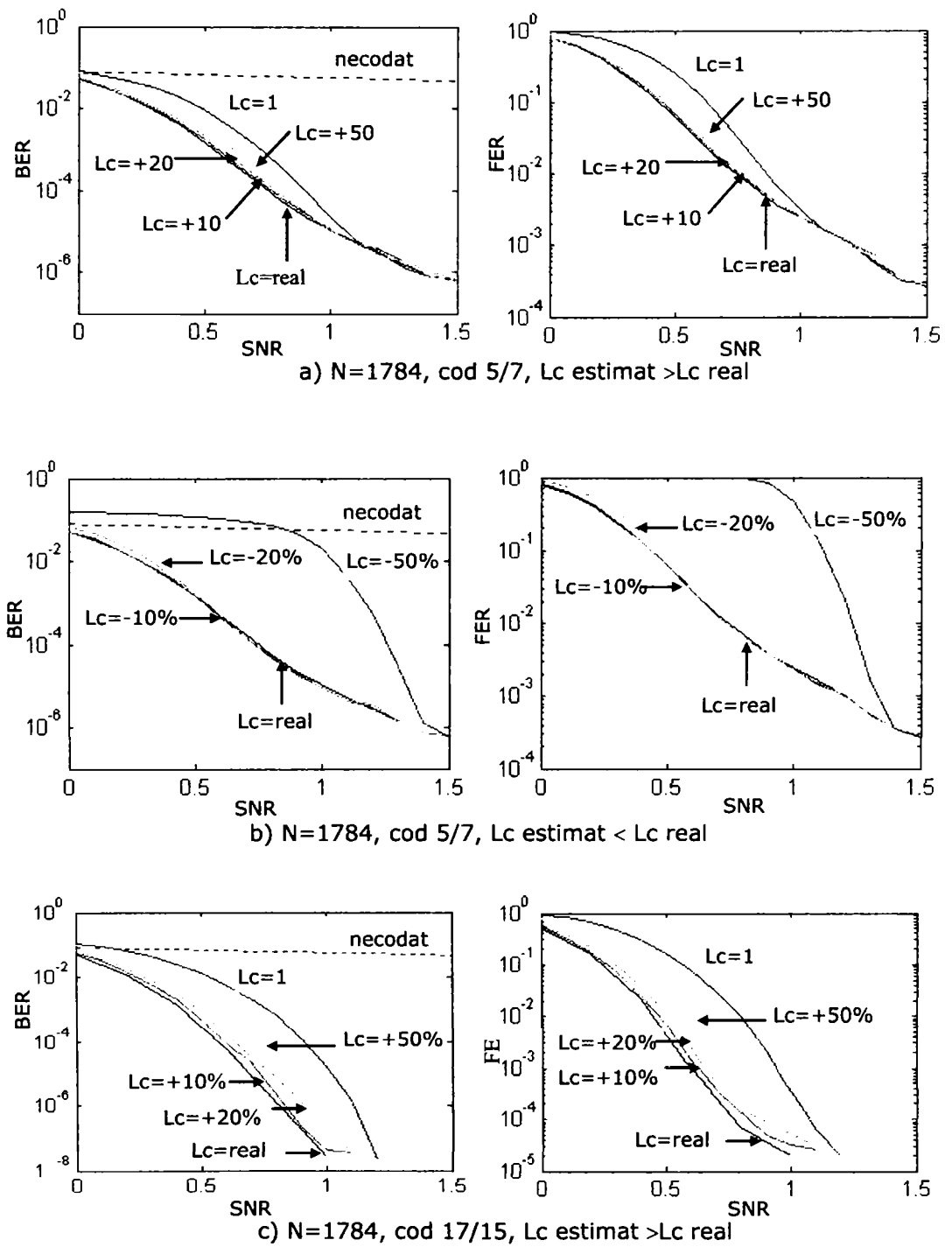


Figura 4.18 Performanțele algoritmului MAP funcție de estimatul L_c -ului

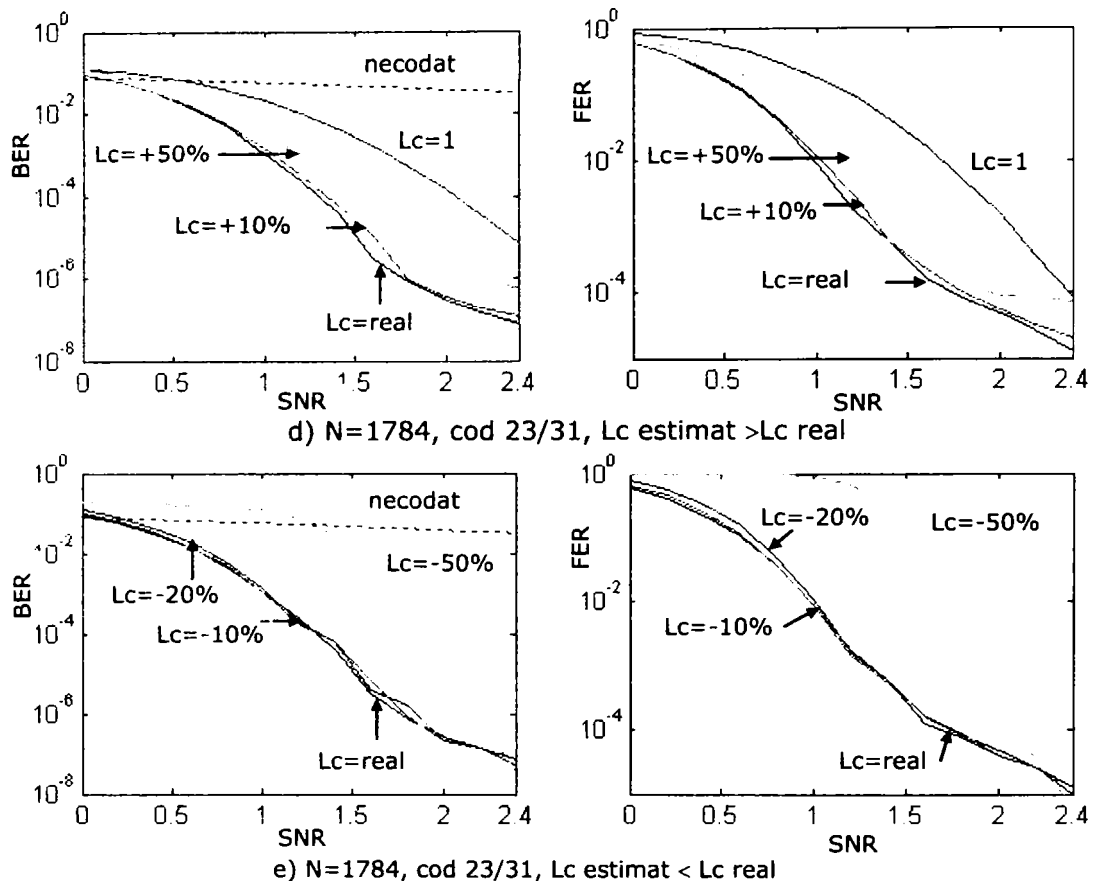


Figura 4.18 (continuare) Performanțele algoritmului MAP funcție de estimatul L_c -ului

Parametrii utilizați în simulări au fost sintetizați în Tabelul 4.11 iar rezultatele simulărilor sunt concretizate prin curbele BER(SNR) și FER(SNR), prezentate în diagramele din Figura 4.18. Curbele comparate în diagrame au fost construite pentru diferite valori ale parametrului L_c utilizat în algoritmul MAP din decodare. Astfel s-au comparat performanțele obținute cu L_c estimat greșit cu +10 procente (curbele notate cu „ $L_c=+10\%$ ”), cu +20 procente („ $L_c=+20\%$ ”) și cu +50 procente („ $L_c=+50\%$ ”) din valoarea adevărată, respectiv cu -10, -20, și -50 procente din valoarea adevărată (notate „ $L_c=-10\%$ ”, „ $L_c=-20\%$ ” și „ $L_c=-50\%$ ”). De asemenea s-a investigat cazul pentru care $L_c=1=constant$ indiferent de valoarea adevărată a SNR-ului. Acest caz a fost notat cu „ $L_c=1$ ”.

Indiferent de codul sau interleaverul utilizat, estimarea valorii SNR-ului canalului AWGN influențează performanțele algoritmului MAP utilizat în TC-uri doar până la un anumit SNR. Peste acest prag (1,1 dB, spre exemplu, pentru diagrama a)), performanțele devin identice, indiferent de valoarea L_c -ului.

În cazul în care nu există o estimare a SNR-ului, se poate alege o valoare fixă pentru L_c (în lucrare s-a propus $L_c=1$). Acest fapt poate conduce la o penalizare maximă de aproximativ 0,2 dB.

Performanțele TC-urilor practic rămân intacte dacă se face o estimare a SNR-ului, iar aceasta nu diferă cu mai mult de $\pm 20\%$ de valoarea adevărată.

4.5.2 Algoritmul MaxLogMAP

Algoritmul Max-Log-MAP, [KOB90] reprezintă o simplificare a algoritmului MAP prin transferarea recurențelor în domeniul logaritmic și prin invocarea unei aproximări care reduce dramatic complexitatea de implementare asociată. Însă, din cauza acestei aproximări performanța algoritmului Max-Log-MAP este sub-optimală.

Aproximația utilizată de algoritmul Max-Log-MAP este:

$$\ln \left(\sum_i e^{x_i} \right) \approx \max_i(x_i), \quad (4.40)$$

unde $\max_i(x_i)$ înseamnă maximul valorilor pentru x_i . Atunci, cu $A_n(s)$, $B_n(s)$ și $\Gamma_n(\hat{s}, s)$ definite astfel:

$$A_n(s) \hat{=} \ln(\alpha_n(s)), \quad (4.41)$$

$$B_n(s) \hat{=} \ln(\beta_n(s)), \quad (4.42)$$

$$\Gamma_n(\hat{s}, s) \hat{=} \ln(\gamma_n(\hat{s}, s)), \quad (4.43)$$

se poate rescrie Ecuația (4.41) ca:

$$\begin{aligned} A_n(s) \hat{=} \ln(\alpha_n(s)) &= \ln \left(\sum_{\text{toti } \hat{s}} a_{n-1}(\hat{s}) \cdot \gamma_n(\hat{s}, s) \right) \\ &= \ln \left(\sum_{\text{toti } \hat{s}} \exp[A_{n-1}(\hat{s}) + \Gamma_n(\hat{s}, s)] \right) \approx \max_{\hat{s}} (A_{n-1}(\hat{s}) + \Gamma_n(\hat{s}, s)). \end{aligned} \quad (4.44)$$

Similar:

$$\begin{aligned} B_{n-1}(\hat{s}) \hat{=} \ln(\beta_{n-1}(\hat{s})) &= \ln \left(\sum_{\text{toti } s} \beta_n(s) \cdot \gamma_n(\hat{s}, s) \right) \\ &= \ln \left(\sum_{\text{toti } s} \exp[B_n(s) + \Gamma_n(\hat{s}, s)] \right) \approx (B_n(s) + \Gamma_n(\hat{s}, s)), \end{aligned} \quad (4.45)$$

obținându-se recurența înapoi. Aceasta este echivalentă cu recurența utilizată în algoritmul Viterbi –valoarea pentru $B_{n-1}(\hat{s})$ se află prin adunarea, pentru orice stare $s^n = s$ ce are ramură dinspre $s^{n-1} = \hat{s}$ (două în trellisul binar), a metricii ramurii $\Gamma_n(\hat{s}, s)$ la valoarea $B_n(s)$ și selectând acea ramură care dă cea mai mare valoare $B_{n-1}(\hat{s})$.

Utilizând Ecuația (4.43), metricile ramurilor $\Gamma_n(\hat{s}, s)$ pot fi scrise ca:

$$\Gamma_n(\hat{s}, s) \hat{=} \ln(\gamma_n(\hat{s}, s)) = \ln \left(C \cdot e^{(u^n \cdot L(u^n) / 2)} \cdot \exp \left[\frac{E_b}{2\sigma^2} \cdot 2a \cdot \sum_{r=1}^{R+1} y_r^n \cdot x_r^n \right] \right) =$$

$$\begin{aligned}
&= \ln \left(C \cdot e^{(u^n \cdot L(u^n) / 2)} \cdot \exp \left[\frac{L_c}{2} \cdot \sum_{r=1}^{R+1} y_r^n \cdot x_r^n \right] \right) = \\
&= \hat{C} + \frac{1}{2} \cdot u^n \cdot L(u^n) + \frac{L_c}{2} \sum_{r=1}^{R+1} y_r^n \cdot x_r^n \quad (4.46)
\end{aligned}$$

unde $\hat{C} = \ln C$ nu depinde de u^n sau de cuvântul de cod transmis x^n și astfel poate fi considerat constant și omis. Așadar metrica ramurii este echivalentă cu cea utilizată în algoritmul Viterbi, cu adunarea termenului LLR a-priori $u^n \cdot L(u^n)$. În plus termenul corelație $\sum_{r=1}^{R+1} y_r^n x_r^n$ este ponderat prin valoarea de încredere a canalului L_c din Ecuația (4.39).

$$\begin{aligned}
L(u^n / \mathcal{Y}) &= \ln \left(\frac{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = +1}} a_{n-1}(\hat{s}) \cdot \gamma_n(\hat{s}, s) \cdot \beta_n(s)}{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = -1}} a_{n-1}(\hat{s}) \cdot \gamma_n(\hat{s}, s) \cdot \beta_n(s)} \right) \\
&= \ln \left(\frac{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = +1}} \exp(A_{n-1}(\hat{s}) + \Gamma_n(\hat{s}, s) + B_n(s))}{\sum_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = -1}} \exp(A_{n-1}(\hat{s}) + \Gamma_n(\hat{s}, s) + B_n(s))} \right) \\
&\approx \max_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = +1}} (A_{n-1}(\hat{s}) + \Gamma_n(\hat{s}, s) + B_n(s)) - \\
&\quad - \max_{\substack{(\hat{s}, s) \Rightarrow \\ u^n = -1}} (A_{n-1}(\hat{s}) + \Gamma_n(\hat{s}, s) + B_n(s)). \quad (4.47)
\end{aligned}$$

Aceasta înseamnă că în algoritmul Max-Log-MAP, pentru fiecare bit u^n , LLR-ul a-posteriori $L(u^n / \mathcal{Y})$ se calculează considerând toate tranzițiile de la stadiul trellisului s^{n-1} către stadiul s^n . Aceste tranziții sunt grupate între acelea ce s-ar fi putut produce doar dacă $u^n = +1$, și acelea ce s-ar fi putut produce doar dacă $u^n = -1$. Pentru ambele grupuri se află tranziția ce dă valoarea maximă pentru $A_{n-1}(\hat{s}) + \Gamma_n(\hat{s}, s) + B_n(s)$, iar LLR-ul a-posteriori este calculat doar pentru „cele mai bune tranziții”. Pentru un trellis binar vor fi $2 \cdot 2^M$ tranziții la fiecare stadiu al trellisului, unde M este memoria codului convoluțional. Așadar trebuiesc considerate 2^M tranziții la fiecare maximizare în Ecuația (4.47).

Algoritmul Max-Log-MAP se poate rezuma după cum urmează. Metrica ramurii $\Gamma_n(\hat{s}, s)$ utilizată, este dată prin Ecuația (4.46), unde termenul constant \hat{C} poate fi omis. Pentru a calcula $A_k(s)$ și $B_n(s)$ se utilizează recurențele înainte și înapoi, Ecuația (4.44) și respectiv Ecuația (4.45), ambele similare recurenței înainte utilizate în algoritmul Viterbi. O dată ce ambele recurențe înainte și înapoi au fost făcute, se poate calcula LLR-ul a-posteriori utilizând Ecuația (4.47).

4.5.3 Algoritmul Log-MAP

Algoritmul Max-Log-MAP dă o ușoară degradare în performanță comparativ cu algoritmul MAP datorită aproximării din Ecuația (4.40). Însă, această aproximare poate fi făcută exactă prin utilizarea logaritmului Jacobian:

$$\begin{aligned} \ln(e^{x_1} + e^{x_2}) &= \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|}) = \\ &= \max(x_1, x_2) + f_c(|x_1 - x_2|) \end{aligned} \quad (4.48)$$

unde $f_c(x)$ poate fi gândită ca un termen de corecție. Acesta este baza algoritmului Log-MAP propus de Robertson ș.a. [RVH95]. Similar cu algoritmul Max-Log-MAP, valorile pentru $A_n(s) = \ln(\alpha_n(s))$ și $B_n(s) = \ln(\beta_n(s))$ sunt calculate utilizând recurența înainte și înapoi. Însă, maximizările din Ecuația (4.41) și Ecuația (4.42) sunt complementate prin termenul de corecție din Ecuația (4.48). Aceasta înseamnă că sunt calculate mai exact decât valorile aproximative $A_n(s)$ și $B_n(s)$. Similar, aproximația din Ecuația (4.47) ce dă LLR-ul a-posteriori poate fi eliminat utilizând logaritmul Jacobian.

Termenul de corecție $f_c(x)$ nu necesită să fie calculat pentru fiecare valoare a lui x , ci poate fi stocat într-un tabel. Variantele algoritmului LogMAP diferă prin modul de construcție al termenului de corecție, adică prin modul de aproximare al funcției:

$$f_c(x) = \ln(1 + e^{-x}), \quad x \geq 0. \quad (4.49)$$

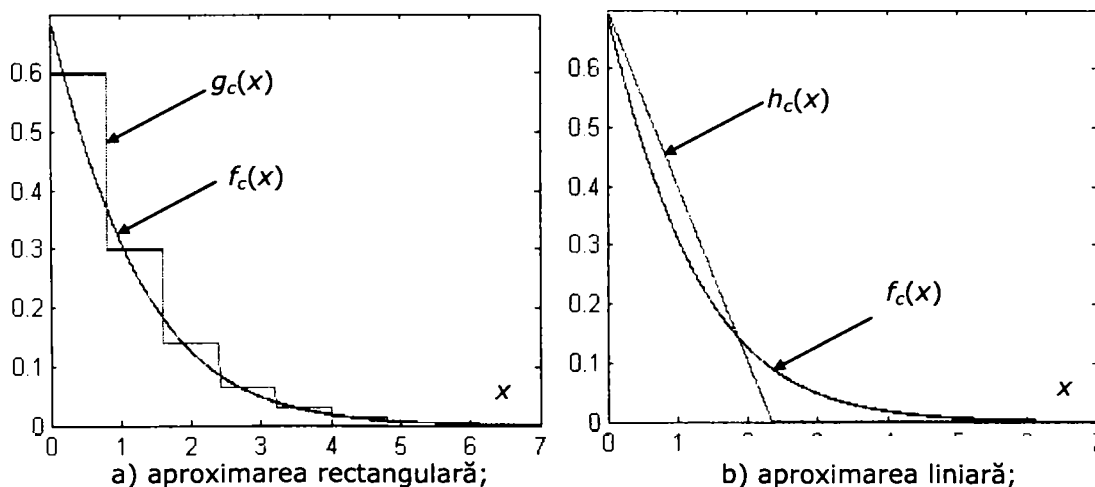


Figura 4.19 Aproximările funcției $f_c(x)$

Funcțiile ce aproximează $f_c(x)$ trebuie să fie simplu de implementat și să reproducă cât mai fidel forma funcției $f_c(x)$. În continuare sunt prezentate două aproximări pentru $f_c(x)$ indicate în Figura 4.19.

Aproximarea rectangulară, $g_c(x)$, a fost propusă de Robertson ș.a. în [RVH95] și este o extrapolare de ordin zero a funcției $f_c(x)$. Valorile funcției $g_c(x)$ utilizate în implementarea algoritmului LogMAP rectangular (numit în continuare LogMAPR) sunt {0,6 0,3 0,14 0,065 0,03 0,014 0,005 0,002 0}.

Aproximarea liniară, $h_c(x)$:

$$h_c(x) = \begin{cases} 0,7 - \frac{0,7}{x_0} \cdot x, & x \leq x_0 \\ 0, & x > x_0 \end{cases}, \quad (4.50)$$

este dată în [BH7]. Valoarea lui x_0 pentru care $h_c(x)$ aproximează cel mai bine pe $f_c(x)$ (în sensul minimizării ariei cuprinse între graficele funcțiilor $h_c(x)$ și $f_c(x)$) este:

$$x_0 = 2,347. \quad (4.51)$$

Pentru distincție, algoritmul construit pe baza aproximației liniare (4.49) va fi numit în continuare LogMAPL.

4.5.4 Performanțele algoritmilor de decodare. Concluzii

În Figura 4.20 sunt comparate performanțele obținute cu cei patru algoritmi de decodare descriși anterior: MAP, MaxLogMAP, LogMAPR și LogMAPL. Parametrii TC-ului utilizat sunt descriși în Tabelul 4.12.

Pentru SNR-uri sub 0,6 dB algoritmul LogMAPL lucrează aproape identic cu algoritmul MAP. În mod surprinzător performanțele algoritmilor LogMAP tind, spre valori mari ale SNR-ului, să depășească în performanță algoritmul MAP. În plus

Tabelul 4.12 Parametrii TC-lui pentru studiul algoritmilor de decodare

	Parametrul BTC-lui	Varianta aleasă pentru simulări
1	Codul component	15/13
2	Configurația TC-ului	paralelă
3	Puncturarea Rata de codare	fără $(N-M)/(3 \cdot N) \approx 1/3$
4	Întrețeserea Lungimea blocurilor	S-iv. $(N=1784 \quad S=29)$
5	Canal	AWGN
6	Modulație	BPSK
7	Algoritm de decodare	analizat în acest paragraf
8	Închiderea trellisului	C1-închis, C2-neînchis coeficienții $\beta(\)$ la start au fost setați echiprobabili
9	Cuantizarea	infinită
10	Număr de iterații Criteriu de oprire	15 $prag \ LLR = 10$

trebuie notat și faptul că LogMAPL este mai ușor de implementat practic. Algoritmul MaxLogMAP este cu circa 0,3 dB mai puțin performant decât celelalte la un BER de 10^{-5} . Acesta este prețul plătit pentru creșterea vitezei de decodare și scăderea complexității ei.

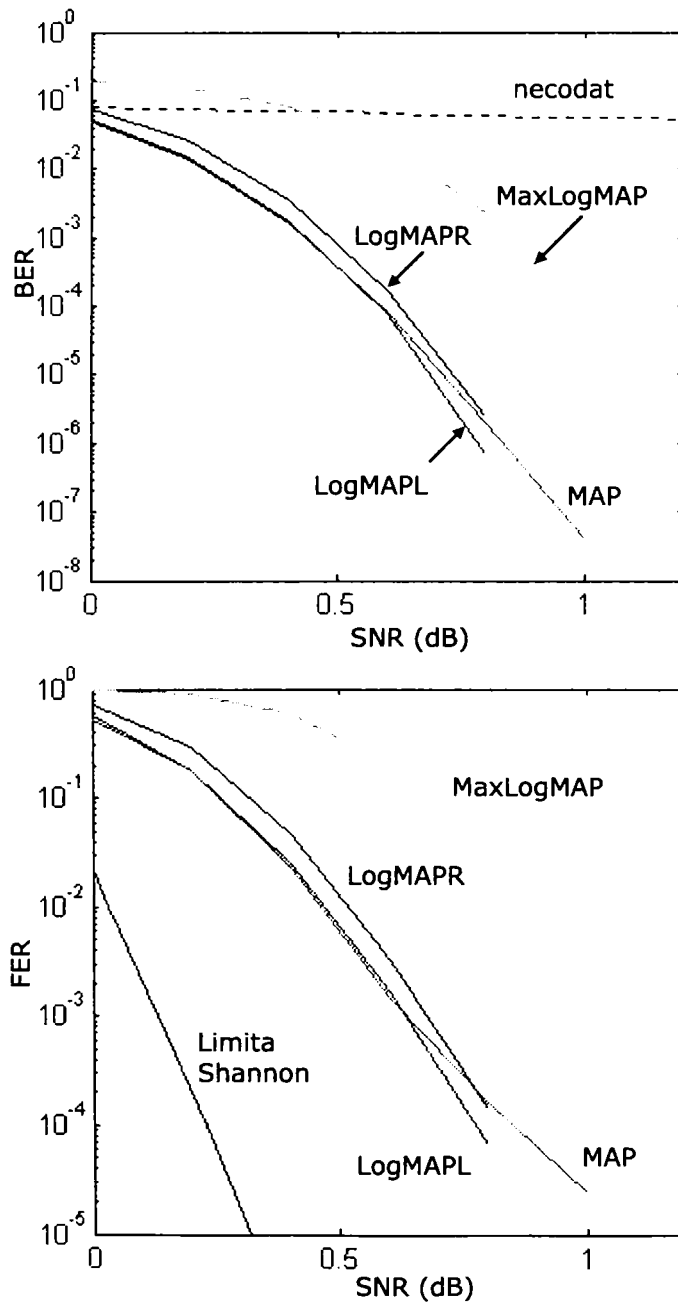


Figura 4.20 Performanțele algoritmilor de decodare

Comparații între algoritmi de decodare se pot găsi în [RVH95] sau în [WHO00]. În [RVH95] autorii propun în premieră algoritmul LogMAPR și-l compară, atât din punct de vedere al performanței cât și din punct de vedere al complexității de decodare, cu algoritmi de decodare cunoscuți la data respectivă (SOVA MAP și MaxLogMAP). În [WHO00], pag. 2225 Figura 15, se face o comparație între algoritmi de decodare MAP, LogMAP (rectangular), MaxLogMAP și SOVA. Performanțele prezentate în Figura 4.20 din prezenta lucrare și cele prezentate în Figura 15 din [WHO00] conduc la concluzii similare relativ la raportul dintre performanțele algoritmilor de decodare. Există totuși câteva diferențe.

În primul rând, în prezenta lucrare, nu am investigat algoritmul SOVA deoarece nu se mai folosește practic la ora actuală. În al doilea rând, în simulările proprii am utilizat un cod convoluțional component de memorie 3 (15/13) față de unul de memorie 2 (5/7) în [WHO00]. În plus am utilizat o lungime de 1784 biți față de 1000 de biți cât s-a utilizat în [WHO00]. Datorită alegerii diferite a acestor parametri curbele prezentate în Figura 4.20 sunt situate la stânga celor din [WHO00]. Însă diferențele între curbe se păstrează.

În al treilea rând, simulările din [WHO00] s-au oprit la un BER de 10^{-4} , în vreme ce curbele din Figura 4.20 a prezentei lucrări continuă până în jurul valorii de 10^{-6} pentru BER. Astfel în [WHO00] nu au putut fi văzute tendințele de depășire în performanță ale algoritmilor LogMAP în detrimentul algoritmului MAP.

4.6 Închiderea trellisului

4.6.1 Introducere

Închiderea trellisului condiționează modul cum vor fi inițializați coeficienții alfa și beta. Închiderea trellisului la începutul său presupune ca la codare, înainte de efectuarea acesteia, să se încarce codorul cu o stare anume, pe care o va cunoaște și decodorul, inițializându-și în mod corespunzător coeficienții alfa și beta. Cu excepția codării circulare, această stare de început este cea nulă. Închiderea trellisului la sfârșitul său (Figura 4.21a) este mai dificil de realizat, făcându-se (exceptând codul circular, Figura 4.21c) cu prețul inserării a M biți (memoria codorului) redundanți în secvența de informație.

Acest fapt conduce la scăderea ratei de codare de la valoarea $1/2$ la valoarea:

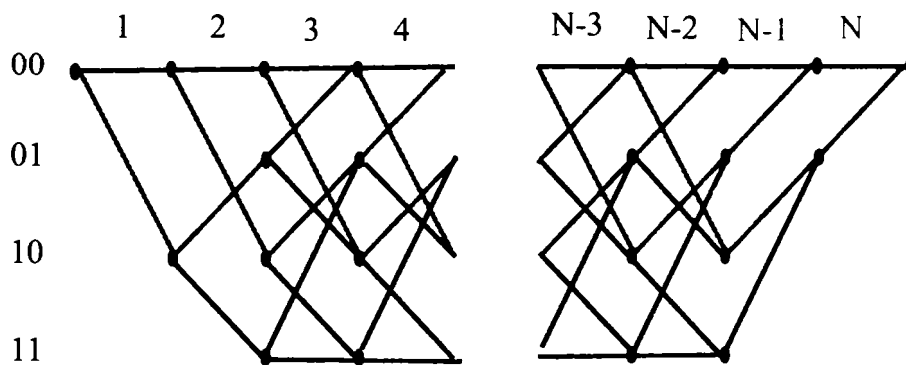
$$R_c = (N - M) / 2N. \quad (4.52)$$

Închiderea trellisului oferă avantajul cunoașterii stării de început (și/sau de sfârșit), fapt ce conduce la cunoașterea fermă a coeficienților alfa (la începutul trellisului) și beta (la sfârșitul trellisului, dacă este cazul). În cazul neînchiderii trellisului (Figura 4.21b), acești coeficienți pot fi doar prezși probabilistic.

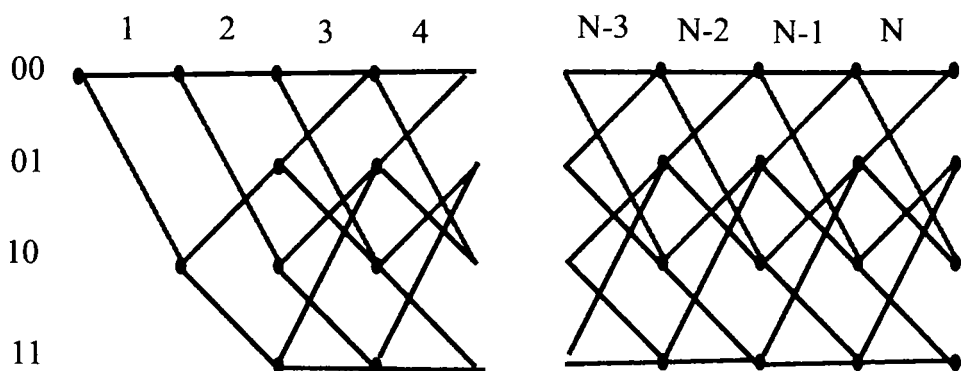
Un turbo-cod implică cel puțin două codoare, C_1 și C_2 , existând două trellisuri, fiecare putând fi terminat într-un mod diferit de celălalt. În Tabelul 4.13 sunt prezentate câteva metode de terminație a trellisurilor unui turbo-cod (paralel). Acestea vor fi detaliate în cele ce urmează.

Tabelul 4.13 Variante de închidere a trellisului

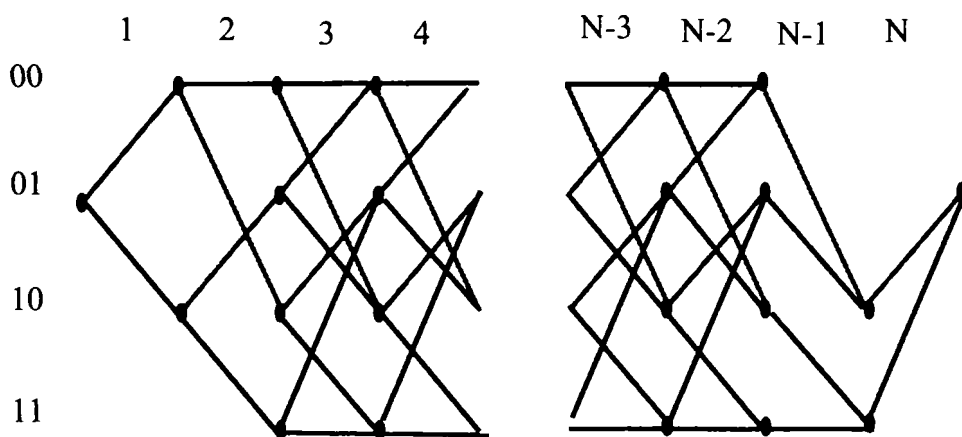
Varianta	Start C_1, C_2	Final C_1, C_2	Rata codare
01	0, 0	0, ?	$(N-M)/3N$
11	0, 0	?, ?	$1/3$
C	S_x, S_y	S_x, S_y	$1/3$



a) trellis închis (terminat) la starea nulă



b) trellis neînchis (neterminat)



c) trellis circular

Figura 4.21 Moduri de închidere a trellisurilor

4.6.2 Variante de închidere a trellisurilor unui TC

TC în varianta închis/neînchis (TC01)

În această variantă primul codor închide trellisul la ambele capete inserând M biți redundanți alături de cei $N-M$ biți de informație. Cel de-al doilea codor nu poate să procedeze în același mod datorită întreteserii secvenței de intrare. Astfel al doilea trellis nu este terminat.

Primul decodor inițializează coeficientul alfa, ce corespunde pasului unu și stării nule, la probabilitate 1 iar pe restul la 0. La fel procedează și cu coeficienții beta de la ultimului pas. Al doilea decodor procedează la fel cu coeficienții săi alfa. Pentru coeficienții beta, al doilea decodor poate proceda într-unul din modurile descrise mai jos.

01s. –egalează coeficienții beta cu coeficienții alfa de la ultimului pas. Acest mod de inițializare a coeficienților beta a fost propus în [ROB94]. Vom numi în continuare acest mod de inițializare „soft”;

01h. –inițializează coeficientul beta, ce corespunde stării pentru care coeficientul alfa (de la ultimul pas) este maxim, la probabilitate 1 iar pe restul la 0. Acest mod de inițializare a coeficienților beta a fost propus în [BGT03]. Vom numi în continuare acest mod de inițializare „hard”;

01e. –face echiprobabilitate între coeficienții beta. Vom numi în continuare acest mod de inițializare „echiprobabil”.

TC în varianta neînchis/neînchis (TC11)

Este varianta în care nici unul dintre trellisuri nu este terminat. Avantajul acestei variante este că oferă o rată de codare mai mare. Însă acest lucru nu este sesizabil dacă N este mult mai mare decât M . Acum ambele decodare trebuie să inițializeze coeficienții beta într-unul din cele trei moduri descrise anterior. Pentru simulările prezentate s-a implementat doar modul de inițializare soft. Așa cum arată și simulările făcute, această variantă este mai puțin performantă decât cea anterioară.

Cazul închiderii ambelor trellisuri este posibil doar cu modificarea corespunzătoare a întreteserii dintre cele două codoare.

TC circular (TCC)

Există posibilitatea ca, printr-o precodare, pentru orice secvență de date x , să se găsească o stare S_0 , în care să se regăsească codorul dacă s-a pornit din ea. Codarea devine astfel *circulară*. Decodorul nu cunoaște S_0 , însă știe că poate utiliza starea de final drept stare de început. Pentru aceasta însă trebuie să efectueze cel puțin o recurență înainte. Am implementat și simulat următoarele variante de turbocod circular:

C1 –decodorul efectuează o recurență înainte și calculează o stare de final S_0 . Inițializează cu S_0 coeficienții alfa și beta și reface ambele recurențe. Memorează noua stare S_0 pentru startul din următoarea iterație.

C2 –decodorul efectuează ambele recurențe în varianta soft și reține pentru următoarea iterație, cu rol de S_0 , valorile beta de la finele recurenței.

C3 – decodorul efectuează ambele recurențe în varianta soft plus încă una doar pentru coeficienții alfa. Starea de început pentru această a doua recurență este dată de valorile finale ale coeficienților beta de la prima iterație. Starea memorată pentru iterația următoare este dată de către finele celei de-a doua recurențe înainte.

C4 – decodorul efectuează recurența înainte și construiește o stare S_0 printr-o decizie hard (caută coeficientul alfa maxim). Reține această stare pentru următoarea iterație și totodată reface ambele recurențe.

4.6.3 Rezultate experimentale. Concluzii

Problematika închiderii trellisurilor codurilor convoluționale componente a fost discutată în numeroase lucrări, ca de exemplu [BGT936], [ROB94] sau [BHW95]. Nu am găsit însă un studiu asupra performanțelor diferitelor moduri de închidere a trellisurilor și/sau de inițializare a coeficienților recurențiali asemeni celui prezentat în acest paragraf.

În Figura 4.22 sunt prezentate rezultatele simulărilor efectuate cu variantele de TC descrise în paragraful anterior. Parametrii TC-ului utilizat sunt descriși de Tabelul 4.14.

Din analiza diagramelor reiese că toate variantele de TC analizate lucrează identic la SNR-uri mici. Acesta se datorează faptului că la SNR-uri mici erorile se produc datorită selecției greșite (de către turbo-decodor) a căii prin trellis iar ponderea erorilor datorate neînchiderii adecvate a trellisului este mică. Abia la valori ale SNR-ului peste 1 dB această pondere devine sesizabilă.

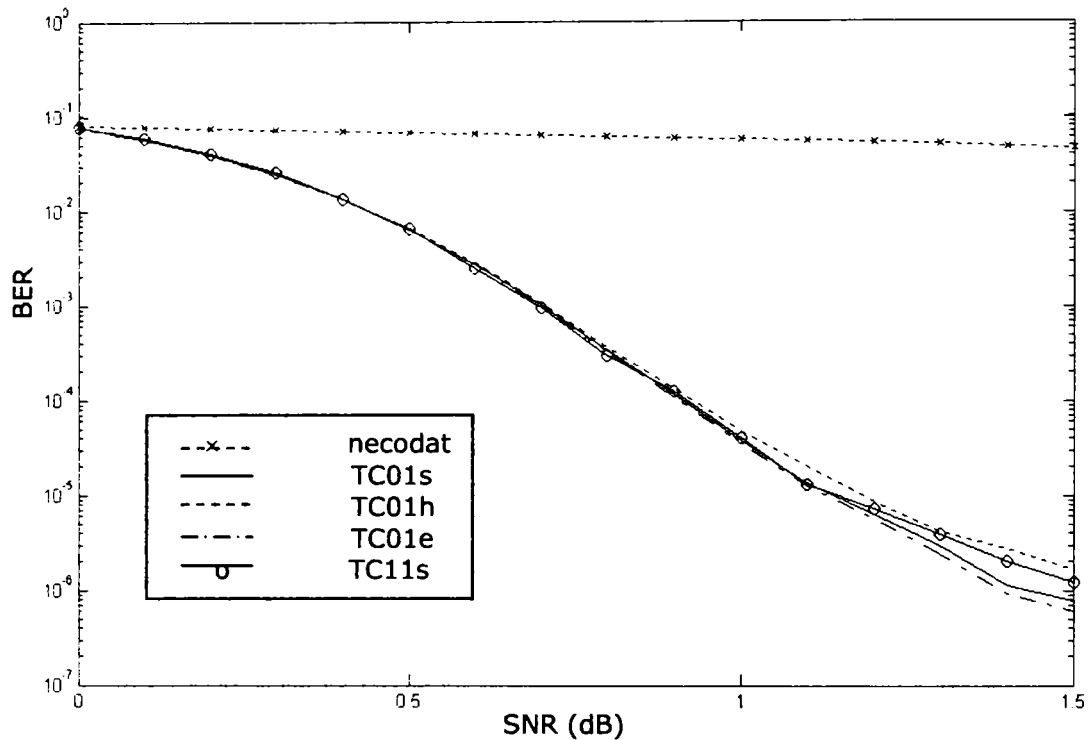
Dintre variantele TC01 simulările recomandă varianta „echiprobabilă” ca fiind cea mai potrivită. Superioritatea acestei variante are următoarea explicație. Algoritmul MAP se bazează pe independența informațiilor furnizate de coeficienții alfa și beta. În varianta „e” coeficienții beta sunt setați independent de valorile coeficienților alfa.

Inițializarea coeficienților prin varianta „hard” (TC01h și TCC4) este o soluție slabă în comparație cu celelalte. Aceasta se datorează „supraestimării” unei stări finale în detrimentul celorlalte.

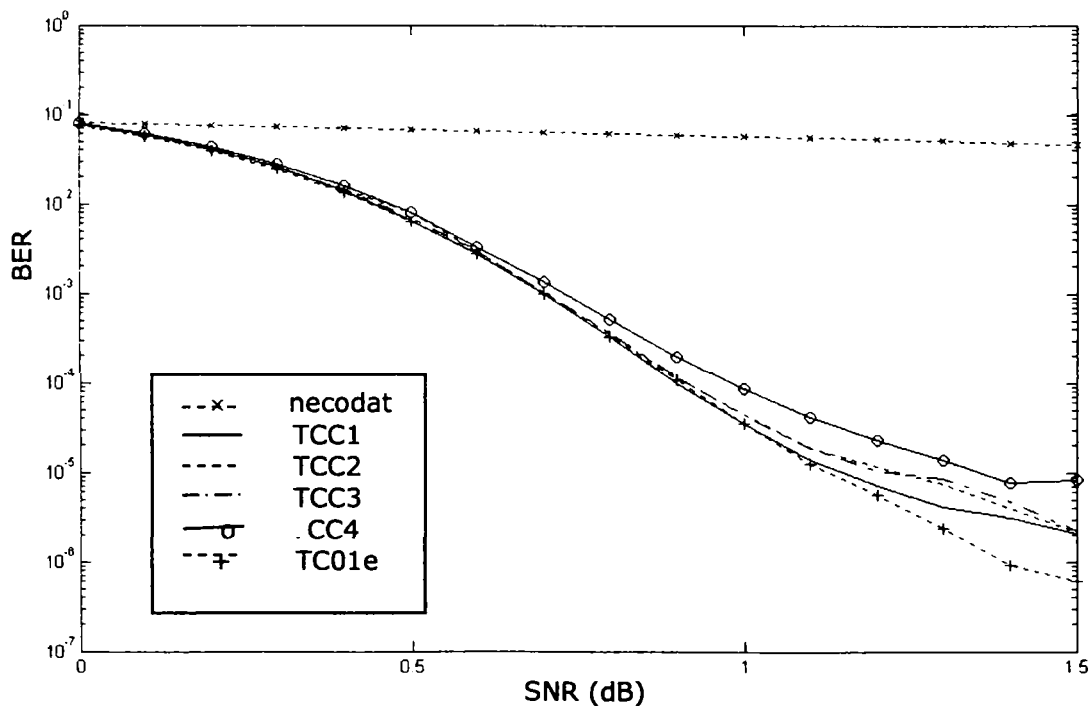
În ceea ce privește TCC-urile, trebuie remarcat faptul că variantele cu trei recurențe nu sunt mai performante decât cea cu două recurențe. Oricum există o anumită penalitate în câștigul de decodare a turbo-codului circular comparativ cu cel închis. Avantajul primului este o rată de codare cu ceva mai mare.

Tabelul 4.14 Parametrii TC-lui utilizat la investigarea variantelor de închidere a trellisurilor

	Parametrul BTC-lui	Varianta aleasă pentru simulări
1	Codul component	5/7
2	Configurația TC-ului	paralelă
3	Puncturarea Rata de codare	fără $(N-K+1)/(3 \cdot N) \approx 1/3$
4	Întreșeserea Lungimea blocurilor	S-ilv. ($N=1784$ $S=29$)
5	Canal	AWGN
6	Modulație	BPSK
7	Algoritm de decodare	MAP
8	Închiderea trellisului	sub investigație în acest paragraf
9	Cuantizare	infinită
10	Număr de iterații Criteriu de oprire	8 fără



a) variantele închis/neînchis și neînchis/neînchis;



b) variantele circulare;

Figura 4.22 Performanțele turbo-codurilor în diferitele variante de închidere a trellisurilor

4.7 Puncturarea

4.7.1 Introducere

Procedeeul de puncturare este utilizat în scopul măririi ratei de codare a turbo-codului [ACR99] [OBS97]. Schema bloc a unui turbo-cod puncturat este prezentată în Figura 4.23. Secvența u este turbo-codată de codoarele convoluționale C1 și C0 prin intermediul interleaverului I, rezultând secvențele c_1 și c_0 , de lungime N , aceeași cu a secvenței de date u . Pentru un TC nepuncturat, aceste secvențe sunt transmise direct în canal. Rata de codare ce rezultă (fără puncturare) este $R_c=1/3$. Pentru a obține rate de codare mai mari se poate utiliza puncturarea secvențelor c_1 și c_0 , procedeu ce constă în ștergerea sau eliminarea unor biți din aceste secvențe (operație realizată prin blocurile P1 și P0 din Figura 4.23).

Biții șterși (puncturați) sunt stabiliți prin matricea de puncturare, G_p , ce conține zero-uri și unu-uri, și care poate fi interpretată ca o „mască”, [OBS97]. Spre exemplu, fie $G_p = [1 \ 0 \ 1; \ 0 \ 1 \ 0]$. Prima linie din G_p constituie masca pentru

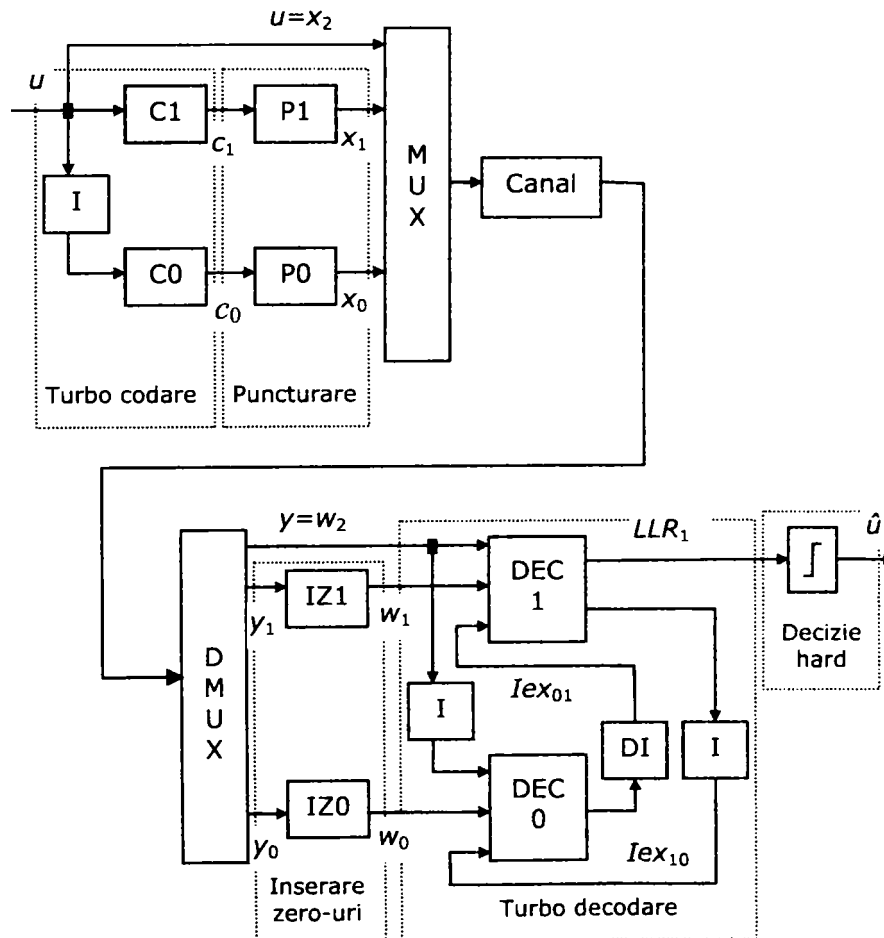


Figura 4.23 Schema unui sistem de tranmisie turbo-codat cu puncturare

puncturarea secvenței c_1 , mai precis din trei biți, va fi șters doar al doilea (biții unu și trei sunt mascați operației de ștergere). Pentru secvența c_0 , primul și al treilea din trei biți vor fi șterși, al doilea fiind mascat. După ștergere, lungimile secvențelor rezultate, x_1 și x_0 , sunt $2 \cdot N/3$ și, respectiv, $N/3$, ceea ce conduce la o rată de turbo-codare $R_{cp}=1/2$. În general, dacă dimensiunea matricii G_p este $2 \times p$ și conține N_1 de unu-uri, atunci rata turbo-codului este:

$$R_{cp} = p/(p+N_1) < 1 \quad (4.53)$$

Secvențele ce rezultă prin codare și puncturare, după multiplexare, sunt transmise prin canal. La recepție, după reconstrucția celor trei secvențe y_0 , y_1 și y_2 prin de-multiplexare, se refac lungimile de N biți prin inserția de zero-uri în locul biților șterși la emisie (operații executate de blocurile IZ1 și IZ0). După aceasta, procesul de turbo-decodare este aplicat secvențelor w_0 , w_1 și w_2 , ca și în cazul fără puncturare. Trebuie reținut faptul că valorile eșantioanelor corespunzătoare secvențelor recepționate din canal pot fi (în lipsa zgomotului) $+1$ sau -1 (transmisie bipolară sau în dublu curent), și nu $+1$ și 0 . În acest fel, inserarea zero-urilor la recepție se traduce printr-o indecizie asupra valorilor respectivilor biți. Procesul de turbo-decodare este cel care va reface valorile și pentru biții indeciși.

4.7.2 Comparație între matricile de puncturare

În acest paragraf se face o comparație asupra modului cum este influențată performanța TC-urilor prin alegerea matricii de puncturare. În figurile 4.24, 4.25 sunt prezentate 18 curbe BER/FER, reproduse conform cu [BH16]. Tabelul 4.15 prezintă matricile de puncturare și codurile utilizate, cât și modul de etichetate a curbelor din Figura 4.24. Restul parametrilor TC-urilor simulate sunt prezentați în Tabelul 4.16.

Tabelul 4.15 Matricile de puncturare

R_c	Matricea de puncturare		Codul component - forma octală			
	Nr.	P	K=3	K=4	K=4	K=5
			5/7	15/13	13/15	23/31
1/2	1	10 01	P1C1	P1C2	P1C3	P1C4
1/2	2	1001 0110	P2C1	P2C2	-	-
1/2	3	10001 01110	P3C1	P3C2	-	-
1/2	4	110 001	P4C1	P4C2	-	-
1/2	5	1110 0001	P5C1	P5C2	-	-
3/5	6	100 001	P6C1	P6C2	P6C3	P6C4
2/3	7	1000 0100	P7C1	P7C2	-	-

Tabelul 4.16 Parametrii TC-lui puncturat

	Parametrul BTC-lui	Varianta aleasă pentru simulări
1	Codul component	5/7, 15/13, 13/15 și 23/31
2	Configurația TC-ului	paralelă
3	Întreșeserea Lungimea blocurilor	S-ilv. $N=1784$ $S=29$
4	Puncturarea	sub investigație în acest paragraf
5	Canal	AWGN
6	Modulație	BPSK
7	Algoritm de decodare	MAP
8	Închiderea trellisului	C1-închis, C2-neînchis coeficienții $\beta(\)$ la start au fost setați echiprobabili
9	Cuantizare	infinită
10	Număr de iterații Criteriu de oprire	15 $prag LLR = 10$

În diagramele 1 și 2 (BER și FER) se face o comparație între diferite matrici de puncturare pentru codurile componente 5/7 și 15/13, la aceeași rată de codare, $R_c=1/2$. Analizând rezultatele se observă că: i) cele mai bune rezultate au fost obținute utilizând matrici simetrice; ii) trebuie să existe o corelație între matricea de puncturare și codul convoluțional implicat. Astfel, s-au obținut rezultate slabe cu matricea P4 și codul 5/7, însă foarte bune cu aceeași matrice și codul 15/13.

În graficele 3 și 4 se pot vedea, comparativ, performanțele codurilor la aceleași matrici de puncturare P1 și, respectiv P6 (pentru $R_c=1/2$ și respectiv $R_c=3/5$). Performanțele obținute prin codurile având lungimea de constrângere $K=4$ sau 5 sunt similare, la un BER peste 10^{-6} .

Graficele 5 și 6 prezintă performanțele obținute pentru diferite rate de codare și codurile componente: 5/7 ($K=3$) și 15/13 ($K=4$). Performanța codului 5/7 la $R_c = 3/5$ este mult mai slabă decât la $R_c=2/3$. Acest lucru se datorează necorelării dintre matricea de puncturare și codul component. Pentru codul 15/13 performanța descrește invers proporțional cu rata de codare. La $BER=10^{-5}$ diferența este de aproximativ 0.5 dB.

4.7.3 Concluzii

Puncturarea este o metodă simplă de „ajustare” a ratei de codare la o anumite valoare dorită pentru creșterea eficienței transmisiei. Puncturarea a fost prezentă și în prima publicație despre turbo-coduri, [BGT93], și este un procedeu folosit în aproape toate aplicațiile turbo-codurilor din prezent, [SRI05].

Înainte de a prezenta propriile concluzii asupra influenței puncturării asupra performanței BER a turbo-codurilor, trebuie subliniat faptul că puncturarea este dependentă de metoda de întreșesere utilizată, așa cum se arată în [OBS97]. În simulările a căror rezultate sunt prezentate în Figura 4.24 s-a utilizat un singur tip

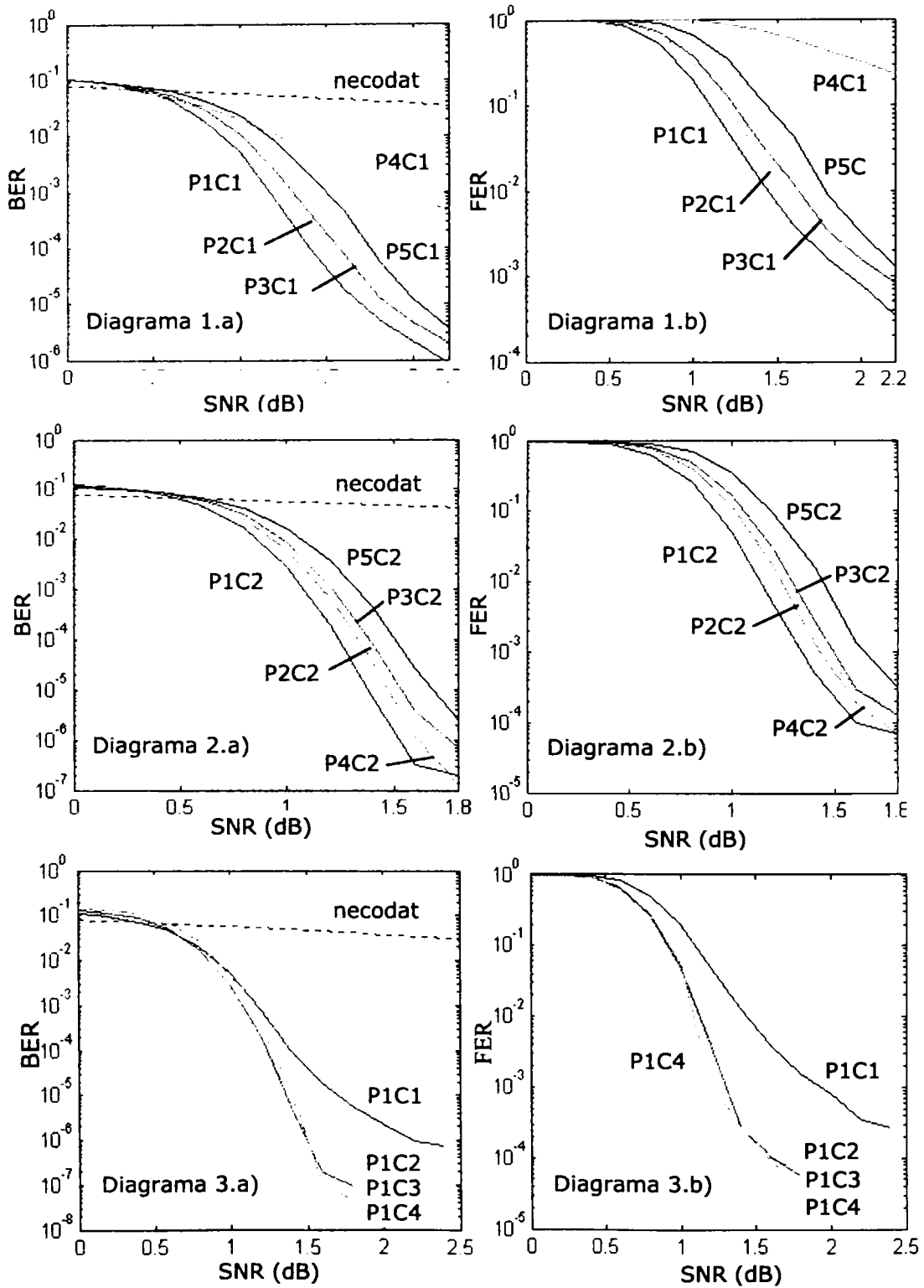


Figura 4.24 Performanțele TC-lor puncturate

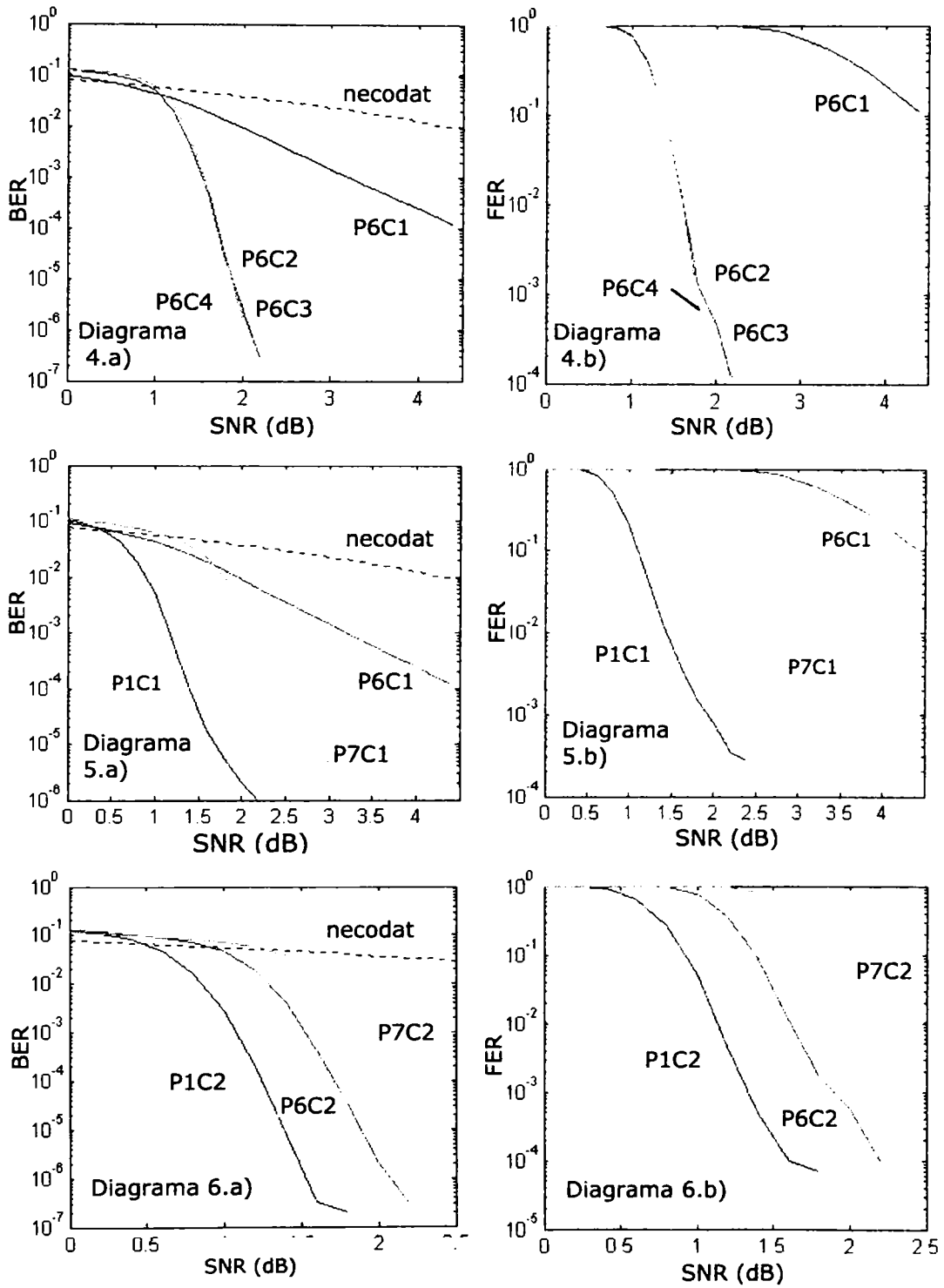


Figura 4.24(continuare) Performanțele TC-ilor puncturate

de întrețesere (de tip S), astfel că aceste rezultate sunt focalizate mai mult asupra performanțelor diferitelor matrici de puncturare.

Graficele din Figura 4.24 arată că degradarea performanței TC-ului ce rezultă prin puncturare crește pe măsură ce crește rata de codare, însă pentru a putea obține „maximum” la o anumite rată de codare este necesar a se alege matricea de puncturare în corelație cu codul convoluțional component. Cele mai potrivite matrici de puncturare sunt cele ce prezintă simetrie în linie. Codurile componente având $K \geq 4$ prezintă performanțe similare la $BER > 10^{-7}$, astfel că rata de codare este factorul decisiv ce influențează performanța TC-ului.

4.8 Numărul de iterații și criteriile de oprire ale iterațiilor

4.8.1 Influența numărului de iterații asupra câștigului de codare

Pe măsură ce procesul de decodare avansează, setul valorilor LLR furnizează, prin semnul lor, cu tot mai mare acuratețe, polaritățile biților din secvența transmisă u^n , [BH9]. De la o iterație la alta valorile LLR-urilor biților secvenței u^n , ce au valoare logică 1, se deplasează spre valori pozitive, în vreme ce LLR-urile biților din u^n cu valoare logică 0 se deplasează spre valori negative. Curbele BER și FER obținute după fiecare iterație a simulării funcționării unui TC având parametrii definiți în Tabelul 4.17, sunt prezentate în Figurile 4.25, 4.26, 4.27.

Tabelul 4.17 Parametrii TC-lui utilizat pentru investigarea criteriului de stop

	Parametrul	Varianta (-ele)
1	Codul component	5/7, 15/13, și 25/31
2	Configurația TC-ului	paralelă
3	Întrețeserea Lungimea blocurilor	S-ilv. ($N=1784$ $S=29$) 1784 biți, conform cu recomandarea CCSDS
4	Puncturarea	fără
5	Canal	AWGN
6	Modulație	BPSK
7	Algoritm de decodare	MAP
8	Închiderea trellisului	C1-închis, C2-neînchis coeficienții $\beta()$ la start au fost setați echiprobabili
9	Cuantizarea	Infinită
10	Număr de iterații Criteriu de oprire	analizat în acest paragraf

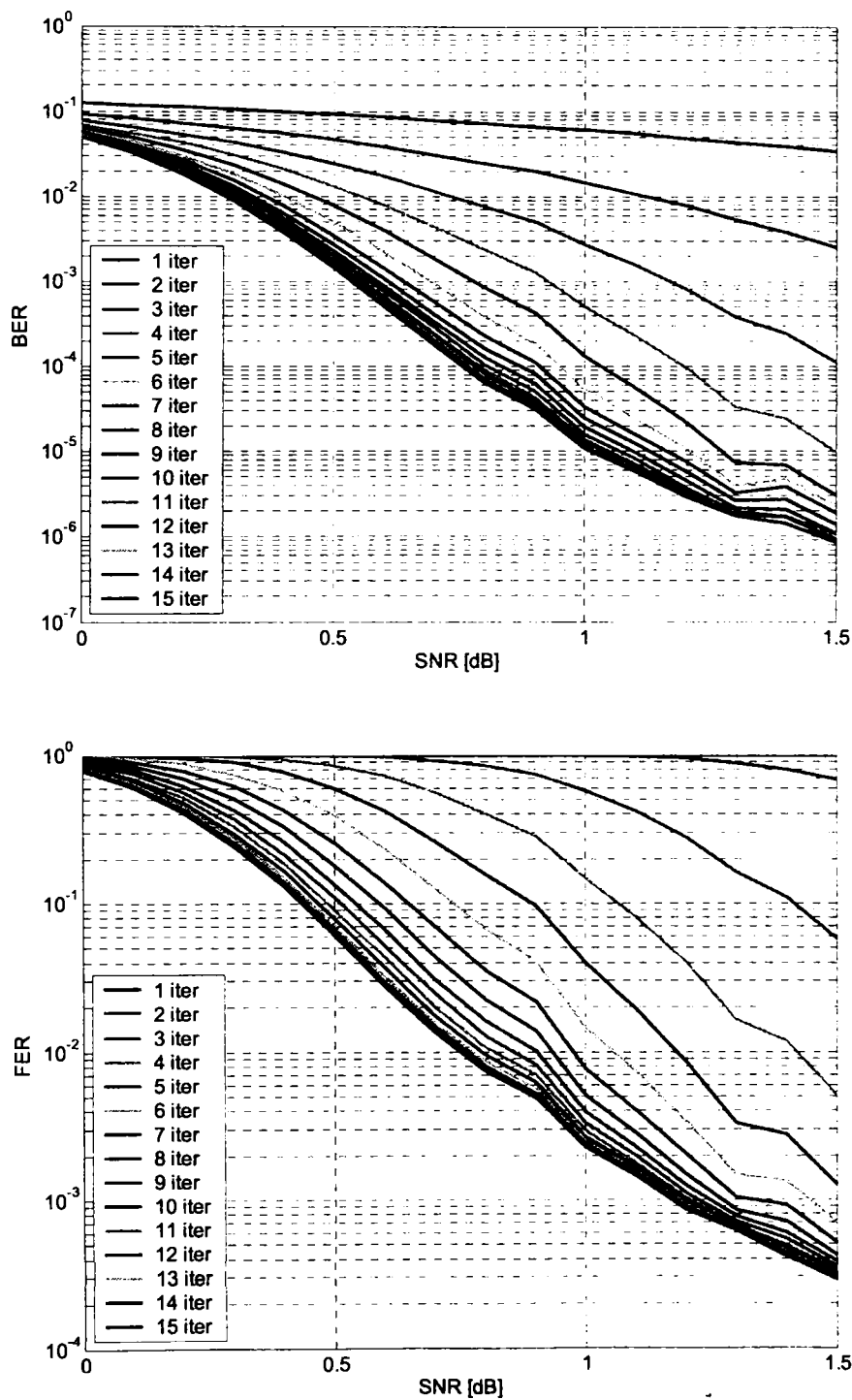


Figura 4.25 Performanțele funcție de numărul de iterații ale TC-ului construit cu codul 5/7

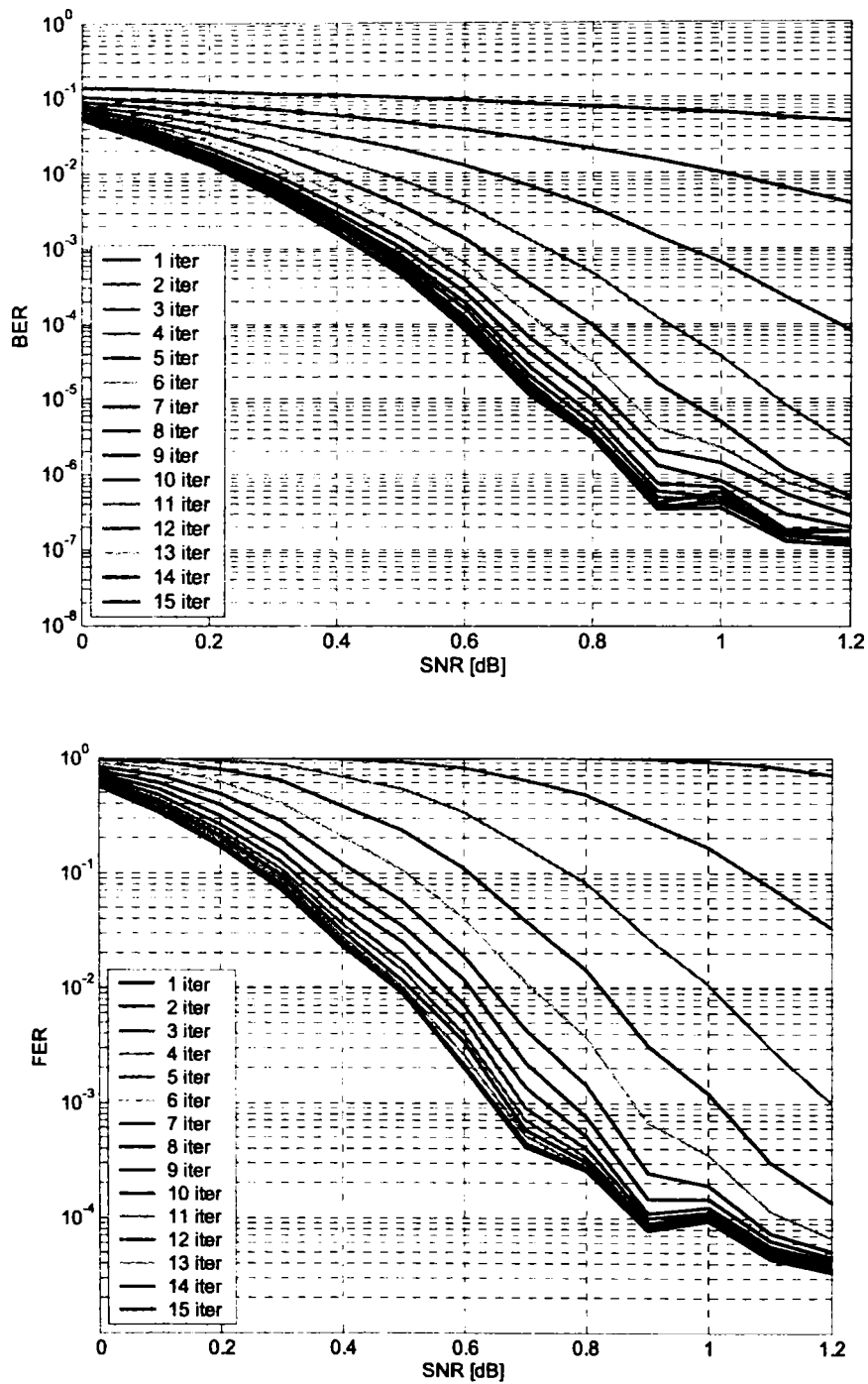


Figura 4.26 Performanțele funcție de numărul de iterații ale TC-ului construit cu codul 15/13

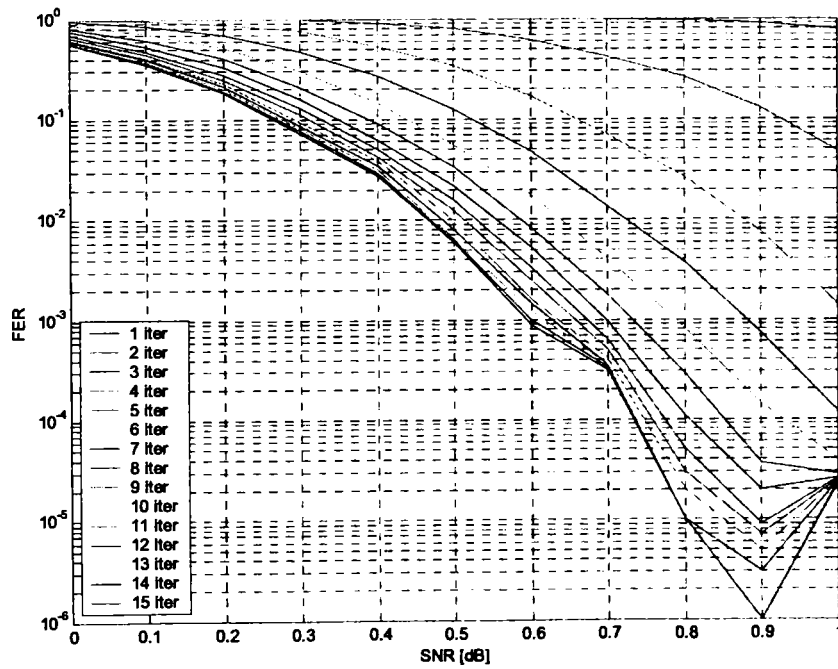
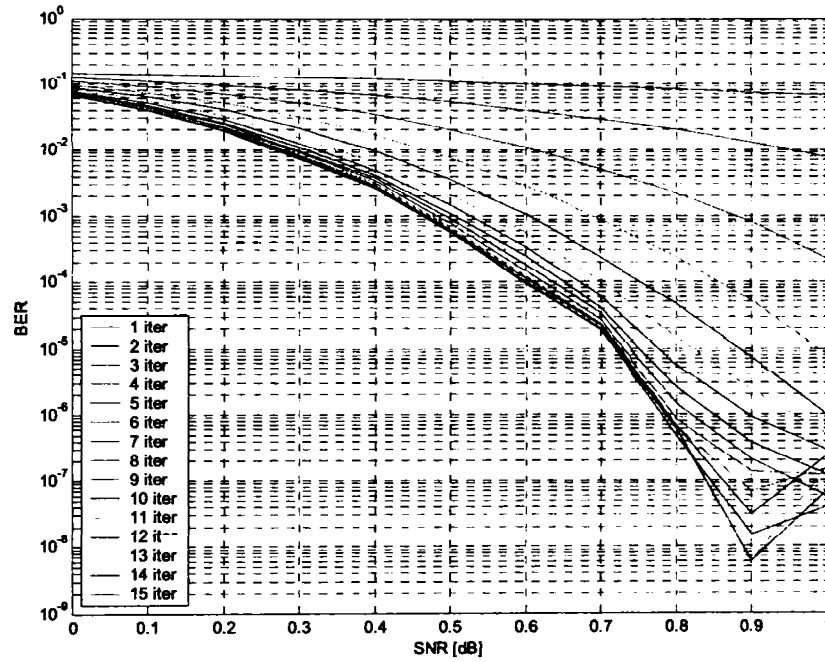


Figura 4.27 Performanțele funcție de numărul de iterații ale TC-ului construit cu codul 25/31



Tabelul 4.18 Timpul de decodare (fără criteriu de oprire)

memorie	cod	Timp (sec)	
		1,6 GHz	533MHz
2	5/7	280	460
3	15/13	585	915
4	25/31	1207	1946

Tabelul 4.19 Variația câștigului de codare cu numărul de iterații

Iter	SNR(iter=1)-SNR(iter) [dB]		
	5/7	15/13	25/31
1	4.871	5.474	5.798
2	2.176	2.064	1.962
3	0.740	0.689	0.618
4	0.305	0.273	0.223
5	0.224	0.148	0.102
6	0.066	0.085	0.059
7	0.042	0.034	0.037
8	0.036	0.024	0.026
9	0.033	0.029	0.015
10	0.027	0.020	0.011
11	0.021	0.013	0.007
12	0.014	0.008	0.004
13	0.008	0.008	0.004
14	0.005	0.002	0.004
15	0.006	0.005	0.003

Creșterea mică (gradientul) de după iterația a 10-a nu justifică efectuarea iterațiilor 11-15. În plus există blocuri pentru care TC-ul generează mai multe erori după iterația a 15-a decât după (spre exemplu) a 10-a iterație. Acest fenomen, de „recul” [BH9], greu sesizabil datorită medierii simulărilor repetate, face ca performanțele TC-ului să scadă dramatic la BER-uri mici.

Strategia unui număr maxim de iterații (15) cu un criteriu de oprire (descriș în paragraful următor) va permite, pe de o parte, renunțarea la iterațiile inutile pentru un bloc ce a fost corectat din mai puțin de 15 iterații iar pe de altă parte, eliminarea efectului de recul asupra blocurilor corectate. Utilizarea unui criteriu de oprire va reduce implicit și timpul de decodare.

Pentru comparație, în Tabelul 4.18 s-a evaluat timpul de decodare (fără criteriu de oprire) pentru 1000 de blocuri transmise la un Pentium 4 având un procesor de 1.6 GHz și la un Pentium 3 având un procesor de 533MHz. Timpul de decodare este constant pentru coduri de aceeași memorie și se dublează la creșterea memoriei cu o unitate datorită dublării numărului de stări ale trellisului. Tabelul 4.19 prezintă câștigul obținut cu fiecare iterație, la un BER=10⁻⁵.

4.8.2 Criteriu de oprire pe baza LLR-ului

În ipoteza unui canal AWGN și utilizând modulația BPSK, LLR_n este o variabilă aleatoare normal distribuită:

$$p_{LLR}(\lambda) = \frac{1}{\sigma_{\lambda} \cdot \sqrt{2\pi}} \cdot \exp\left\{-\frac{(\lambda - m_{\lambda})^2}{2 \cdot \sigma_{\lambda}^2}\right\} \quad (4.54)$$

unde, în funcție de semnul bitului transmis, media m_{λ} ia valori pozitive sau negative. Pentru valori pozitive ale biților, histograma setului de valori $LLR_{k=1+N}$, atunci când datele sunt aleatoare, include două sub-seturi ce au forme gaussiene, centrate pe $m_{\lambda 1}$ și respectiv pe $m_{\lambda 0} \approx -m_{\lambda 1}$ cu dispersii aproximativ egale.

Pentru același număr de iterații, dacă SNR-ul crește, valorile LLR (cele două gaussiene) se depărtează de origine și se subțiază (adică vor avea medii absolute mai mari și dispersii mai mici). Acest fapt este observabil în diagramele prezentate în Figura 4.28 construite pentru același bloc de date cu codul $G=[1, 5/7]$ și 15 iterații la trei valori ale SNR-ului.

Tabelul 4.20 prezintă mediile și dispersiile valorilor pozitive și negative în cele trei cazuri considerate.

Faptul că, la $SNR=0,1\text{dB}$, cele două curbe gaussiene se întrepătrund, arată că există erori de decodare (există valori ale lui k pentru care $LLR_k \cdot uk < 0$). Același lucru se produce și pentru valori mari ale SNR-ului, pentru un număr mic de iterații (Figura 4.28d).

Tabelul 4.20 Parametrii histogramelor LLR-urilor

SNR [dB]	media valorilor LLR		dispersia valorilor LLR	
	LLR neg	LLR poz	LLR neg	LLR poz
0,1	-4.642	4.409	11.360	10.269
0,5	-31.815	31.797	2.282	2.510
1,0	-32.162	32.161	1.748	1.813

Am găsit că, în medie după 10 iterații, cele două curbe gaussiene sunt separate total de un interval simetric în jurul originii ce, cu rare excepții, nu conține valori LLR. Din acest moment BER-ul nu se va mai îmbunătăți semnificativ, iar iterațiile rămase, cu rare excepții, sunt inutile. Putem utiliza acest fapt prin implementarea următorului criteriu:

Procesul de decodare iterativ al unui bloc de date va fi oprit dacă:

$$|LLR_n| > \mu, \quad n = 1 \div N, \quad (4.55)$$

unde μ este un prag a cărui valoare va fi discutată în secțiunea următoare.

Dacă condiția cerută în Ecuația 4.55 este îndeplinită iterațiile rămase nu se mai efectuează, ci se procedează la reconstrucția secvenței binare $u^{n=1+N}$ printr-o decizie hard asupra LLR-urilor rezultate după ultima iterație efectuată.

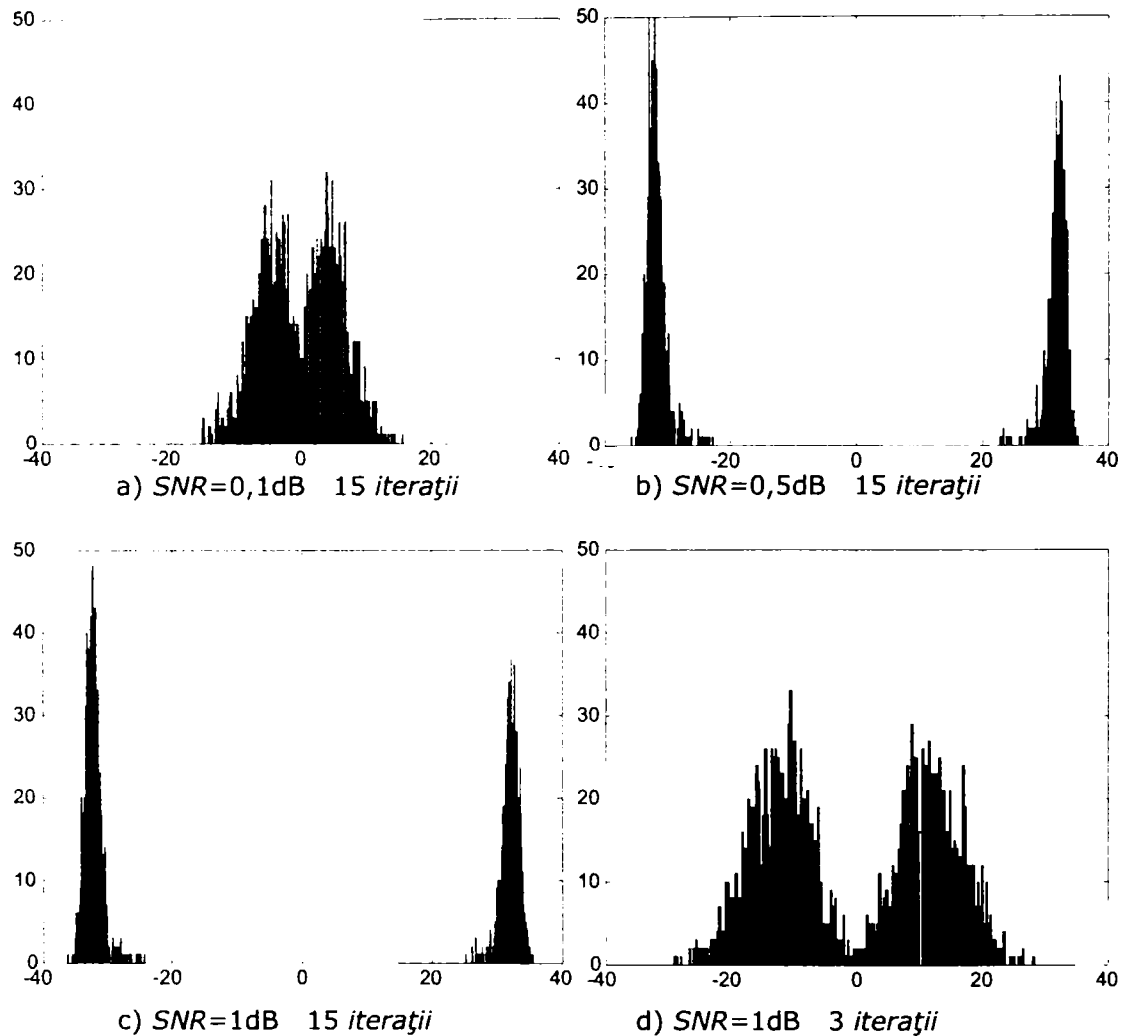


Figura 4.28 Histograme ale valorilor LLR (pe abscisă –valori ale LLR-ului, pe ordonată –număr de biți ce au valoarea LLR dată pe abscisă)

Turbo-codul având criteriul de oprire (TCS) va face o eroare în plus față de cel fără criteriul de oprire dacă există cel puțin o valoare din setul LLR-urilor corespunzătoare biților pozitivi mai mică decât $-\mu$ și toate celelalte valori sunt în exteriorul intervalului $[-\mu, \mu]$ sau dacă există cel puțin o valoare din setul LLR-urilor corespunzătoare biților negativi mai mare decât μ și toate celelalte valori sunt în exteriorul intervalului $[-\mu, \mu]$.

Presupunând că:

- i) evenimentele menționate anterior sunt independente;
- ii) secvența transmisă este pur aleatoare ($P\{0_{emis}\} = P\{1_{emis}\}$) și
- iii) $m_{\lambda 1} = -m_{\lambda 0} = m_{\lambda}$ și $\sigma_{\lambda 1} = \sigma_{\lambda 0} = \sigma_{\lambda}$, probabilitatea ca TCS să facă o eroare în plus față de TC este, [BH9]:

$$P\{\mu\} = C_{N/2}^1 \cdot a^{N/2-1} \cdot b \quad (4.56)$$

unde:

$$a = 0,5 \cdot \operatorname{erfc} \left[\frac{\mu - m_A}{\sqrt{2} \cdot \sigma_A} \right] \quad b = 1 - 0,5 \cdot \operatorname{erfc} \left[\frac{-\mu - m_A}{\sqrt{2} \cdot \sigma_A} \right], \quad (4.57)$$

iar $\operatorname{erfc}(\cdot)$ este funcția complementară eroare.

4.8.3 Performanțele TCS-ului. Concluzii

Pornind de la analiza histogramelor obținute pentru codurile 5/7, 15/13 și 25/31, s-au ales trei praguri $\mu_1=10$, $\mu_2=10$ și $\mu_3=10$. Figura 4.29 prezintă simulările efectuate cu TCS cu cele trei coduri utilizând cele trei praguri, cu un start de 15 iterații.

Tabelele 4.21, 4.22 și 4.23 corespund celor trei coduri, în care:

r.it.e. = rata iterațiilor efectuate (numărul mediu de iterații efectuate de turbo-decodor per bloc raportat la 15 -numărul maxim de iterații);

r.b.s. = rata blocurilor stopate (raportul dintre numărul de blocuri la care s-a aplicat criteriul de oprire raportat la numărul total de blocuri transmise);

t.b.s. = numărul total de blocuri simulate.

Tabelul 4.21 Performanța TC-lui 5/7 funcție de pragul din criteriul de stop

μ	r.it.e. (%)	r.s.b. (%)	t.s.b. $\times 10^3$
5	26,3	99,4	2373
10	30,2	99,2	2558
20	42,3	94,6	3661

Tabelul 4.22 Performanța TC-lui 15/13 funcție de pragul din criteriul de stop

μ	r.it.e. (%)	r.s.b. (%)	t.s.b. $\times 10^3$
5	25,0	99,8	2092
10	27,9	99,7	1742
20	31,8	94,3	2242

Tabelul 4.23 Performanța TC-lui 25/31 funcție de pragul din criteriul de stop

μ	r.it.e. (%)	r.s.b. (%)	t.s.b. $\times 10^3$
5	22,4	99,8	2368,7
10	24,0	99,8	2379,9
20	27,1	99,6	2562,7

Atât Figura 4.29 cât și probabilitatea calculată prin Ecuația 4.56 arată că practic TCS nu face erori în plus față de TC. ($P(\mu)$ are valori sub 10^{-10} pentru orice set μ , m_N , σ_A dintre cele discutate anterior). Există chiar posibilitatea ca TCS să depășească TC în performanță datorită diminuării efectului de recul.

Pentru aprecierea timpului de decodare, discuția trebuie să cuprindă atât SNR cât și μ . Astfel, pentru SNR-uri mici rata de stop a iterațiilor este mică pentru μ_2 și μ_3 și nu există diferențe între performanțele TCS(μ_1), TCS(μ_2) și TCS(μ_3). Pentru valori mari ale SNR-ului (unde BER este sub 10^{-5}) performanțele TCS(μ_1) sunt cu ceva inferioare TCS(μ_2) și TCS(μ_3). Identitatea performanțelor pentru TCS(μ_2) și TCS(μ_3) nu justifică alegerea unui prag superior lui 10, prag ce ar conduce la un timp de decodare mai mare. În concluzie soluția aleasă este μ_2 , dar există și posibilitatea utilizării unui prag variabil funcție de SNR.

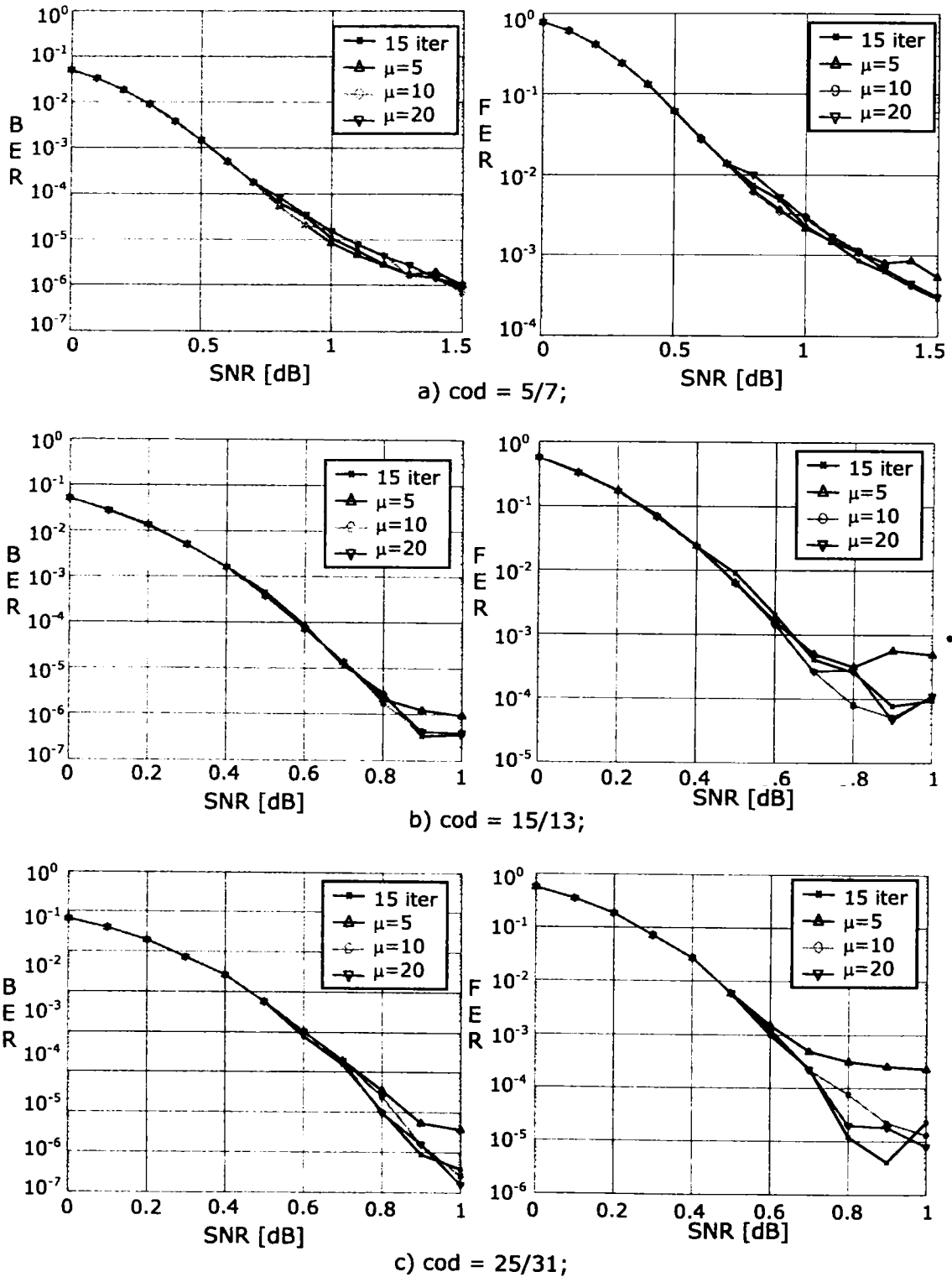


Figura 4.29 Performanțele TCS-ului

5. Turbo-Coduri Multi-Binare

5.1 Configurații de codoare convoluționale multi-binare

5.1.1 Introducere

Recent, C. Berrou ș.a., [DOB05], [BEJ99], [BER97], [BJDK01], au propus, o nouă familie de turbo-coduri multi-binare (MBTC) bazată pe concatenarea paralelă a două coduri convoluționale componente, de rată $R/(R+1)$, care au performanțe globale superioare turbo-codului clasic. Construcția codorului se bazează pe registrul de deplasare cu reacție (linear feedback shift register LFSR) cu R intrări. În această lucrare, am propus o metodă de codare alternativă, care este o generalizare a codorului Fibonacci, [HAS71], [GOK02]. În paragraful 5.2 se arată că pentru $R > 1$, cele două metode conduc la seturi de codoare diferite, notate H și respectiv G. Am arătat, pe de o parte, că orice codor în forma H are un unic echivalent în forma G, cu toate că setul codoarelor G este mai puțin numeros decât setul codoarelor H, pentru orice valoare a lui R , cu $R > 1$. Pe de altă parte, am arătat că orice codor G este unic. Aceste rezultate simplifică în mare măsură proiectarea MBTC-lor, care în mod uzual se bazează pe o căutare exhaustivă pe mulțimea codurilor constituente și al interleaverelor.

MBTC-le, de asemenea numite ca și turbo-coduri non-binare [BEJ99], [GHA04] oferă câteva avantaje în comparație cu turbo-codurile binare precum o convergență mai rapidă și un efect de „error floor” mai scăzut. Mai multe detalii se pot găsi în [BJDK01]. Proiectarea TC-lor necesită, în general, o căutare exhaustivă pe mulțimea codurilor potențiale [BGM98], [RGT05] și a interleaverelor [SUT05]. În cazul turbo-codurilor cu intrări multiple, binare sau non-binare (MBTC [DOB93] și respectiv MNBTC [BH30]), această căutare devine o problemă majoră pentru optimizarea lor deoarece complexitatea căutării computaționale crește exponențial cu numărul de intrări R . Astfel, restrângerea căutării la un set limitat de $M(N)$ BTC-le „bune” este esențială.

5.1.2 Prima formă canonică a codoarelor convoluționale: Configurația H

Figura 5.1 prezintă structura unui codor convoluțional recursiv și sistematic cu intrări multiple ce a fost utilizat în [DOB93] pentru fiecare dintre codurile componente ale turbo-codului. Codorul este construit pe baza registrului de deplasare cu reacție (LFSR) cu R intrări. Acest codor nu este decompozabil, în general, în R codoare uni-intrare. În prezenta lucrare, această configurație este numită ca forma canonică de tip „H”.

Codorul are R intrări u_1, u_2, \dots, u_R și $R+1$ ieșiri corespunzătoare celor R intrări și unui bit redundant u_0 numit, de asemenea, și c . Starea curentă a codorului este dată de ieșirile celor M celule ale registrului de întârziere S_1, S_2, \dots, S_M . Intrările $u_r, r=1 \dots R$ sunt conectate fizic la sumatorul m dacă $h_{r,m}=1$; $s^n = [s_M^n \dots s_2^n s_1^n]^T$ și $u^n = [u_R^n u_{R-1}^n \dots u_1^n]^T$ descriu starea codorului la momentul n și, respectiv, vectorul

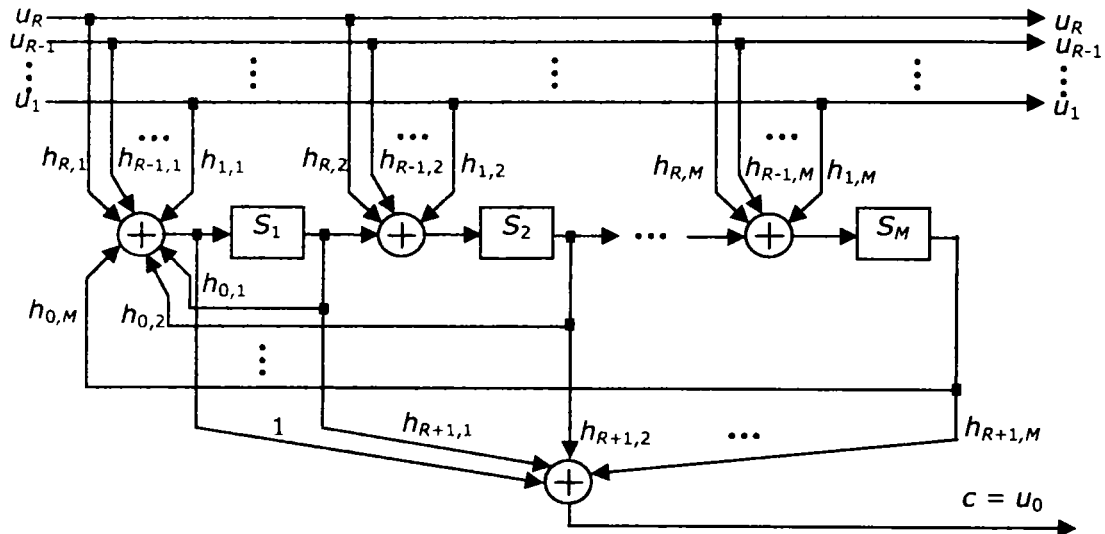


Figura 5.1 Structura generală a unui codor convoluțional multi-intrare, recursiv și sistematic, de rată $R/(R+1)$: configurația H.

de intrare de dimensiune $R \times 1$, unde $[x]^T$ denotă transpusa vectorului x . Relațiile intrare/stare curentă și ieșire/stare curentă ale codorului la momentul de timp (tact) n pot fi exprimate în forma compactă:

$$s^{n+1} = H_0 \cdot u^n + T \cdot s^n \quad (5.1)$$

$$c^n = H_E \cdot s^n + W \cdot s^{n+1} \quad (5.2)$$

unde matricea generatoare H_0 și matricea de tranziție T sunt definite prin relațiile:

$$H_0 = \begin{bmatrix} h_{R,M} & \dots & h_{1,M} \\ \dots & \dots & \dots \\ h_{R,2} & \dots & h_{1,2} \\ h_{R,1} & \dots & h_{1,1} \end{bmatrix} \quad (5.3)$$

și

$$T = \begin{bmatrix} 0_{(M-1) \times 1} & I_{M-1} \\ & H_F \end{bmatrix} \quad (5.4)$$

unde

$$H_F = [h_{0,M} \dots h_{0,2} h_{0,1}]. \quad (5.5)$$

Pentru a avea un cod decodabil [JOZ99], se presupune că H_0 este de rang maxim. În (5.2), vectorul W este egal cu $[0 \ 0 \ \dots \ 0 \ 1]_{1 \times M}$, și:

$$H_E = [h_{R+1,M} \dots h_{R+1,2} h_{R+1,1}]. \quad (5.6)$$

Pentru a compara această formă canonică cu structura prezentată în secțiunea următoare, se determină funcția de transfer a codorului, în formă matricială. În

acest scop definim transformata „ D ” a unei secvențe w_n ca și $W(D) = \sum_{n=-\infty}^{\infty} w_n \cdot D^n$.

În ipoteza că matricea $I_M + D \cdot T$ este inversabilă (Anexa C), adică $\det(I_M + D \cdot T) = h_0(D) \neq 0$ cu $h_r(D) = \sum_{m=1}^M h_{r,m} \cdot D^{m-1}$, funcția de transfer corespunzătoare secvenței redundante a codorului, $C(D)$, și funcția de transfer matricială globală a codului $M_h(D)$ sunt:

$$C(D) = M_h(D) \cdot U(D) \quad (5.7)$$

$$M_h(D) = D \cdot H_E \cdot (I_M + D \cdot T)^{-1} \cdot H_0 + W \cdot (I_M + D \cdot T)^{-1} \cdot H_0. \quad (5.8)$$

Prin câteva calcule algebrice (prezentate în Anexa C) se poate arăta că:

$$(I_M + D \cdot T)^{-1} = \frac{1}{h_0(D)} \cdot P_M^T(D) \cdot H_F \cdot \Delta_M(D) + D^{-1} \cdot \Delta_M(D) \quad (5.9)$$

unde $P_M(D) = [D^{M-1} \dots D \ 1]$ iar $\Delta_M(D)$ este o matrice Toeplitz cu prima linie $[D \ D^2 \ \dots \ D^M \ 1 \ D^M]$ și prima coloană $[D \ 0 \ \dots \ 0 \ 0]^T$. În acest fel, funcția de transfer matricială $M_h(D)$ poate fi simplificată la forma:

$$M_h(D) = \left(\frac{h_{R+1}(D)}{h_0(D)} \cdot H_F + H_E \right) \cdot \Delta_M(D) \cdot H_0 + [h_{R,1} \ \dots \ h_{2,1} \ h_{1,1}]. \quad (5.10)$$

În secțiunea următoare vom introduce o altă formă canonică pentru codoarele multi-intrare, bazată pe reprezentarea Fibonacci (observator).

5.1.3 A doua formă canonică a codoarelor convoluționale: Configurația G

Un codor convoluțional recursiv și sistematic cu intrări multiple nu este, în general, decompozabil în R codoare uni-intrare, adică nu există câte un codor echivalent cu un registru cu reacție pentru fiecare intrare. Cu toate acestea, în această secțiune vom prezenta o structură realizabilă ce conține un registru cu reacție pentru unica ieșire c așa cum se arată în Figura 5.2. Vom numi această schemă ca și forma canonică de tip „G”. (această formă canonică este de asemenea cunoscută în literatură, ca și forma canonică „observator” [JOZ99].)

Fie $s^n = [s_{M-1}^n \ s_{M-2}^n \ \dots \ s_0^n]^T$ vectorul stare a codorului de la momentul n și $u^n = [u_R^n \ u_{R-1}^n \ \dots \ u_1^n]^T$ vectorul intrare de dimensiune $R \times 1$. Relațiile intrare/stare curentă și ieșire/stare curentă ale codorului, la momentul n pot fi exprimate în forma compactă:

$$s^{n+1} = G_T \cdot u^n + T^T \cdot s^n \quad (5.11)$$

$$c^n = G_L \cdot u^n + W \cdot s^n \quad (5.12)$$

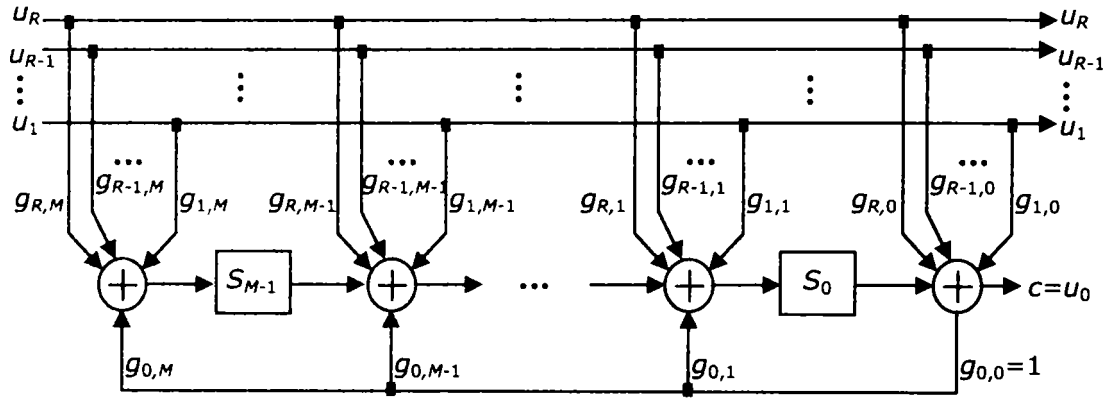


Figura 5.2 Structura generală a unui codor convoluțional multi-intrare, recursiv și sistematic, de rată $R/(R+1)$: configurația G

unde G_T și T^T sunt definite prin relațiile:

$$G_T = G_0 + G_F \cdot G_L \quad (5.13)$$

$$T^T = \begin{bmatrix} 0_{1 \times M-1} & G_F \\ I_{M-1} & \end{bmatrix}, \quad (5.14)$$

iar G_L și W sunt egale cu $[g_{R,0} \ g_{R-1,0} \ \dots \ g_{1,0}]$ și, respectiv, $[0 \ 0 \ \dots \ 0 \ 1]_{1 \times M}$. Pentru a avea un cod decodabil, vom presupune că G_T este de rang maxim [JOZ99]. G_0 este definită ca:

$$G_0 = \begin{bmatrix} g_{R,M} & g_{R-1,M} & \dots & g_{1,M} \\ g_{R,M-1} & g_{R-1,M-1} & \dots & g_{1,M-1} \\ \dots & \dots & \dots & \dots \\ g_{R,1} & g_{R-1,1} & \dots & g_{1,1} \end{bmatrix} \quad (5.15)$$

iar $G_F = [g_{0,M} \ g_{0,M-1} \ \dots \ g_{0,1}]^T$.

După câteva calcule algebrice și definind $g_r(D) = \sum_{m=0}^M g_{r,m} \cdot D^m$, se poate arăta

că transformata „D”, $C(D)$, a secvenței redundante și funcția de transfer matricială, $M_g(D)$, a codorului sunt date prin relațiile:

$$C(D) = G_L \cdot U(D) + W \cdot S(D) \quad (5.16)$$

$$M_g(D) = \frac{1}{g_0(D)} \cdot (D \cdot P_M(D) \cdot G_0 + G_L). \quad (5.17)$$

După deducerea funcțiilor de transfer matriciale $M_h(D)$ și $M_g(D)$ pentru ambele forme canonice H și G în (5.10) și, respectiv, (5.17), în paragraful următor vom investiga relațiile de echivalență dintre formele canonice H și G.

5.2 Echivalența formelor canonice H și G

În primul rând vom defini relația de echivalență dintre două forme canonice H și G:

Definiția 5.1: Formele canonice H și G sunt echivalente dacă pentru orice secvență de date de intrare $U(D)$, ambele forme canonice generează aceeași ieșire $C(D)$, adică:

$$M_g(D) = M_h(D) \quad (5.18)$$

unde $M_h(D)$ și $M_g(D)$ sunt definite prin (5.10) și, respectiv (5.17).

5.2.1 Cazul uni-binar clasic ($R=1$ și $g_{1,0}=1$)

Vom investiga pentru început echivalența dintre cele două metode în cazul unei singure intrări și când $g_{1,0}=1$ ca și în cazul clasic [BER97].

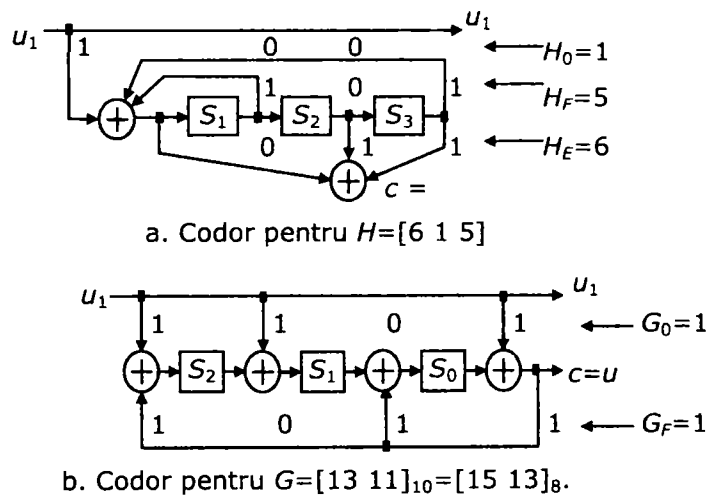


Figura 5.3 Codul $15/13_8$: formele echivalente H și G pentru codor

Teorema 5.1 ([JOW93] p.1220): În cazul binar clasic al unei singure intrări ($R=1$ și $g_{1,0}=1$), ambele configurații sunt echivalente.

Demonstrație:

Pentru $R=1$ avem că $H_0=W^T$. Prin multiplicarea ambelor membri ai ecuației (5.10) prin $h_0(D)$, avem că:

$$\begin{aligned} h_0(D) \cdot M_h(D) &= (h_2(D) \cdot H_F + h_0(D) \cdot H_E) \cdot D \cdot P_M^T(D) + h_0(D) \\ &= h_2(D) \cdot (h_0(D) + 1) + h_0(D) \cdot (h_2(D) + 1) + h_0(D) \\ &= h_2(D). \end{aligned}$$

Deoarece G_0 este un vector coloană și G_L este egal cu 1 în cazul unibinar, Ecuația (5.17) poate fi scrisă ca: $g_0(D) \cdot M_g(D) = g_{1,M} \cdot D^M + \dots + g_{1,2} \cdot D^2 + g_{1,1} \cdot D + 1 = g_1(D)$.

Astfel, dacă $h_2(D) = g_1(D)$ și $h_0(D) = g_0(D)$ ambele configurații H și G sunt echivalente una câte una în cazul uni-binar clasic.

Vom ilustra acest rezultat prin următorul exemplu. Presupunem codul recursiv și sistematic având polinoamele de ieșire $15_8=1+D^2+D^3$ și, respectiv de reacție, $13_8=1+D+D^3$. Definind matricele generatoare complete H și G ca:

$$H = [H_E^T \quad H_0 \quad H_F^T] = [h_{R+1} \quad h_R \quad \dots \quad h_1 \quad h_0]_{10} \quad (5.19)$$

și

$$G = \begin{bmatrix} G_0 & G_F \\ G_L & 1 \end{bmatrix} = [g_R \quad \dots \quad g_1 \quad g_0]_{10}, \quad (5.20)$$

H și G sunt egale cu [6 1 5] și [13 11] în acest exemplu. Cele două forme canonice H și G ale codorului sunt prezentate în Figura 5.3.a și, respectiv, 5.3.b.

Deoarece avem o echivalență strictă între formele canonice H și G în cazul uni-binar clasic, unde $R=1$ și $g_{10}=1$, optimizarea turbo-codurilor poate fi făcută utilizând oricare dintre formele G sau H. Afirmatia nu mai este adevărată în cazul multi-binar.

5.2.2 Cazul intrărilor multiple ($R>1$)

Vom începe făcând următoarea remarcă. Fie G și H seturile codoarelor generate de cele două forme canonice G și, respectiv, H. Dimensiunile matricilor H și G definite prin relațiile (5.19) și (5.20) sunt $M \times (R+2)$ și, respectiv, $(M+1) \times (R+1)$. Presupunând că $g_{0,0}=1$ și $M \geq R$, matricea H are mai multe elemente decât G. Rezultă că setul H conține mai multe elemente decât setul G, astfel că, în general, formele canonice H și G nu vor fi echivalente una câte una în cazul intrărilor multiple.

Condițiile pentru care cele două forme canonice sunt echivalente sunt date se următoarele două teoreme.

Teorema 5.2: Pentru orice matrice generatoare H având forma canonică H, există o unică matrice generatoare G echivalentă, având forma canonică G, care este soluție a următorului sistem:

- (α) $g_{0,0}=1$,
- (β) $g_{r,0} = h_{r,1}$ pentru orice $r = 1, \dots, R$,
- (γ) $g_{0,m} = h_{0,m}$ pentru orice $m = 1, \dots, M$,
- (δ) $D \cdot P_M(D) \cdot G_0 = (h_{R+1}(D) \cdot H_F + h_0(D) \cdot H_E) \cdot \Delta_M(D) \cdot H_0 + (h_0(D) + 1) \cdot [h_{R,1} \dots h_{2,1} \quad h_{1,1}]$.

Demonstrație:

Ec. (α) impune ca structura să fie recursivă. După câteva calcule algebrice, este ușor de văzut că relațiile (β), (γ) și (δ) sunt echivalente cu (5.10), (5.17) și (5.18). Pentru ca demonstrația să fie completă, mai este necesar să arătăm că (δ) are ca soluție unică pe G_0 .

Lema 5.1: Ecuația (δ) este echivalentă cu ecuația: $G_T = A \cdot H_0$ unde matricea A de dimensiune $M \times M$ este determinată recursiv după cum urmează:

- Inițializare: $H_{EF} = [H_E \quad 1]^T \cdot H_F + [H_F \quad 1]^T \cdot H_E$ și $A(1, \cdot) = H_{EF}(1, \cdot)$;
- Iterația $m = 2, 3, \dots, M$: $A(m, \cdot) = H_{EF}(i, \cdot) + A(m-1, \cdot) \cdot I_M(2)$;

unde $I_M(k)$ este matricea unitate de dimensiune $M \times M$ deplasată cu $k-1$ poziții spre dreapta. $A(m,:)$ reprezintă vectorul linie al elementelor din linia m a matricii A , în notație MATLAB, [GHF95].

Demonstrație:

Prin definiție avem: $h_{R+1}(D) \cdot H_F + h_0(D) \cdot H_E = [D^M \ D^{M-1} \ \dots \ D \ 1] \{ [H_E \ 1]^T \cdot H_F + [H_F \ 1]^T \cdot H_E \}$, și $\Delta_M(D) = I_M(1) \cdot D + I_M(2) \cdot D^2 + \dots + I_M(M) \cdot D^M$. Utilizând (β) , (γ) și (5.15), putem scrie:

$$\sum_{m=1}^M D^m \cdot G_T(m,:) = \sum_{m=1}^{M+1} \sum_{k=1}^M D^{m+k-1} \cdot H_{EF}(m,:) \cdot I_M(k) \cdot H_0 \quad (5.21)$$

sau, echivalent, termen cu termen:

$$\begin{aligned} G_T(m,:) &= \sum_{k=1}^m H_{EF}(m,:) \cdot I_M(m-k+1) \cdot H_0 \\ &= \left[H_{EF}(m,:) + \sum_{k=1}^{m-1} H_{EF}(m,:) \cdot I_M(m-k) \cdot I_M(2) \right] \cdot H_0. \end{aligned} \quad (5.22)$$

Definim matricea A prin: $A(m,:) = \sum_{k=1}^m H_{EF}(m,:) \cdot I_M(m-k+1)$. Utilizând (5.22), linia m din A poate fi exprimată ca: $A(m,:) = H_{EF}(m,:) + A(m-1,:) \cdot I_M(2)$ iar $G_T(m,:)$ este egal cu $A(m,:) \cdot H_0$, $i = 1, 2, \dots, M$ sau echivalent: $G_T = A \cdot H_0$.

În acord cu Lema 5.1, pentru orice codor definit prin matricea sa H_0 , există o matrice generatoare unică G_T și astfel un codor echivalent de forma canonică G unic, fapt ce completează demonstrația teoremei 5.2.

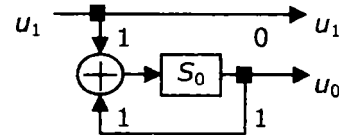


Figura 5.4. Codorul pentru $G=[2 \ 3]$

Teorema 5.2 arată că pentru orice matrice H există o unică matrice echivalentă G (dar nu și pentru orice G există o matrice echivalentă H). Considerăm următorul exemplu: presupunem $M=R=1$ și matricea generatoare $G = [2 \ 3]$. Codorul corespunzător este prezentat în Figura 5.4. Acest exemplu nu aparține cazului clasic uni-binar deoarece $g_{10}=0 \neq 1$. Avem succesiv: $G_R = [1] = H_F$, $G_L = [0] = [h_{11}] = H_0$, $H = [H_E^T \ H_0 \ H_F^T] = [x \ 0 \ 1]$. Este ușor de arătat că relația (δ) devine $[1] = [1+x] \cdot [0]$ ce nu poate fi satisfăcută pentru nici o valoare $x \in \{0, 1\}$. Astfel, codorul prezentat în Figura 5.4 nu poate fi construit în forma canonică H .

Teorema 5.3: Pentru orice matrice generatoare din G , există i) nici una, ii) una sau iii) mai multe matrici generatoare echivalente din H .

Demonstrație:

O matrice generatoare H din H care are un echivalent în G verifică sistemul de ecuații: (β) $[h_{R,1} \ h_{R-1,1} \ \dots \ h_{1,1}] = G_L$;

(γ) $H_F = G_R$;

(δ) $A \cdot H_0 = G_T$ (Lemma 1).

Pentru a afla vectorul H_E și matricea H_0 ce verifică (δ) se poate utiliza o căutare computațională exhaustivă peste cele $2^{(m-1)r+m}$ perechi posibile $\{H_E, H_0\}$. Astfel, ecuația (δ) are i) niciuna, ii) una sau iii) mai multe soluții:

i) Am văzut prin exemplul anterior că o matrice din G poate să nu aibă un echivalent în H . În secțiunea următoare vom arăta și alte exemple.

ii) Există soluție unică în cazul uni-binar clasic așa cum arată Teorema 5.2. Însă, și pentru $R > 1$, există cazuri de soluție unică H_0 a ecuației (δ) pentru anumite matrici G_T .

iii) În mod cert avem: $|H| \geq |G|$, unde $|\cdot|$ reprezintă cardinalul mulțimii. Pe de altă parte, din Teorema 5.2, știm că orice matrice generatoare din H are o matrice echivalentă unică în G . Rezultă că există cel puțin două codoare din H care au aceeași matrice echivalentă în G .

În acord cu Teorema 5.3, sistemul de ecuații (α), (β), (γ) și (δ) stabilesc o funcție de echivalență $\xi: H \rightarrow G$, care nu este nici injectivă nici surjectivă.

Teorema 5.3 are un impact foarte mare asupra proiectării TC-lor. Astfel, o căutare a unui codor bun în G în locul setului H nu este doar o căutare mai rapidă, în acord cu condiția (iii), dar există posibilitatea găsirii unei soluții care nu există în H așa cum indică condiția (i). Mai mult, vom arăta prin următoarea teoremă că nu există două codoare distincte în G care să fie echivalente unul cu celălalt.

Teorema 5.4: $\forall \{G^1, G^2 \neq G^1\} \in G$, G^1 și G^2 nu pot fi echivalente între ele, adică două matrici distincte din G nu pot fi echivalente.

Demonstrație:

Presupunem că ar exista două matrici G^1 și G^2 din G ce sunt echivalente. Din relațiile (5.17) și (5.18) rezultă că $g_0^1(D) = g_0^2(D)$, $G_R^1 = G_R^2$ și $D \cdot P_m(D) \cdot G_0^1 + G_L^1 = D \cdot P_m(D) \cdot G_0^2 + G_L^2$. Ultima relație este verificată dacă și numai dacă: $G_0^1 \equiv G_0^2$ și $G_L^1 \equiv G_L^2$ adică $G^1 = G^2$.

5.2.3 Cazul codurilor convoluționale multi-non-binare

Este ușor de văzut că toate calculele anterioare sunt valide și pentru cazul în care intrările codorului sunt non-binare, caz tratat în capitolul următor. Astfel, toate rezultatele pot fi generalizate pentru cazul în care $(u_r, s_m, h_{r,m}, g_{r,m})$ sunt elemente din $GF(2^Q)$, $Q > 1$. Aceste rezultate pot fi folosite direct pentru proiectarea MNBTC [BH30]. În acest caz, fiecare conexiune corespunde la o magistrală de Q biți; coeficienții h_{ij} și g_{ij} corespund unor circuite de multiplicare peste $GF(2^Q)$; registrele de întârziere înmagazinează caractere (pachete) de Q biți, iar sumatoarele sumează caractere sau simboluri în $GF(2^Q)$.

5.2.4 Concluzii

În această secțiune vom evalua numeric importanța proiectării MBTC-lor prin căutare asupra codoarelor candidați, generați prin forma canonică G comparativ cu o căutare asupra celor generați prin forma canonică H:

- Avem că $|H| \geq |G|$. Astfel, căutarea $M(N)BTC$ -lor în G este mai rapidă decât în H. Însă, pentru valori practice ale lui M și R , adică pentru coduri constituente cu memorie de 3 sau 4, care este câștigul efectiv?
- În Teorema 5.3 am arătat că există codoare din G care nu au echivalent în H. Însă, pentru valorile practice ale lui M și R , sunt aceste coduri bune pentru construcția unor MBTC-uri bune?

A. Cardinalul subseturilor G, H și $\xi(H)$

Fără a pierde din generalitate, vom limita căutarea asupra codurilor recursive și decodabile din G și H, adică G_R și H_F nu sunt vectori nuli. În plus, inversând intrările între ele codurile rezultate sunt echivalente. Astfel vom presupune că primele R elemente ale matricei G sunt sortate în ordine descrescătoare. Fie cele două submulțimi rezultate G_0 și H_0 . În plus,

Tabelul 5.1 Cardinalul seturilor G_0 , H_0 și $\xi(H_0)$

M	R	$ G_0 $	$ H_0 $	$ \xi(H_0) $
2	1	16	21	15
	2	36	18	18
3	1	80	233	78
	2	552	663	411
	3	1568	784	756
4	1	352	2169	348
	2	5712	15189	4887
	3	48832	56588	32704

definim subsetul $\xi(H_0)$ a lui G_0 conținând elementele ce nu au echivalent în H_0 . În Tabelul 5.1 am comparat cardinalul seturilor G_0 , H_0 și $\xi(H_0)$ pentru valori practice ale lui M și R . Diferența $|G_0| - |\xi(H_0)|$ crește exponențial cu R . Raportul $|H_0| / |\xi(H_0)|$ crește exponențial cu $M-R$. Cu cât memoria este mai mare și/sau rata de codare este mai mică, cu atât este mai indicat să se caute asupra formei canonice G în detrimentul formei canonice H.

Dacă polinomul de reacție $g_0(D) = h_0(D)$ este un polinom primitiv cu $g_{0,M} = h_{0,M} \neq 0$, care este cazul practic al codurilor „bune”, orice matrice din G are 2^{M-R} matrici echivalente în H. Așadar, chiar dacă căutarea este limitată la matricile generatoare cu polinom de reacție primitiv, câștigul căutării codurilor constituente

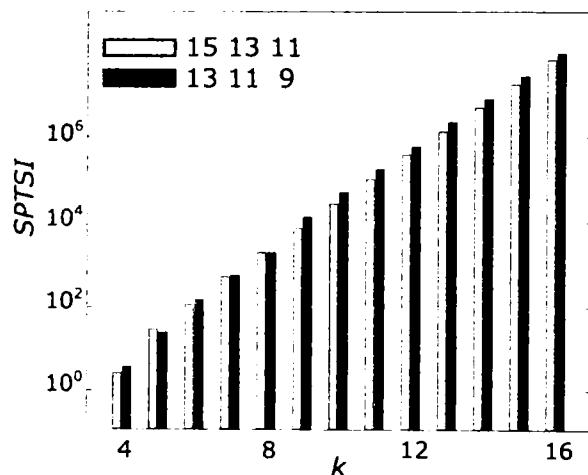
Tabelul 5.2 Matricile generatoare echivalente pentru cele două coduri de rată 2/3 propuse în [DOB05]

M	R	G	H			
			H_1	H_2	H_3	H_4
3	2	15 13 11	2 5 7 5	6 7 1 5		
4	2	21 23 25	8 3 7 12	10 1 13 12	11 11 1 12	14 9 11 12

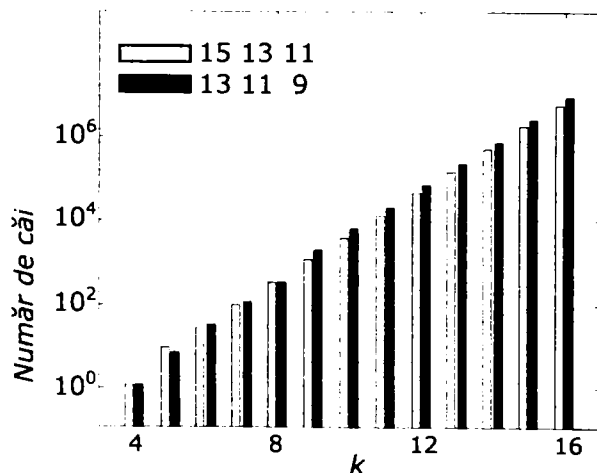
pentru MBTC din G în comparație cu H este de același ordin, adică complexitatea computațională a căutării exhaustive descrește exponențial cu $M-R$. Spre exemplificare, am determinat în Tabelul 5.2 matricile echivalente pentru cele două coduri de rată $2/3$ propuse în [DOB05] de memorie $M=3$ și $M=4$ cu matricile generatoare $H=[6\ 7\ 1\ 5]$ și, respectiv, $H=[11\ 11\ 1\ 12]$.

B. Distanța minimă și ponderea căilor

Cu toate că scopul nostru inițial a fost simplificarea proiectării MBTC-ilor, forma canonică G poate fi, de asemenea, utilizată pentru codurile convoluționale cu intrări multiple (MBCC). Am găsit, printr-o căutare exhaustivă, cele mai bune coduri de memorie 3 [BH4].



a) suma ponderilor tuturor secvențelor de intrare a căror secvențe de ieșire corespunzătoare au ponderi egale cu k , funcție de ponderea k



b) numărul căilor de pondere k , funcție de ponderea k

Figura 5.5 Comparatie între spectrele distanțelor ale MBCC-lui propus în [DOB05] (cu $G=[15\ 13\ 11]$) și cel mai bun MBCC găsit printr-o căutare exhaustivă (cu $G=[13\ 11\ 9]$)

Deși cel mai bun cod convoluțional duo-binar nu este neapărat necesar cel mai bun cod constituent pentru MBTC, el este utilizat în sistemele de transmisie ce necesită o complexitate computațională scăzută sau o întârziere scăzută.

Criteriile principale pentru proiectarea MBCC sunt:

- distanța minimă, și
- spectrul ponderilor de cod [JOZ99].

Alt criteriu introdus în [JEZ03]: este suma ponderilor tuturor secvențelor de intrare (*SPTSI*) a căror secvențe de ieșire corespunzătoare au ponderi egale cu k . Și anume *SPTSI* dă o măsură a mediei numărului de erori ce se produc atunci când decodorul selectează o cale greșită din trellisul codului, cale care corespunde uneia dintre aceste secvențe de intrare. Cu cât *SPTSI* este mai mic cu atât codul este mai bun.

Surprinzător, cel mai bun cod convoluțional din G aparține mulțimii $G \setminus H$. (Subsetul $G \setminus H$ cuprinde elementele din G ce nu au nici un echivalent în H , adică nu are nici un cod echivalent în H). Matricea sa generatoare este egală cu: $G = [13 \ 11 \ 9]$. Distanța minimă de cod este aceeași (egală cu 4) cu a celui mai bun cod din H , propus în [DOB05], având matricea generatoare $G = [15 \ 13 \ 11]$, dar *SPTSI* și spectrul distanțelor de cod sunt semnificativ mai bune pentru primul, așa cum se poate vedea în Figura 5.5.

Este ușor de văzut că matricele ce aparțin mulțimii $G \setminus H$ nu au ca polinom de reacție $g_0(D)$ un polinom primitiv de grad maxim M . Matricile din $G \setminus H$ au ca polinom de reacție fie un polinom primitiv de grad mai mic decât M fie un polinom care nu este primitiv. Deoarece cele mai bune MBTC sunt bazate pe codoare cu polinoame primitive în reacție, nu este necesar să se considere codurile ce aparțin mulțimii $G \setminus H$ în proiectarea MBTC. Forma canonică G rămâne însă interesantă în proiectarea MBTC grație proprietăților de echivalență prezentate în secțiunea anterioară.

5.3 Construcția Turbo-Codurilor Multi-Binare

Structural un MBTC nu diferă de un BTC, singura diferență o constituie legăturile multiple între codoare (via interleaver) precum și între decodoare. În plus, pentru decodarea per caracter a MBTC-lor, care le conferă de fapt superioritate față de BTC-uri, algoritmul MAP suferă o modificare care constă în înlocuirea rapoartelor de plauzibilitate logaritmice (LLR-lor) cu probabilitățile aposteriori (APP –APosteriori Probability). În acest paragraf sunt descrise toate aceste modificări necesare pentru construcția MBTC-lor.

5.3.1 Structura unui TCMB

Considerând că turbo-codul prezentat în Figura 4.1 este multi-binar, așa cum am menționat, trebuie să înlocuim LLR-le prin APP-uri. Cu această schimbare structura unui MBTC arată ca și în Figura 5.6. În ceea ce privește notațiile trebuie să adăugăm un nou indice pentru numărul intrării. Fie astfel R numărul de intrări și r indexul acestora. În consecință vom avea: $u = (u^1, u^2, \dots, u^N)$ blocul simbolurilor de informație, cu $u^n = (u_1^n, u_2^n, \dots, u_R^n)^T$; $x = (x^1, x^2, \dots, x^N)$ secvența emisă, cu $x^n = (x_0^n, x_1^n, \dots, x_{R+1}^n)^T$ și $x_r^n = \pm 1$; $y = (y^1, y^2, \dots, y^N)$ secvența recepționată, cu $y^n = (y_0^n, y_1^n, \dots, y_{R+1}^n)^T$ și $y_r^n = x_r^n + w_r^n$, unde w_r^n este o variabilă aleatoare gaussiană cu

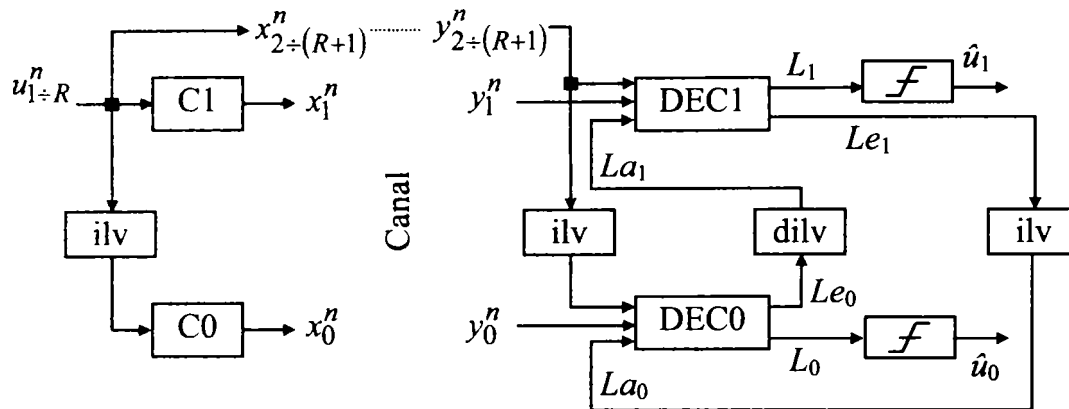


Figura 5.6 Structura unui MBTC

medie zero și dispersie σ^2 ; $\hat{u}_i = (\hat{u}_i^1, \hat{u}_i^2, \dots, \hat{u}_i^N)$, cu $\hat{u}_i^n = (\hat{u}_{i,1}^n, \hat{u}_{i,2}^n, \dots, \hat{u}_{i,R}^n)$ și $i = 1$ sau 0 , denotă estimatul lui u dat de DEC1 și respectiv DEC0. Pentru simplitate, vom considera că u^n și \hat{u}_i^n sunt întregi din mulțimea $J = \{0, 1, \dots, (2^R - 1)\}$. Astfel, în algoritmul de decodare MAP, $La_{i,t}^n(d)$, $Le_{i,t}^n(d)$ și $L_{i,t}^n(d)$ reprezintă probabilitățile a priori, extrinseci și a posteriori (APP) de la decodorul i (după iterația t), pentru ca simbolul original u^n să fie estimat la întregul d , adică probabilitățile indicate ca $\hat{u}_i^n = d \in J$. Notăm de asemenea că un bloc de informație conține N simboluri și $R \times N$ biți.

5.3.2 Diagrama de stare. Diagrama trellis

Pornind de la sistemul de ecuații (5.1) și (5.2), sau, alternativ, de la sistemul de ecuații (5.11) și (5.12), se poate construi diagrama de stare a codorului multi-binar, [BH14] [BH15]. Diagrama va conține 2^M noduri, corespunzătoare celor 2^M stări posibile. Din fiecare nod vor pleca 2^R ramuri, corespunzătoare celor 2^R posibili vectori de intrare. Astfel diagrama va conține 2^{M+R} ramuri sau tranziții. Fiecare tranziție este asociată cu durata unui simbol. Pe durata simbolului sunt preluați (de către codorul multi-binar) R biți de la intrare și sunt generați $R+1$ biți la ieșire. În Figura 5.7a se prezintă codorul duo-binar corespunzător matricii generatoare:

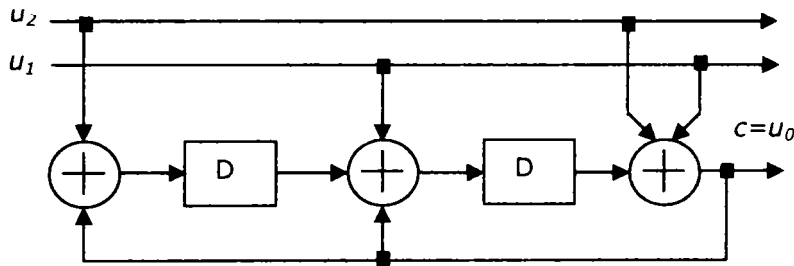
$$G = [5 \ 3 \ 7]_8 = [1 \ 0 \ 1; 0 \ 1 \ 1; 1 \ 1 \ 1]. \quad (5.23)$$

iar în Figura 5.7 b),c) sunt prezentate diagramele de stare și trellis corespunzătoare.

5.4 Închiderea trellisului MBC-ilor

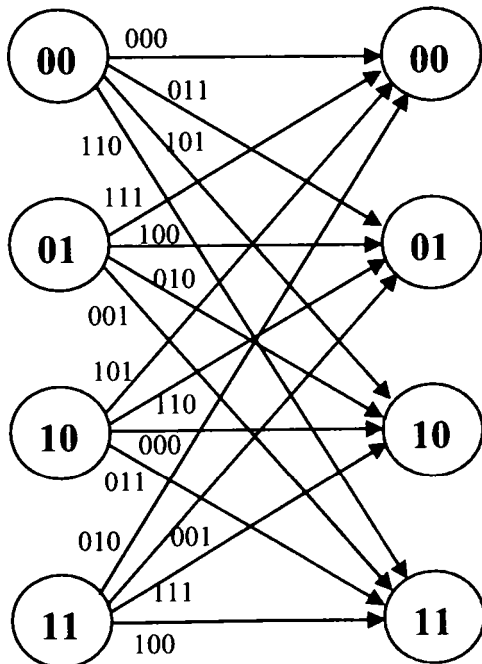
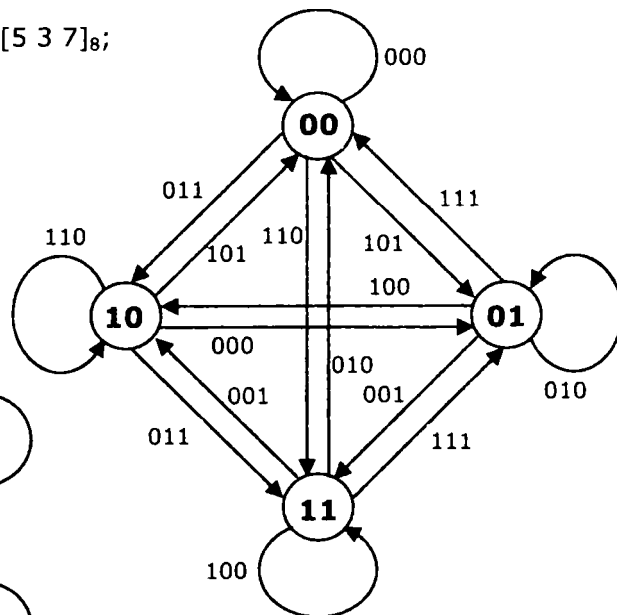
Asemenea turbo-codurilor clasice, cu o singură intrare, și turbo-codul multi-binar segmentează secvențele de date în blocuri, asupra cărora se efectuează codarea și decodarea, turbo-codul în sine devenind un cod bloc. Apare astfel necesitatea de a finaliza codarea printr-o închidere a trellisului. Aceasta presupune

inserarea, în secvențele de date, a unor biți redundanți, cu scopul de a forța în decodor, la finele codării, o stare anume. Cu excepția codurilor circulare, această stare va fi starea nulă. În acest paragraf sunt descrise atât metodele de închidere a trellisului codului convoluțional multi-binar, independent de postura sa de cod component al turbo-codului, cât și problematica închiderii trellisurilor codurilor componente turbo-codului.



a) codorul asociat matricii $G = [5 \ 3 \ 7]_8$;

b) diagrama de stare;



c) diagrama trellis;

Figura 5.7 Exemplu de cod duo-binar

5.4.1 Coduri multi-binare circulare

Asemănător codurilor uni-binare circulare, codurile multi-binare circulare execută codarea fiecărui bloc de date pornind și sfârșind cu aceeași stare a codorului (diferită, bineînțeles, de la un bloc de date la altul). Pentru a realiza acest lucru, codorul multi-binar execută în prealabil o pre-codare, fapt ce-i permite să afle starea din care trebuie să pornească codarea respectivului bloc. Această procedură este descrisă în continuare.

Utilizând recurența descrisă de Ecuația (5.11) se poate obține o relație între starea codorului multi-binar de la finele codării unui bloc de date de lungime N (prin N se înțelege lungimea fiecărei secvențe binare de intrare, din cele r aferente aceluiași bloc de date), stare notată s^N , și starea inițială a codorului, notată s^0 :

$$\begin{aligned} s^N &= G_T \cdot u^{N-1} + T^T \cdot s^{N-1} = G_T \cdot u^{N-1} + T^T \cdot (G_T \cdot u^{N-2} + T^T \cdot s^{N-2}) = \dots \\ &= \sum_{n=0}^{N-1} (T^T)^{N-n-1} \cdot G_T \cdot u^n + (T^T)^N \cdot s^0. \end{aligned} \quad (5.24)$$

În pre-codare se pornește cu starea nulă, $s^0 = 0_{M \times 1}$, obținându-se:

$$s^{pc} = \sum_{n=0}^{N-1} (T^T)^{N-n-1} \cdot G_T \cdot u^n. \quad (5.25)$$

Impunând ca $s^N = s^0$ și utilizând (5.25) rezultă condiția pentru cod circular:

$$s^0 = s^{pc} + (T^T)^N \cdot s^0. \quad (5.26)$$

Acest sistem de M ecuații cu M necunoscute (valorile vectorului s^0) este compatibil doar dacă:

$$T^N \neq I_M. \quad (5.27)$$

În ipoteza că reacția, $g_0(D)$ este constituită de un polinom primitiv, ce are perioada n_c , atunci:

$$T^{n_c} = I_M. \quad (5.28)$$

Astfel că îndeplinirea condiției (5.27) este realizată dacă N nu este un multiplu al lui n_c . Cu această prezumție, Ecuația (5.26) are soluția:

$$s^0 = (I_M + (T^T)^N)^{-1} \cdot s^{pc}. \quad (5.29)$$

În concluzie, starea s^0 ce trebuie setată în codor pentru asigurarea condiției de cod circular depinde de secvențele de date, prin s^{pc} dat de (5.25), de matricea generatoare a codului, prin T și s^{pc} și de lungimea de întretesere, N .

În continuare este prezentat, pentru un caz particular, un mod de calcul al soluției dată de (5.29). Astfel, în ipoteza:

$$\text{rest } \frac{N}{n_c} = 1, \quad (5.30)$$

rezultă că:

$$T^N = T. \quad (5.31)$$

În această ipoteză, matricea $(I_M + (T^T)^N)^{-1}$ este dată de relația (vezi Anexa C):

$$(I_M + (T^T)^N)^{-1} = (I_M + T^T)^{-1} = I_{\Delta M} + \Sigma_g \quad (5.32)$$

unde:

$$I_{\Delta M} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}_{M \times M}, \quad (5.33)$$

$$\Sigma_g \stackrel{\text{def}}{=} \begin{bmatrix} \sum_{m=M}^M g_{0,m} & \sum_{m=M}^M g_{0,m} & \sum_{m=M}^M g_{0,m} \\ \sum_{m=M-1}^M g_{0,m} & \sum_{m=M-1}^M g_{0,m} & \sum_{m=M-1}^M g_{0,m} \\ \dots & \dots & \dots \\ \sum_{m=1}^M g_{0,m} = 0 & \sum_{m=1}^M g_{0,m} = 0 & \dots \sum_{m=1}^M g_{0,m} = 0 \end{bmatrix}_{M \times M} \quad (5.34)$$

Obs. De notat că este posibil ca polinomul reacției $g_0(D)$ (sau corespunzător vectorul reacție G_R) să aibă gradul maxim $\text{grad}(g_0) < M$. În acest caz primele $M - \text{grad}(g_0)$ linii din matricea Σ_g sunt identic nule. Indiferent însă de gradul lui $g_0(D)$, matricea $I_{\Delta M} + \Sigma_g$ are rangul maxim M .

Exemplu 5.1

Fie codul convoluțional definit prin parametrii: $M=2$, $R=2$, $G=[5 \ 3 \ 7]_8$. Pentru a construi un cod circular trebuie calculată matricea $(I_M + T^T)^{-1} = I_{\Delta M} + \Sigma_g$. Pentru aceasta se calculează succesiv:

$$G = [5 \ 3 \ 7]_8 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \quad G_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad G_R = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad G_L = [1 \ 1];$$

$$G_T = G_0 + G_R \cdot G_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot [1 \ 1] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix};$$

$$T^T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}; \quad (T^T)^2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}; \quad (T^T)^3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2.$$

Sub restricția $N = 3 \cdot N_1 + 1$, rezultă:

$$I_M + T^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = (T^T)^2, \quad \text{de unde} \quad (I_M + T^T)^{-1} = T^T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Ultima relație mai poate fi găsită calculând pe rând:

$$I_{\Delta M} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}; \quad \Sigma_g = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

de unde:

$$I_{\Delta M} + \Sigma_g = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Codarea va decurge astfel:

1. se execută o precodare calculând matricea $s^{pc} = \begin{bmatrix} s_1^{pc} \\ s_0^{pc} \end{bmatrix}$, cu pornire din starea nulă.

2. se execută codarea cu pornire din starea $S_0 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_1^{pc} \\ s_0^{pc} \end{bmatrix} = \begin{bmatrix} s_0^{pc} \\ s_0^{pc} + s_1^{pc} \end{bmatrix}$.

5.4.2 Închiderea trellisului la starea nulă

Asemenea închiderii trellisului codului uni-binar, și în cazul multi-binar trellisul poate fi închis la starea nulă, $s^N = 0_{M \times 1}$, cu prețul înșării a M biți redundanți în blocul de date. Obiectivul acestui paragraf este de a determina acești biți. Presupunem că între M -numărul de stări a codorului (memoria codorului) și R -numărul de intrări, există relația:

$$R \cdot (K - 1) < M \leq R \cdot K. \quad (5.35)$$

Atunci, făcând apel la relația (5.24), starea finală $s^N = 0_{M \times 1}$ poate fi exprimată funcție de setul ultimilor K vectori de intrare:

$$s^N = 0_{M \times 1} = \sum_{k=0}^{K-1} (T^T)^k \cdot G_T \cdot u^{N-k-1} + (T^T)^K \cdot s^{N-K}. \quad (5.36)$$

Ecuția matricială (5.36) reprezintă un sistem de M ecuații cu $R \cdot K$ necunoscute care sunt elementele vectorilor u^{N-k-1} , cu k de la 0 la K . Această ecuație poate fi pusă sub forma:

$$\left[G_T \quad (T^T) \cdot G_T \quad (T^T)^2 \cdot G_T \quad \dots \quad (T^T)^K \cdot G_T \right]_{M \times (R \cdot K)} \cdot \begin{bmatrix} u^{N-1} \\ u^{N-2} \\ \dots \\ u^{N-K} \end{bmatrix}_{(R \cdot K) \times 1} = T^K \cdot s^{N-K}_{M \times 1}. \quad (5.37)$$

Dacă $M=R \cdot K$ atunci sistemul (5.37) are soluție unică, iar dacă $M < R \cdot K$, atunci rămân un număr de $R \cdot K - M$ elemente din u^{N-K} nedeterminate. În acest ultim caz, elementele nedeterminate pot fi biți de informație sau pot fi setați la o valoare predefinită (spre exemplu zero). Pentru o tratare unitară a celor două cazuri, notăm:

$$\stackrel{\text{def}}{P} = R \cdot K - M, \text{ cu } R - 1 \geq P \geq 0, \quad (5.38)$$

și:

$$\left[G_T \quad T^T \cdot G_T \quad (T^T)^2 \cdot G_T \quad \dots \quad (T^T)^K \cdot G_T \right]_{M \times (R \cdot K)} = [A_{M \times M} \quad B_{M \times P}] \quad (5.39)$$

$$\begin{bmatrix} u^{N-1} \\ u^{N-2} \\ \dots \\ u^{N-K} \end{bmatrix}_{(R \cdot K) \times 1} = \begin{bmatrix} u^{N-1} \\ u^{N-2} \\ \dots \\ u^{N-K+1} \\ u^{N-K} (P+1 \rightarrow N) \\ u^{N-K} (1 \rightarrow P) \end{bmatrix}_{(R \cdot K) \times 1} = \begin{bmatrix} u^F \\ u^{N-K} (1 \rightarrow P) \end{bmatrix} \quad (5.40)$$

unde matricile $B_{M \times P}$ și $u^{N-K}(1 \rightarrow P)$ sunt inexistente dacă $M = R \cdot K$. Cu aceste notații, cei M biți redundanți, conținuți în u^F , ce servesc închiderii trellisului, sunt dați prin relația:

$$u^F = A_{M \times M}^{-1} \cdot [T^K \cdot s^{N-K} + B_{M \times P} \cdot u^{N-K}(1 \rightarrow P)] = C_{M \times M} \cdot s^{N-K} + D_{M \times P} \cdot u^{N-K}(1 \rightarrow P). \quad (5.41)$$

Relația (5.41) spune că, cei M biți necesari închiderii trellisului codului convoluțional multi-intrare, se calculează funcție de: i) starea s^{N-K} la care a ajuns codorul după codarea celor $N-K$ seturi de R biți de intrare, unde K este dat de (5.35), și ii) valorile a $R \cdot K - M$ biți, ce pot fi atribuiți ca biți de informație sau pot fi setați la zero, caz în care ecuația (5.41) se scrie:

$$u^F = C_{M \times M} \cdot s^{N-K}. \quad (5.42)$$

Ultima relație se utilizează și dacă $M = R \cdot K$. Matricile $C_{M \times M}$ și $D_{M \times P}$ depind exclusiv de elementele matricii generatoare, fapt ce permite închiderea trellisului la starea nulă indiferent de lungimea de întrețesere.

Exemplu 5.2

Închiderea la zero a trellisului codului convoluțional având parametrii: $M=2$, $R=2$, $G=[5 \ 3 \ 7]_8$, necesită o relație de calcul a ultimilor 2 biți. Utilizând și rezultatele Exemplului 5.1 rezultă:

$$G_T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = G_T^{-1} = A_{2 \times 2}; \quad K=1; \quad T^T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix};$$

Dacă: $s^{N-1} = \begin{bmatrix} a \\ b \end{bmatrix}$, atunci:

$$U^F = U^{N-1} = C_{M \times M} \cdot S^{N-K} = G_T \cdot T^T \cdot S^{N-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a+b \\ b \end{bmatrix}.$$

Exemplu 5.3

Fie codul convoluțional definit prin parametrii: $M=3$, $R=2$, $G=[13 \ 7 \ 15]_8$. Urmărind procedura din Exemplul 5.1 rezultă:

$$G = [13 \ 7 \ 15]_8 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \quad G_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}; \quad G_F = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}; \quad G_L = [1 \ 1];$$

$$G_F \cdot G_L = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \cdot [1 \ 1] = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}; \quad G_T = G_0 + G_F \cdot G_L = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}; \quad T^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix};$$

Cod circular. Sub restricția $N = 7 \cdot N_1 + 1$, rezultă:

$$I_{\mathcal{M}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \quad \Sigma_g = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{de unde} \quad I_{\mathcal{M}} + \Sigma_g = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Codarea va decurge astfel:

1. se execută o precodare calculând matricea $s^{pc} = \begin{bmatrix} s_2^{pc} \\ s_1^{pc} \\ s_0^{pc} \end{bmatrix}$, cu pornire din

starea nulă.

2. se execută codarea cu pornire din starea

$$s^0 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_2^{pc} \\ s_1^{pc} \\ s_0^{pc} \end{bmatrix} = \begin{bmatrix} s_1^{pc} + s_0^{pc} \\ s_2^{pc} + s_1^{pc} \\ s_2^{pc} + s_1^{pc} + s_0^{pc} \end{bmatrix}.$$

Închidere la zero.

$$K = 2; \quad P=1; \quad [G_T \ T^T \cdot G_T] = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}; \quad A_{3 \times 3} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \quad B_{3 \times 1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};$$

$$C_{3 \times 3} = A_{3 \times 3}^{-1} \cdot (T^T) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix};$$

$$D_{3 \times 1} = A_{3 \times 3}^{-1} \cdot B_{3 \times 1} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix};$$

Dacă:

$$s^{N-2} = \begin{bmatrix} s_2^{N-1} \\ s_1^{N-1} \\ s_0^{N-1} \end{bmatrix}, \quad u^{N-2} = \begin{bmatrix} u_1^{N-2} \\ u_0^{N-2} \end{bmatrix}, \quad u^{N-1} = \begin{bmatrix} u_1^{N-1} \\ u_0^{N-1} \end{bmatrix},$$

atunci u_0^{N-2} se alocă unui bit de informație, și:

$$\begin{aligned} u^F &= \begin{bmatrix} u_1^{N-1} \\ u_0^{N-1} \\ u_1^{N-2} \end{bmatrix} = C_{3 \times 3} \cdot s^{N-2} + D_{3 \times 1} \cdot u_0^{N-2} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} s_2^{N-1} \\ s_1^{N-1} \\ s_0^{N-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \cdot u_0^{N-2} = \\ &= \begin{bmatrix} s_2^{N-1} + s_1^{N-1} + s_0^{N-1} + u_0^{N-2} \\ s_2^{N-1} + s_1^{N-1} \\ s_2^{N-1} + u_0^{N-2} \end{bmatrix}. \end{aligned}$$

5.5 Decodarea per caracter a MBTC-lor

Considerând că cele două decodare DEC1 și DEC0 implementează unul dintre algoritmi de decodare iterativi, (MAP, LogMAP, MaxLogMAP sau SOVA), decodarea MBTC-lor poate fi făcută în două moduri: per caracter sau per bit. Decodarea per bit este una și aceeași cu cea pentru BTC-uri. Astfel pentru fiecare bit din secvența de date se construiește logaritmul de raport de plauzibilitate (LLR), așa cum este descris în paragraful 4.5.

Decodarea per cuvânt presupune ca pentru fiecare caracter din secvența de date să se calculeze setul probabilităților aposteriori (APP). În acest caz, probabilitățile APP, extrinseci și a priori se referă, în fiecare moment n de timp, la un caracter din cele 2^R posibile. Cu notațiile din Figura 5.6, se pot scrie ecuațiile:

$$\begin{aligned} L_1^{n,i}(d) &= La_1^{n,i}(d) + \gamma_1^n + Le_1^{n,i}(d), \\ L_0^{n,i}(d) &= La_0^{n,i}(d) + \gamma_0^n + Le_0^{n,i}(d), \end{aligned} \quad (5.43)$$

unde $L_j^{n,i}(d)$ și $Le_j^{n,i}(d)$ sunt probabilitățile (a posteriori – APP, respectiv extrinsecă, calculate de decodorul j , $j = 0$ sau 1 , la iterația i și pasul n) ca u^n să ia valoarea $d \in J = \{0, 1 \dots 2^R - 1\}$. (Considerând o exprimare zecimală pentru toate cuvintele posibile u^n .) Probabilitățile a priori se calculează atunci:

$$\begin{aligned} La_1^{n,i}(d) &= \pi^{-1}(Le_0^{n,(i-1)}(d)), \\ La_0^{n,i}(d) &= \pi(Le_1^{n,(i-1)}(d)). \end{aligned} \quad (5.44)$$

Operațiile $\pi(\cdot)$ și $\pi^{-1}(\cdot)$ semnifică întrețeserea, respectiv de-întrețeserea (notate cu „ilv” și „dilv” în Figura 5.6). Y_1^n și Y_0^n sunt termeni calculați pe baza secvenței recepționate și a dispersiei zgomotului σ^2 :

$$Y_j^n = \frac{1}{\sigma^2} \cdot \sum_{r=2q=0}^{R+1Q-1} x_{r,q}^n \cdot Y_{r,q}^n + \frac{1}{\sigma^2} \cdot \sum_{q=0}^{Q-1} x_{j,q}^n \cdot Y_{j,q}^n, \quad (5.45)$$

cu $j = 0$ sau 1 . De remarcat că setul probabilităților extrinseci este numeric egal cu cel al ramurilor ce pleacă dintr-un nod al trellisului (fiecare ramură corespunde unei valori posibile a cuvântului de informație).

5.6 Criteriu de oprire a iterațiilor pentru MBTC-uri

5.6.1 Criteriul de oprire propus

În acest paragraf se prezintă un criteriu de oprire a iterațiilor, în două variante adaptate pentru algoritmi MAP și MaxLogMAP ai MBTC-ilor, criteriu ce este o adaptare a regulii mLLR pentru cazul MBTC-ilor, [BH30].

Considerând decodarea unui bloc convergent prin algoritmul de decodare MAP, pe măsură ce procesul iterativ avansează, probabilitățile APP corespunzătoare simbolurilor originale iau valori aproape de 1. Datorită acestui fapt, oricare alte probabilități APP, adică $L_j^{n,i}(d)$ cu $d \neq u^n$, vor lua valori apropiate de 0. Vom utiliza acest fapt pentru a construi un criteriu de oprire a iterațiilor:

Procesul iterativ de decodare este oprit la iterația i , dacă există, la orice moment de timp n , o probabilitate APP (dată de decodorul j) mai mare decât pragul μ impus:

$$\text{oprește iterațiile dacă } \forall n, 1 \leq n \leq N, \exists d \in J \text{ astfel încât } \mu < L_j^{n,i}(d) < 1. \quad (5.46)$$

Am numit acest criteriu the minimum likelihood APP (MLAPP). Însă, regula MLAPP nu poate fi utilizată în această formă pentru cazul algoritmului de decodare MaxLogMAP, datorită transferului valorilor APP-ilor în domeniul logaritmic.

Pentru cazul MaxLogMAP facem următorul raționament. Dacă $d = u^n$ atunci $L_j^{n,i}(d) \approx -\log(P\{\hat{u}_{nk} = u_k\} \approx 1)$ astfel că $L_j^{n,i}(d) \rightarrow 0$ când i crește. Dacă $d \neq u^n$, atunci $L_j^{n,i}(d) \approx -\log(\text{o probabilitate foarte mică})$ astfel că $L_j^{n,i}(d) \gg 0$. (În raționamentul anterior am presupus că în algoritmul de decodare MaxLogMAP s-a făcut o normalizare asupra valorilor APP corespunzătoare aceluiași moment de timp n , adică din toate valorile APP s-a scăzut cea mai mică valoare.)

În acest fel, criteriul de oprire MLAPP pentru cazul algoritmului MaxLogMAP este:

Procesul de decodare iterativ este oprit la iterația i dacă, la orice moment de timp n , toate probabilitățile APP (date de decodorul j), cu excepția celor egale cu zero, sunt mai mari decât pragul μ impus:

$$\text{oprește iterațiile dacă } \forall n, 1 \leq n \leq N, \text{ și } \forall L_j^{n,i}(d) > 0 \text{ rezultă că } L_j^{n,i}(d) > \mu. \quad (5.47)$$

Sau:

$$\text{oprește iterațiile dacă } \min_{d \in J^*} \{L_j^{n,i}(d)\} > \mu, \forall n, 1 \leq n \leq N. \quad (5.48)$$

unde $J^* = J \setminus \{d^*\}$ iar d^* este valoarea simbolului k pentru care $L_j^{n,i}(d^*) = 0$.

5.6.2 Rezultatul simulărilor

Am efectuat simulările utilizând turbo-codul duo-binar cu 8 stări și întrețeserea definite în [DOB05], cu o lungime a blocului de date $N = 752$ caractere = 1504 biți. Am presupus un canal AWGN și o modulație QPSK. Nu am utilizat cuantizare.

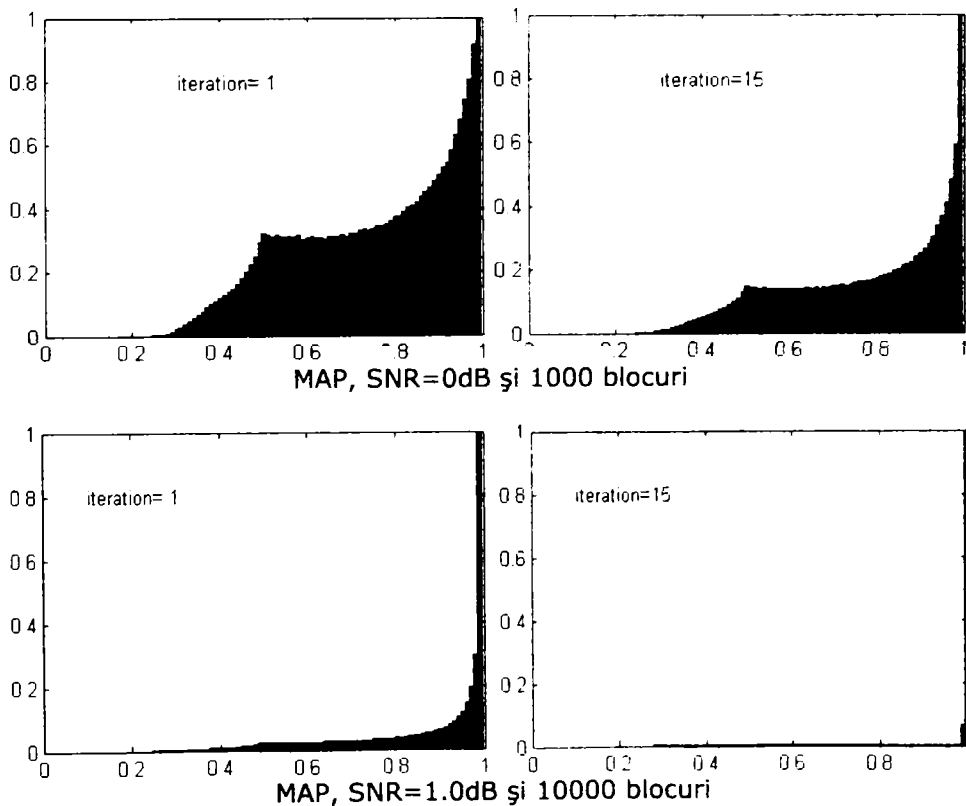


Figura 5.8 Histogramele APP normalizate pentru algoritmul de decodare MAP

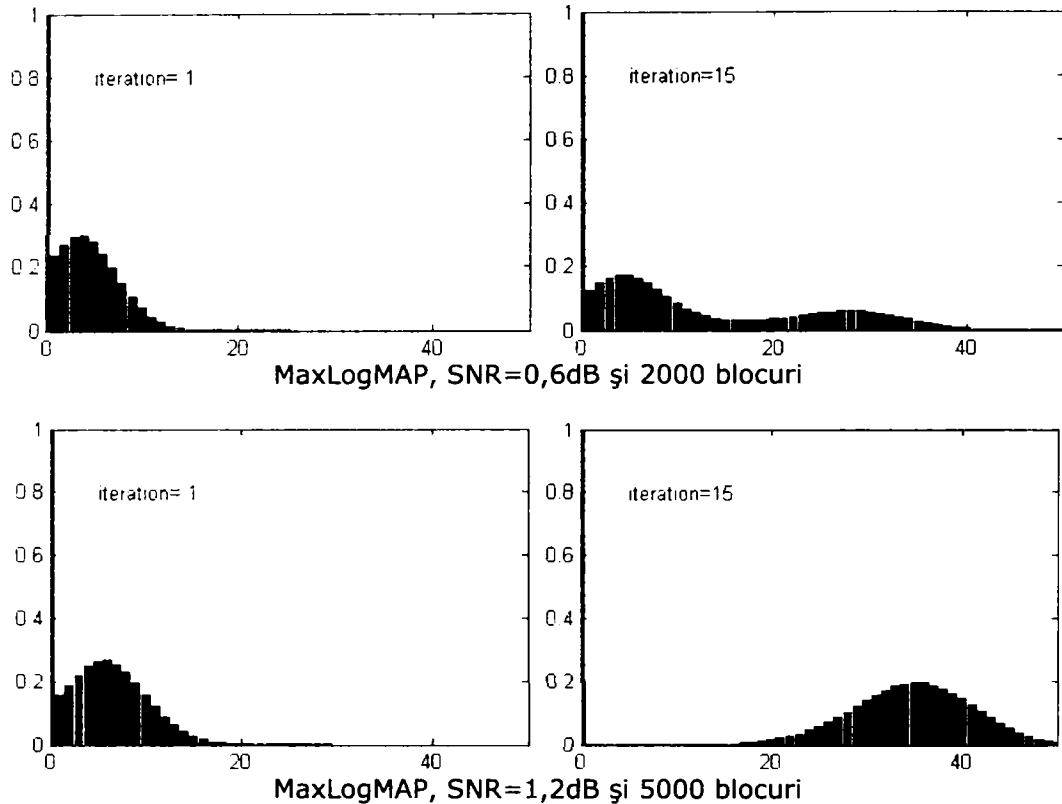


Figura 5.9 Histogramele APP normalizate pentru algoritmul de decodare MaxLogMAP

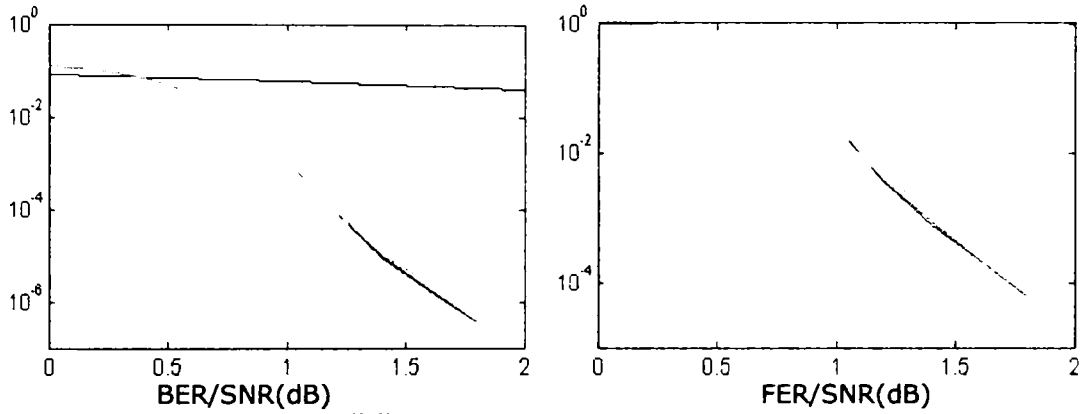
Diagramele din Figura 5.8 și Figura 5.9 prezintă histogramele APP normalizate pentru ambele cazuri MAP și MaxLogMAP, realizate efectuând 15 iterații fără stop. Normalizarea histogramelor presupune că toate valorile obținute au fost împărțite la cea mai mare, după fiecare iterație.

În cazul MAP am considerat doar APP-le pentru simbolurile maxim plauzibile. Simulările arată că aceste valori APP converg către 1 când SNR-ul și numărul iterației i cresc.

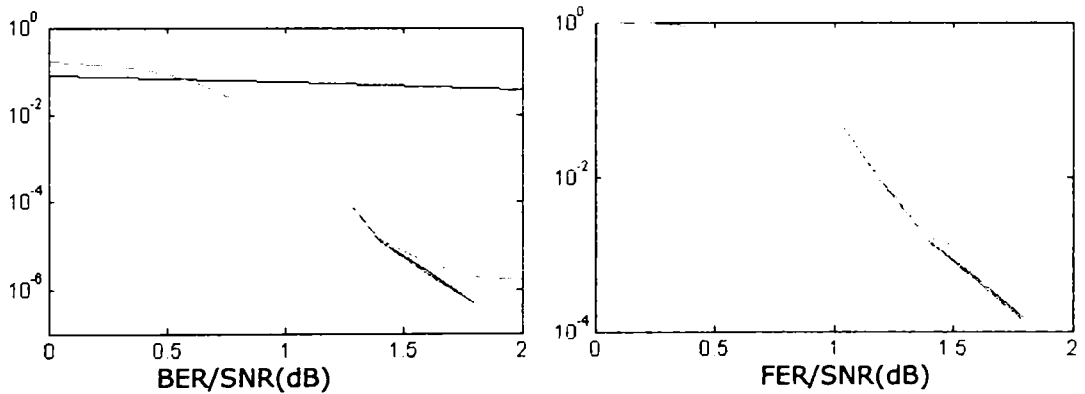
În cazul MaxLogMAP s-au luat în considerare toate valorile APP. Exceptând APP-le simbolurilor maxim plauzibile (care sunt toate nule) celelalte APP-uri iau valori tot mai mari pe vreme ce SNR-ul și i cresc.

Aceste rezultate confirmă supozițiile anterioare asupra valorilor APP. Întrebarea care rămâne este asupra valorilor pragurilor μ optime. Curbele BER și FER din Figura 5.10 compară performanțele turbo-codului cu și fără criteriul de stop. Am utilizat praguri cu valorile: 0,99 și 0,9999 pentru MAP și 4 și 10 pentru MaxLogMAP. Rezultatele arată că a doua valoare a pragurilor pentru fiecare algoritm dă practic aceleași performanțe ca și în cazul fără stop.

Pentru a evalua eficiența criteriilor MLAPP propuse în Figura 5.11 am trasat dependența numărului mediu de iterații efectuate ca o funcție de SNR, și am comparat cu cazurile "fără oprire" și "genial" [NGT01]. Criteriul "genial" oprește iterațiile când (și numai când) nu mai există erori în blocul decodat.



a) MAP, albastru – 15 it. fără stop; verde – stop cu prag =0,99; roșu – stop cu prag =0,9999;



b) MaxLogMAP, albastru – 15 it. fără stop; verde – stop cu prag =4; roșu – stop cu prag =10;

Figura 5.10 Performanțele BER și FER pentru două valori ale pragului μ

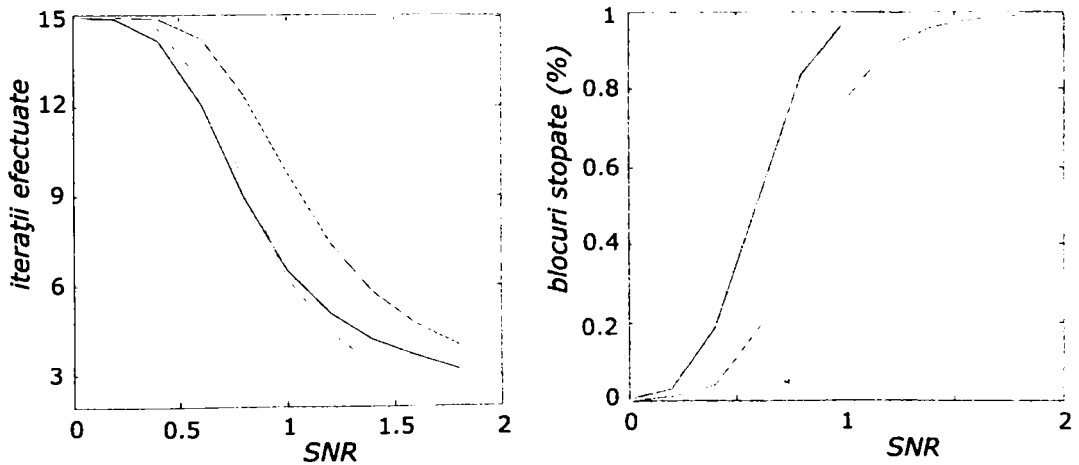


Figura 5.11 Eficiența criteriului de stop MLAPP (albastru-MAP cian-MAPgenial roșu-MaxLogMAP verde-MaxLogMAPgenial)

5.7 Proiectarea MBTC-lor

În paragraful 5.3 s-a arătat superioritatea formei canonice G față de forma canonică H. Acest rezultat a fost utilizat în proiectarea MBTC. Proiectarea, în contextul prezentei lucrări, presupune o căutare exhaustivă asupra mulțimii codurilor convoluționale candidate și asupra mulțimii interleaverelor. Criteriul de selecție propus în această lucrare a fost convergența MBTC-lui. În acest paragraf sunt prezentate rezultate ale căutării codurilor convoluționale ce pot fi utilizate ca și coduri componente în MBTC-uri. În această primă etapă a căutării, pentru construcția MBTC-lui s-a folosit un interleaver aleator de tip S. O a doua etapă a proiectării a constatat într-o căutare (de asemenea exhaustivă) asupra interleaverelor de tipul celui propus în [BSD04]. Mulțimea codurilor componente investigate a fost cea a codurilor duo-binare de memorie 2, 3, 4 și 5. Proiectarea a fost pe deplin finalizată doar în cazul codurilor de memorie 2 și 3. Un rezultat remarcabil prezentat în secțiunea 5.3.7 îl constituie două MBTC-uri de memorie 3 a căror performanță BER/FER depășește semnificativ pe cea a MBTC propus în [DOB05].

5.7.1 Convergența procesului iterativ de decodare

Atunci când se utilizează un criteriu de oprire a iterațiilor, asemeni celui descris în paragraful anterior, convergența procesului iterativ de decodare devine un criteriu în plus de apreciere a calității unui cod component. Mai mult, în această lucrare s-a utilizat convergența turbo-codului pentru selecția codurilor componente sau a interleaverelor. Vom începe prin a da o definiție pentru convergența turbo-codului. Și anume, convergența turbo-codului este egală cu raportul dintre numărul (total, cumulat) de iterații necesare, $iter(Nb)$ pentru decodarea a Nb blocuri turbo-codate și Nb , atunci când Nb este foarte mare:

$$C_v = \lim_{Nb \rightarrow \infty} \frac{iter(Nb)}{Nb}. \quad (5.49)$$

De remarcat că în definiția anterioară nu s-a specificat dacă respectivele blocuri au fost decodate corect sau nu, astfel că cele Nb blocuri pot să fie blocuri convergente sau nu, decodate corect sau nu, la care s-a aplicat criteriul de oprire sau nu. Cu alte cuvinte, C_v dă o măsură a „rapidității” de decodare a turbo-codului în cauză. Un alt amănunt de care trebuie să se țină seama în definiția convergenței este faptul că la blocurile neconvergente se efectuează numărul maxim de iterații, $iter_{max}$, care este setat apriori. Rezultă că C_v va fi o funcție și de acest parametru. De asemenea, C_v va depinde și de SNR-ul la care se lucrează. Cu certitudine, la SNR-uri mici turbo-decodorul va decoda corect puține blocuri sau chiar deloc, încât C_v va tinde spre $iter_{max}$. Pe vreme însă ce SNR-ul crește, ponderea blocurilor neconvergente scade astfel că dependența lui C_v de $iter_{max}$ este tot mai mică. C_v va depinde tot mai mult de turbo-codul utilizat. Concluzionând, am putea scrie că:

$$C_v = C_v(iter_{max}, SNR, \text{turbo-cod}). \quad (5.50)$$

Cu toate că în definiția (5.49) am considerat că Nb tinde spre infinit, un fapt remarcat prin simulări practice este că C_v calculat pentru un Nb mic, de ordinul a câtorva zeci sau sute, este o bună aproximație a convergenței. Pe de altă parte, tot prin observații practice s-a dovedit că un turbo-cod bun, care prezintă bune

performanțe BER sau FER, are și o bună convergență. Pornind de la aceste două observații practice, am construit o metodă de proiectare a turbo-codurilor prin căutare exhaustivă asupra mulțimii codurilor componente și a interleaverelor.

5.7.2 Selecția codului component

Am avut în vedere, ca și coduri componente, codurile convoluționale duobinare (cu $R=2$ intrări), de memorie $M=2, 3, 4$ și 5 . Utilizând forma canonică G , descrisă în paragraful 5.3, rezultă că o matrice G are $(R+1)(M+1)$ elemente. Cu restricția ca $g_{0,0} = 1$, rezultă că există $2^{(R+1)(M+1)-1}$ variante de alegere a matricii G pentru fiecare R și M . Practic însă numărul de variante este mai mic datorită a câtorva restricții. În primul rând, inversarea intrărilor între ele nu conduce la un alt cod. Astfel s-au considerat doar cazurile în care $g_2 > g_1$. O altă restricție se referă la impunerea ca polinomul de reacție (g_0) să fie de grad maxim. Cu toate aceste restricții numărul codurilor potențial candidate, în cazurile $M>3$ este de ordinul miilor. Este imperios necesară o selecție eficientă, în sensul de rapidă, a codurilor componente. Dintre criteriile de selecție ar putea fi distanța minimă de cod a turbo-codului sau simularea performanței BER/FER. Simularea performanței este un criteriu foarte elocvent, însă este o procedură anevoioasă datorită volumului mare de simulări necesare. Deși nici calculul distanțelor de cod nu este foarte simplu, reprezintă totuși o cale mai rapidă, mai ales dacă se face nu foarte exact ci estimativ [GAC04]. Nu este însă la fel de elocventă în acest ultim caz.

Criteriul de selecție propus în continuare este destul de eficient în ceea ce privește timpul de căutare și, de asemenea, s-a dovedit destul de precis. Anume, selecția s-a bazat pe convergența turbo-decodorului. Concret, s-a simulat funcționarea turbo-codului cu fiecare dintre codurile componente candidate. Simulările s-au oprit dacă convergența C_v a depășit un prag impus:

$$\text{oprește simularea dacă } C_v(Nb) > C_{v0} - Nb \cdot dC_v. \quad (5.52)$$

Pragul impus $C_{v0} - Nb \cdot dC_v$ scade cu Nb pentru ca simularea pentru un anumit cod să nu dureze la nesfârșit. În plus, alegând convenabil parametrii C_{v0} și dC_v se poate eficientiza selecția, în sensul că să se simuleze puține blocuri pentru codurile componente neperformante și suficiente blocuri pentru a forma o „păreră” pertinentă asupra codurilor componente performante. Pentru a evita cazul în care un cod performant să fie simulat (datorită hazardului) printr-un bloc neconvergent, caz în care ne-am forma o păreră greșită despre acel cod și l-am pierde, se poate impune ca verificarea condiției de selecție (5.52) să nu fie făcută după fiecare bloc ci după efectuarea simulării unui număr minim, elocvent, de blocuri. Practic acest număr a fost ales egal cu 10. Parametrii C_{v0} și dC_v au fost aleși funcție de memoria codului component.

5.7.3 Rezultatele căutării codurilor componente

În continuare vom prezenta câteva dintre rezultatele căutării codurilor componente pentru fiecare valoare a lui M , de la 2 la 5. Rezultatele sunt prezentate sub forma unor tabele ale căror căsuțe sunt colorate pentru a sugera performanța în convergență. În cazul $M=2$ sunt posibile două coduri cu rol de polinom de reacție: $g_0=5$ sau $g_0=7$. Pentru cele două polinoame în Figura 5.12 sunt prezentate rezultatele căutării.

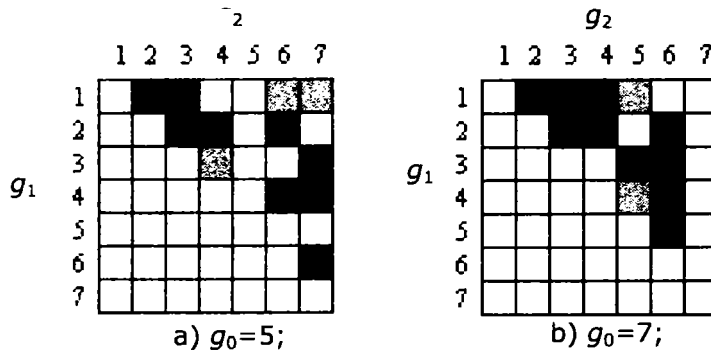


Figura 5.12 Rezultatele căutării codurilor componente cu $R=M=2$;
Lungime întretesere 752×2 biți; $SNR=3.4dB$; Roșu: $C_v \leq 3.5$;
verde: $3.5 < C_v < 4.5$; negru: $C_v \geq 4.5$; alb: nedecodabile;

Cele mai bune 4 coduri găsite au matricele generatoare G egale cu: $G_1=[7,3,5]$, $G_2=[7,6,5]$, $G_3=[5,3,7]$ și respectiv $G_4=[6,5,7]$. De remarcat că matricea G_2 nu are un echivalent în forma canonică H , adică codorul respectiv nu poate fi construit decât în forma canonică G . Acest rezultat este un argument în plus pentru utilizarea formei canonice G pentru MBTC.

În cazul $M=3$ există patru polinoame candidate pentru rolul de polinom de reacție (g_0): 9, 11, 13 și 15. În Figura 5.13 sunt prezentate rezultatele căutării în cazurile polinoamelor de reacție primitive: 11 și 13. Codurile componente cu convergența mai mică decât 4 au fost selectate pentru a doua etapă a căutării. În această etapă s-a simulat, cu fiecare dintre codurile selectate, transmiterea a $Nb=500000$ blocuri la un $SNR=2.0dB$. Rezultatele sunt prezentate în Tabelul 5.3. Culoarea codului simbolizează convergența găsită în faza de căutare (partea stângă a tabelului). Cu destul de bună aproximare convergența stabilită în această fază este egală cu cea stabilită pentru $Nb=500000$ blocuri (partea din dreapta a tabelului).

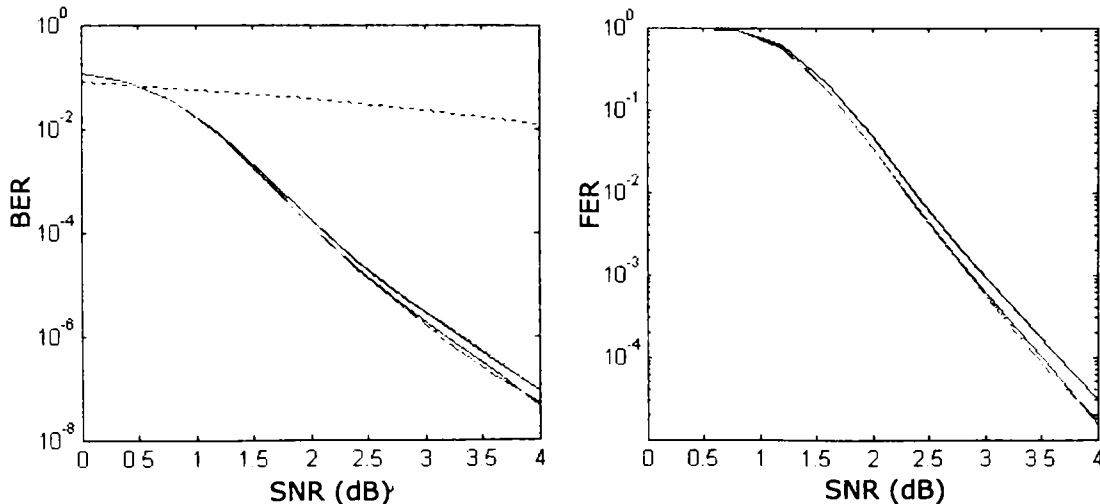
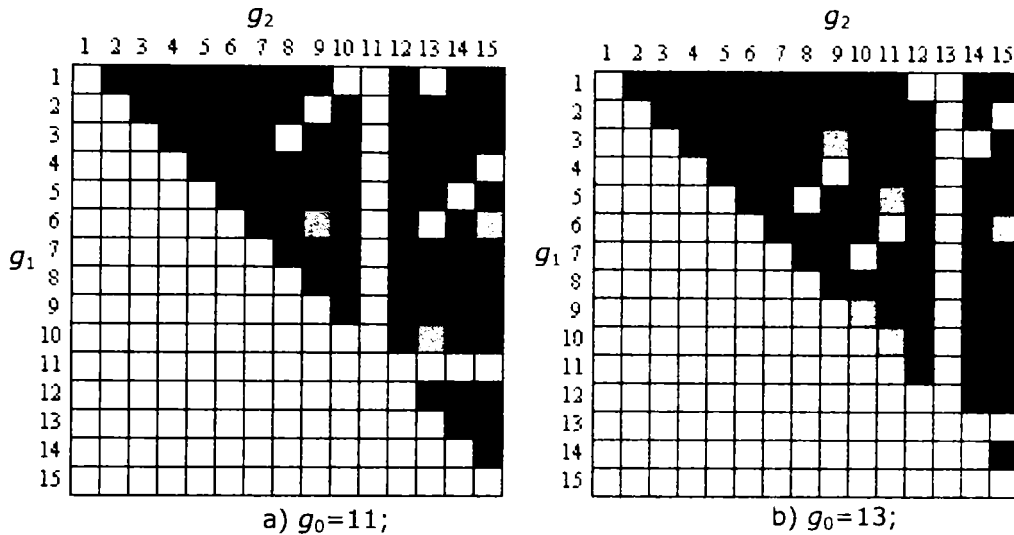


Figura 5.13 Performanțele codurilor componente selectate la memorie 2:
albastru: $G_1=[7,3,5]$, roșu: $G_2=[7,6,5]$, verde: $G_3=[5,3,7]$, negru $G_4=[6,5,7]$



a) $g_0=11$;
 b) $g_0=13$;
 Figura 5.14 Rezultatele căutării codurilor componente cu $R=2$ $M=3$;
 Lungime întretesere 752×2 biți; $SNR=2.0$ dB; Roșu: $C_v \leq 3.5$; albastru: $3.5 < C_v < 3.7$;
 verde: $3.7 < C_v < 4.0$; negru: $C_v \geq 4.0$; alb: nedecodabile;

Tabelul 5.3 Codurile duobinare componente de memorie 3 selectate.

	Căutarea după convergență									Simulare la $N_b=500000$ blocuri				
	G	C_v	bl_{nst}	N_b	er	BER	FER	er	BER	FER	C_v	bl_{nst}		
1	14 9 5	9.300	4	10	0	0	0							
2	11 1 7	7.800	2	10	0	0	0							
3	13 6 9	4.045	11	580	0	0	0							
4	9 6 11	3.777	10	750	0	0	0	178	236	82	3.697	3017	***	
5	13 9 11	3.647	8	850	0	0	0	186	247	70	3.595	2822	***	
6	13 10 11	3.846	8	700	8	7598	2857	567	753	262	3.935	8853		
7	15 6 11	3.800	5	750	0	0	0	184	244	98	3.814	4278	****	
8	15 9 11	3.358	4	1000	0	0	0	158	210	72	3.378	2398	***	
9	15 13 11	3.501	5	900	0	0	0	223	296	106	3.491	3054	****	
10	9 3 13	3.994	4	650	0	0	0	328	436	166	4.020	6814		
11	9 6 13	3.675	4	800	2	1662	1250	100	132	54	3.690	3068	**	
12	10 9 13	3.926	7	650	0	0	0	507	674	224	4.024	8255		
13	11 5 13	3.874	12	700	0	0	0	383	509	174	3.929	8856		
14	11 9 13	3.585	7	850	0	0	0	89	118	34	3.592	2782	*	
15	11 10 13	3.994	9	650	0	0	0	550	731	226	4.036	8734		
16	15 6 13	3.771	3	750	0	0	0	175	232	78	3.780	3541	***	
17	15 9 13	3.348	2	1000	0	0	0	160	212	70	3.372	2158	***	
18	15 11 13	3.472	4	950	0	0	0	107	142	50	3.497	3040	**	
19	11 9 15	3.844	12	700	0	0	0	547	727	218	3.781	7717		
20	13 9 15	3.861	12	690	0	0	0	525	698	224	3.795	8003		
21	13 11 15	3.902	17	660	8	8059	1515	1668	2218	550	3.910	13712		

Așa cum arată Tabelul 5.3, cele mai bune coduri au ca și polinom de reacție polinoamele primitive care există la memorie 3 (11, 13). Deși au fost investigate, codurile care au polinom primitiv în reacție prezintă performanțe mult mai slabe.

În coloana din dreapta a Tabelului 5.3 au fost selectate cele mai bune 10 coduri ce au rezultat în urma simulării transmiției a 500000 de blocuri cu fiecare cod. Printre acestea se regăsește codul de memorie 3 propus în [DOB05], care are matricea generatoare echivalentă $G = [15 \ 13 \ 11]$. Pentru aceste 10 coduri, s-a făcut o căutare asupra mulțimii interleaverelor de tipul celui definit în [BSD04]. Interleaverelor în discuție sunt definite de 4 parametri: P , P_1 , P_2 și P_3 . Relația de întrețesere utilizată în [BSD04] este:

$$\pi(j) = (P*j + Q + 1) - N * \text{int}((P*j + Q + 1) / N), \quad \forall 0 \leq j < N \quad (5.53)$$

unde $P=19$, $P_1=376$, $P_2=224$, $P_3=600$, $\text{int}(x)$ semnifică partea întregă din x iar:

$Q=0$ dacă $j=0 \pmod{4}$,

$Q=\text{int}(N/2)+P_1$ dacă $j=1 \pmod{4}$,

$Q=P_2$ dacă $j=2 \pmod{4}$ și

$Q=\text{int}(N/2)+P_3$ dacă $j=3 \pmod{4}$.

În căutarea efectuată s-a considerat $P=19$ și $P_1=376$, iar P_2 și P_3 variabile întregi cu valori cuprinse între 0 și $N=752$.

La fel ca și în cazul căutării codurilor, și în cazul interleaverelor s-a utilizat criteriul de selecție „convergența turbo-codului”. Rezultatele căutării sunt prezentate în Tabelul 5.4. În coloanele notate „total_it” și „bl_st” s-au indicat numărul total de iterații efectuate respectiv numărul de blocuri stopate, adică numărul de blocuri pentru care, datorită aplicării criteriului de stop a iterațiilor, nu s-au efectuat numărul maxim de iterații (care a fost 15). Cele mai bune combinații cod-interleaver găsite, conform Tabelul 5.4 sunt:

- cod cu $G_1=[15 \ 9 \ 11]$ și interleaver cu $P_2=328$, $P_3=196$,

- cod cu $G_2=[15 \ 9 \ 13]$ și interleaver cu $P_2=203$, $P_3=677$.

Tabelul 5.4 Rezultatele căutării interleavelor pentru MBTC-uri cu $M=3$

	G	index	P_2	P_3	C_v	er	BER	FER	total	it	bl	st
1	11 9 13	a	259	665	3.4200	200	265	104	1799960	496983		
		b	335	449	3.4198	162	215	76	1793783	497504		
		c	483	209	3.4074	167	222	140	1789548	497396		
2	9 6 13	a	259	281	3.5490	141	187	64	1846614	497301		
3	15 11 13	a	80	736	3.3977	102	135	42	1748396	497347		
		b	176	64	3.3988	173	230	70	1750041	497145		
		c	228	204	3.3988	115	152	48	1758102	496533		
		d	259	501	3.3988	154	204	76	1747692	497439		
		e	283	273	3.4023	158	210	72	1753357	497326		
4	15 9 11	a	239	273	3.3012	33	43	18	1683494	498414		
		b	267	729	3.3167	330	438	284	1699131	497232		
		c	300	516	3.3167	29	38	18	1676157	498515		
		d	315	561	3.3052	60	79	32	1680238	498503		
		e	328	196	3.3167	18	23	10	1684857	498128		
		f	332	144	3.3086	45	59	24	1680969	498350		
		g	352	132	3.2652	115	152	50	1683686	498098		
		h	403	517	3.3049	66	87	28	1678328	498367		
		i	440	548	3.3165	306	406	142	1715723	497171		
		j	476	276	3.3086	45	59	18	1687245	498291		
		k	480	728	3.3049	100	132	48	1689809	497904		
		l	547	17	3.3167	86	114	42	1683585	498301		
		m	607	37	3.2624	70	93	30	1682036	498357		
		n	623	593	3.3167	39	51	20	1679437	498433		
5	15 6 13	a	507	217	3.5000	174	231	78	1886266	497531		
6	15 9 13	a	3	49	3.3089	51	67	26	1689552	498455		
		b	79	93	3.3089	53	70	28	1680917	498483		
		c	87	113	3.3049	92	122	46	1680369	498399		
		d	203	677	3.2577	29	38	12	1676133	498580		
		e	232	208	3.2846	58	77	36	1679852	498319		
		f	240	644	3.2884	79	105	44	1682216	498287		
		g	240	708	3.2790	37	49	16	1674804	498574		
		h	259	33	3.3012	156	207	82	1691601	498199		
		i	263	129	3.2841	75	99	30	1690224	498454		
		j	276	128	3.3049	129	171	64	1680597	498232		
		k	291	37	3.3012	66	87	36	1683319	498453		
		l	291	521	3.3049	99	131	32	1685177	498448		
		m	295	201	3.2688	35	46	16	1676686	498524		
		n	300	440	3.2809	44	58	22	1679844	498299		
		o	396	648	3.3049	75	99	36	1692226	497798		
		p	408	528	3.3052	59	78	30	1680337	498408		
		q	588	440	3.3086	102	135	40	1689776	498274		
r	611	749	3.3086	89	118	40	1675730	498444				
7	9 6 11	a	536	108	3.5200	128	170	74	1845454	497335		
8	13 9 11	a	363	577	3.4326	198	263	104	1796021	497226		
		b	427	593	3.4420	185	246	88	1798979	497205		
9	15 6 11	a	311	221	3.5490	117	155	54	1906276	496937		
10	13 15 11	brst				230	305	80	1750802	497510		
		A	83	125	3.4316	57	75	28	1735435	497614		
		C	251	641	3.4590	89	118	44	1733983	497938		
		D	479	689	3.4426	211	280	92	1743028	497595		
		a	148	140	3.3898	63	83	30	1735187	497590		
		b	295	353	3.3889	113	150	52	1747256	497413		
		c	339	197	3.3898	91	121	44	1732105	497959		
d	343	505	3.3889	77	102	38	1732616	497934				
B	595	97	3.3817	53	70	28	1731179	497916				

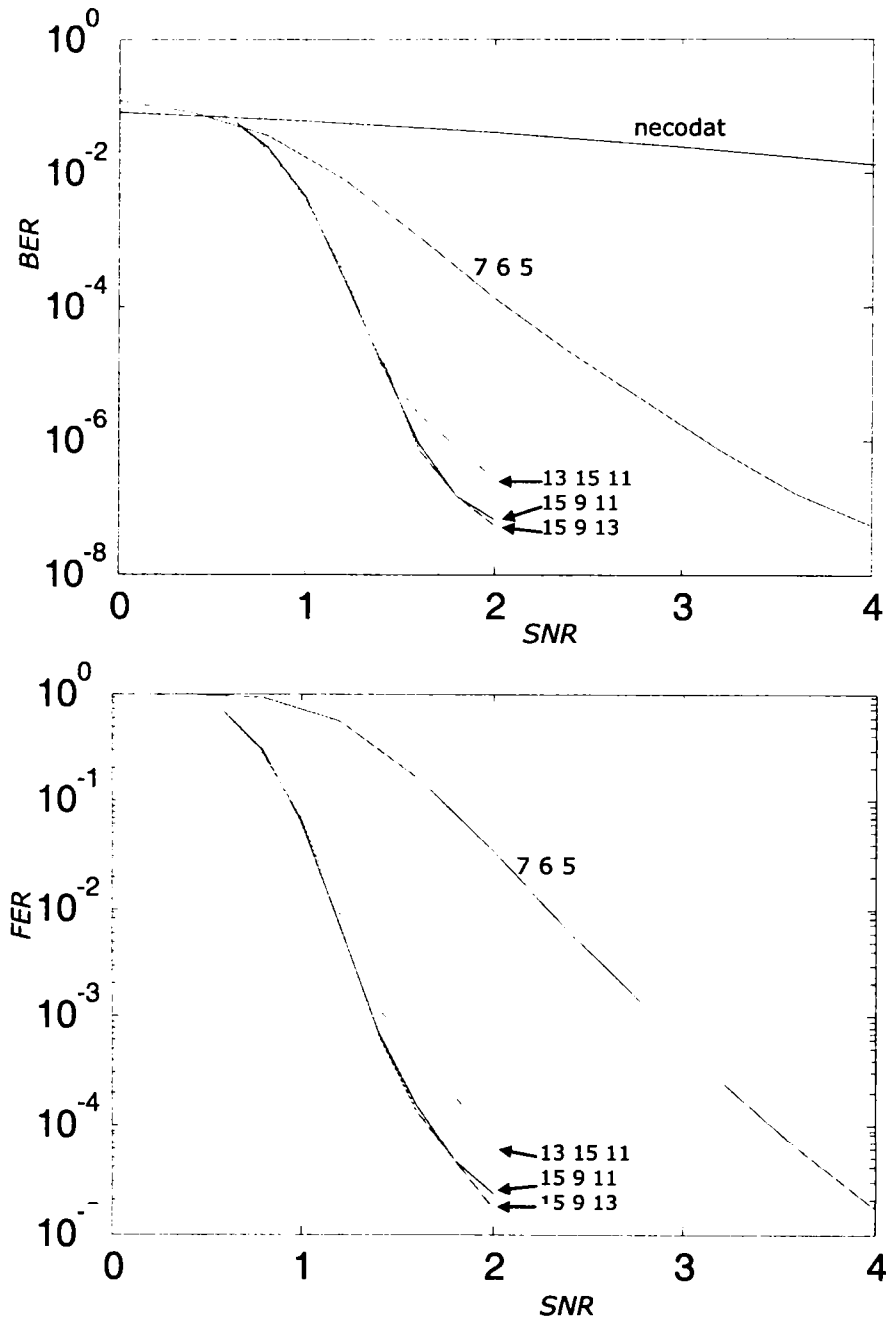


Figura 5.15 Performanțele BER și FER vs SNR ale MBTC-urilor de memorie 2 și 3

Pentru aceste două coduri, în Figura 5.15 s-au trasat curbele BER/SNR și FER/SNR și s-au comparat cu codul definit în [DOB05]. Se observă că cele două coduri oferă performanțe BER și FER superioare începând cu un SNR ≥ 1.5 dB. Considerând un FER $< 10^{-4}$, câștigul de codare este de aproximativ 0.3dB față de codul de referință.

Și în cazul codurilor de memorie 4 selecția prin căutare după criteriul convergenței a pus în evidență codurile cu polinoame de reacție primitive, 19 și 25

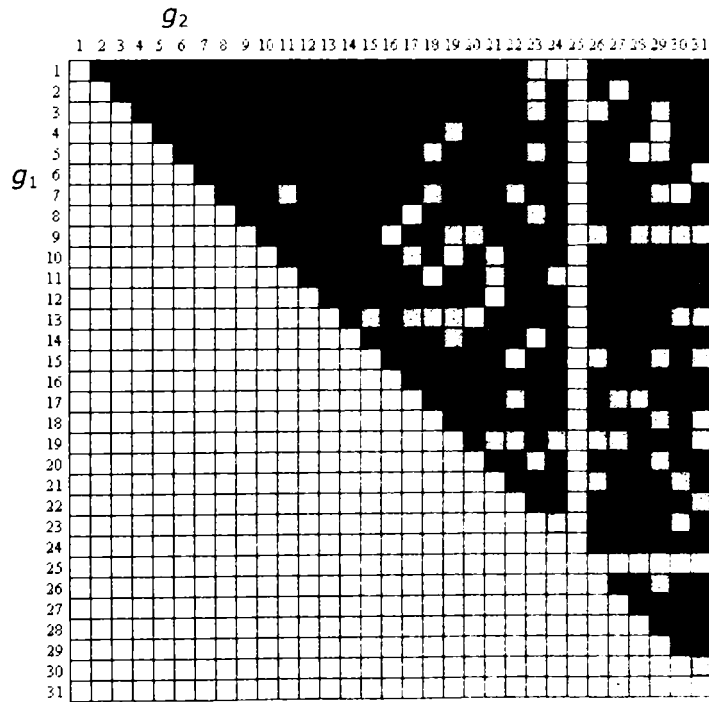
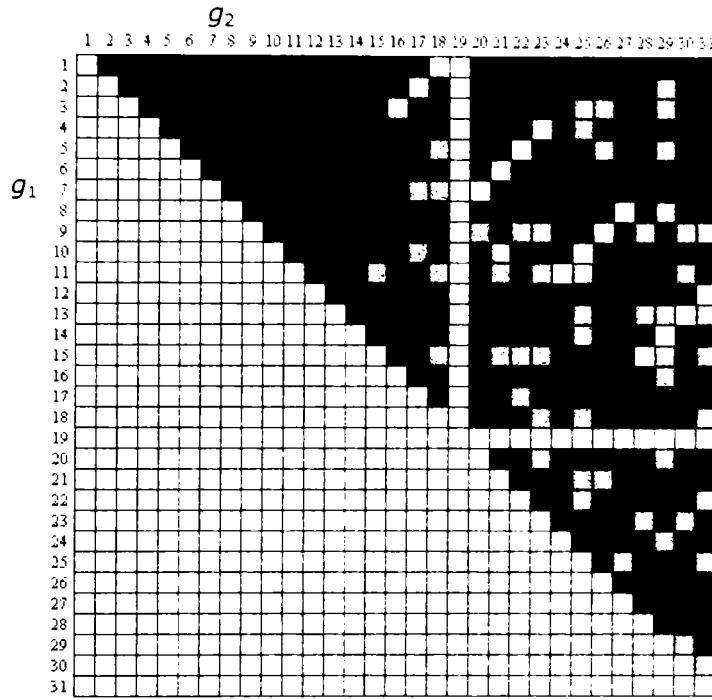


Figura 5.16 Rezultatele căutării codurilor componente cu $R=2$ $M=4$;
 Lungime întrețesere 752×2 biți; $SNR=1.6dB$; Roșu: $C_v < 3.5$; albastru: $3.5 < C_v < 3.7$;
 verde: $3.7 < C_v < 4.0$; negru: $C_v \geq 4.0$; alb: nedecodabile;

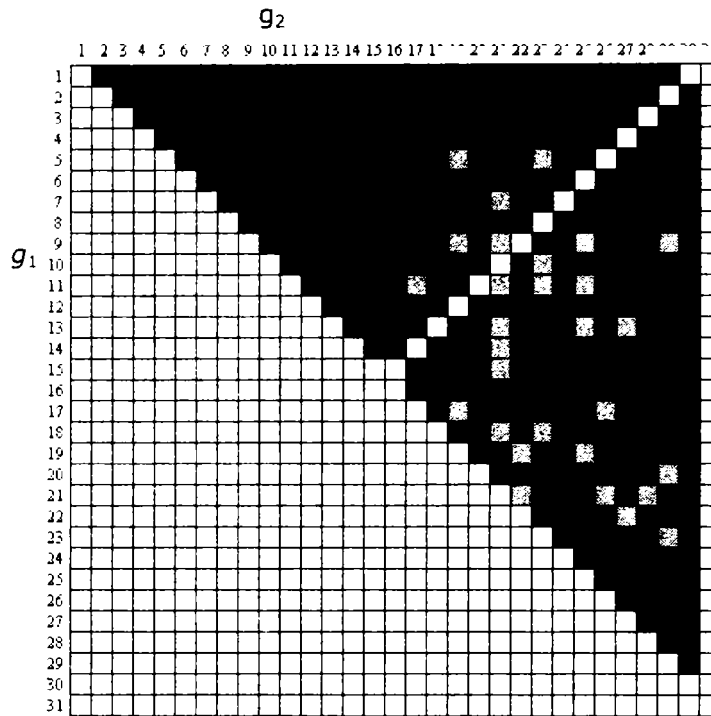
c) $g_0=31$;

Figura 5.16 (continuare) Rezultatele căutării codurilor componente cu $R=2$ $M=4$; Lungime întretesere 752×2 biți; $SNR=1.6dB$; Roșu: $C_v \leq 3.5$; albastru: $3.5 < C_v < 3.7$; verde: $3.7 < C_v < 4.0$; negru: $C_v \geq 4.0$; alb: nedecodabile;

în acest caz. În Figura 5.16 sunt prezentate rezultatele căutării pentru codurile ce au ca și polinoame de reacție polinoamele 19, 25 respectiv 31 iar în Anexa B sunt prezentate codurile rezultate în urma selecției după criteriul convergenței. Tabelul din Anexa B prezintă comparativ căutarea făcută la un $SNR=1.6dB$ cu cea făcută la $SNR=1.8dB$. Practic ierarhia este cu bună aproximație aceeași pentru ambele cazuri.

De asemenea, pentru toate cele 178 de coduri selectate după faza de căutare s-au efectuat simulări la 500000 de blocuri. Rezultatele acestor simulări sunt prezentate în coloanele din partea dreaptă (Anexa B). Performanțele codului de referință, propus în [DOB05], sunt marcate în linia notată cu numărul „0” în tabelul Anexei B.

Atât în căutare cât și pentru simulările la 500000 de blocuri în construcția TCMB-ului s-a utilizat un interleaver de tip S cu lungimea de 752 de simboluri de doi biți, ca și în cazul $M=3$. Excepție o reprezintă codul de referință, pentru care s-a utilizat interleaverul indicat în aceeași referință [DOB05]. De remarcat că același cod component, definit prin matricea $G=[23 \ 21 \ 25]$ și aflat la poziția 116 (Anexa B), deși s-a regăsit printre codurile selectate, nu prezintă aceleași performanțe în conjuncție cu interleaverul de tip S.

6. Turbo-Coduri Multi-Non-Binare

6.1 Introducere

În acest capitol se propune o nouă familie de turbo-coduri ale căror coduri constituente au $R \geq 1$ intrări non-binare și $R+1$ ieșiri, [BH31]. Această nouă familie, numită „Turbo-coduri multi-non-binare” -The Multi-Input Non-Binary Turbo-Codes (MNBTC), reprezintă o extensie a conceptului MBTC-ilor la cazul non-binare. Noua familie include turbo-codurile multi-binare (MBTC) care la rândul ei include turbo-codurile clasice uni-binare (BTC). Formal codul are aceeași structură ca și în cazul multi-binar, însă în acest caz, operațiile aritmetice sunt efectuate într-un câmp Galois de ordin superior lui 2, $GF(2^Q)$.

Capitolul 6 este structurat în felul următor. În paragraful 6.2 se prezintă construcția codurilor constituente pentru un MNBTC. În paragraful 6.3 se prezintă adaptarea algoritmului de decodare Maximum A Posteriori (MAP) pentru decodarea MNBTC-ilor. Paragraful 6.4 este dedicat problematicii închiderii trellisurilor codurilor convoluționale multi-non-binare. În paragraful 6.5 se arată că, impunând câteva restricții asupra codurilor convoluționale multi-non-binare se pot obține coduri Reed-Solomon, astfel că familia MNBTC-ilor include o nouă și atractivă (îmbinând calitățile TC-ilor cu ale codurilor Reed-Solomon) clasă a turbo-codurilor Reed-Solomon (RSTC). În final, în paragraful 6.6 sunt prezentate și argumentate pe baza simulărilor efectuate beneficiile utilizării acestei noi clase de turbo-coduri.

6.2 Codor convoluțional multi-non-binar

În acest paragraf este descris procesul de codare pentru codurile constituente MNBTC-ilor. Fiecare codor constituent are R intrări non-binare fiind numit în continuare MNB (multi-non-binary). În Figura 6.1 se prezintă schema generală a unui codor convoluțional MNB, de rată $R_c=R/(R+1)$. Pe parcursul acestei lucrări s-au avut în vedere codurile recursive și sistematice știută fiind superioritatea lor în turbo-coduri. Fiecare celulă de întârziere din Figura 6.1 stochează un vector de Q biți la fiecare moment de tact. Toate legăturile sunt magistrale de Q legături simple astfel că se transportă simultan Q biți. Fiecare bloc notat $g_{r,m}$, cu $r=1, \dots, R$, $m=0, \dots, M-1$, reprezintă un multiplicator peste $GF(2^Q)$ în vreme ce sumatoarele efectuează sume peste același câmp $GF(2^Q)$. La fiecare moment de tact n , codorului i se furnizează, prin cele R intrări, R simboluri sau caractere $u_1^n, u_2^n, \dots, u_R^n$. Codorul, la rândul său, generează, în fiecare moment de tact n , la cele $R+1$ ieșiri ale sale, $R+1$ simboluri notate $x_1^n, x_2^n, \dots, x_R^n$, corespunzătoare celor R simboluri de intrare și un simbol redundant x_0^n numit de asemenea c^n , pentru distincție. Starea curentă a codorului este dată de ieșirile celor M celule ale registrului de întârziere $s_0^n, s_1^n, \dots, s_{M-1}^n$. Vom adopta, pentru vectorul stare al codorului și, respectiv, pentru „cuvântul

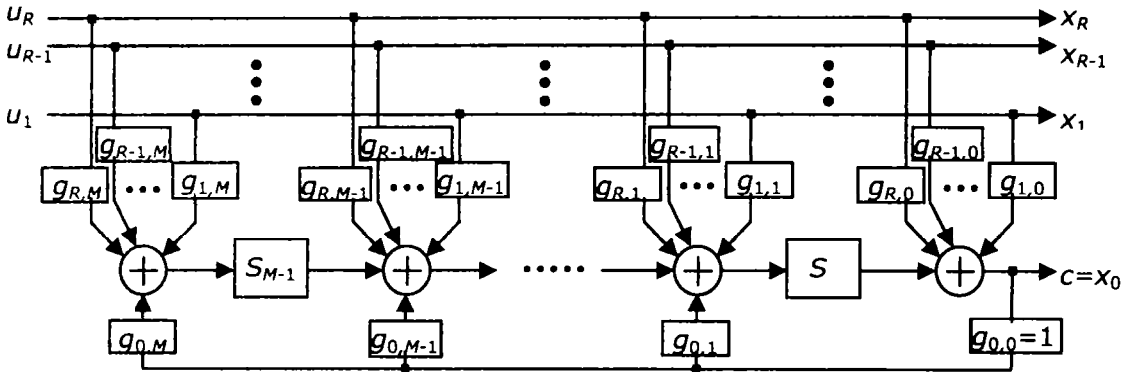


Figura 6.1 Codor convoluțional Multi-Non-Binar- schemă generală

de intrare”, următoarele notații compacte: $s^n = [S_{M-1}^n \ S_{M-2}^n \ \dots \ S_0^n]^T$,

$u^n = [u_R^n \ u_{R-1}^n \ \dots \ u_1^n]^T$, cu $0 \leq n \leq N$.

Relațiile intrare/stare curentă și ieșire/stare curentă ale codorului pentru momentul de tact n pot fi exprimate în forma compactă:

$$(s^{n+1})_{M \times 1} = (G_T)_{M \times R} \cdot (u^n)_{R \times 1} + (T)_{M \times M} \cdot (s^n)_{M \times 1}, \quad (6.1)$$

$$c^n = G_L \cdot u^n + W \cdot s^n. \quad (6.2)$$

unde: $G_T = G_F \cdot G_L + G_0$ și $W = [0 \ 0 \ \dots \ 0 \ 1]_{1 \times M}$. $G_0 = [g_{r,m}]_{M \times R}$ denotă matricea generatoare restrânsă, cu $1 \leq m \leq M$, $1 \leq r \leq R$, matrice ce exclude coeficienții de reacție și coeficienții utilizați pentru generarea simbolului redundat. Vectorul $G_F = [g_{0,M} \ g_{0,M-1} \ \dots \ g_{0,1}]^T$ conține coeficienții de reacție iar $G_L = [g_{R,0} \ g_{R-1,0} \ \dots \ g_{1,0}]$. Matricea de tranziție T este:

$$T = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & g_{0,M} \\ 1 & 0 & 0 & \dots & 0 & g_{0,M-1} \\ 0 & 1 & 0 & \dots & 0 & g_{0,M-2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & g_{0,1} \end{bmatrix} = \begin{bmatrix} 0_{1 \times M-1} & G_F \\ I_{M-1} & \end{bmatrix}.$$

Matricea generatoare completă G este: $G = [g_{r,m}]_{(M+1) \times (R+1)} = [g_R \ g_{R-1} \ \dots \ g_1 \ g_0]_{10}$, cu $0 \leq m \leq M$, $0 \leq r \leq R$.

Condiția necesară și suficientă pentru a avea un cod decodabil este ca matricea G_T să fie de rang maxim.

Aplicând transformata „ D ” ($X(D) = \sum_{k=-\infty}^{+\infty} x^k \cdot D^k$) ecuațiilor (6.1) și (6.2)

obținem:

$$D^{-1} \cdot S(D) = G_T \cdot U(D) + T \cdot S(D), \quad (6.3)$$

$$C(D) = G_L \cdot U(D) + W \cdot S(D). \quad (6.4)$$

Eliminând pe $S(D)$ din cele două ecuații, se poate arăta că:

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Figura 6.2 Adunarea și multiplicarea în $GF(4)$.

$$C(D) = \sum_{r=1}^R \frac{g_r(D)}{g_0(D)} \cdot U(D) \quad (6.5)$$

unde $g_r(D) = \sum_{m=0}^M g_{r,m} \cdot D^m$.

Exemplul 6.1 Cu scopul de a explica procedura de codare, vom da un exemplu în $GF(4)$. Operațiile de adunare și multiplicare din $GF(4)$ sunt prezentate în Figura 6.2. Considerăm codul convoluțional duo-non-binar definit prin matricea generatoare:

$$G = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 3 & 1 \\ 3 & 1 & 1 \end{bmatrix}. \quad (6.6)$$

De asemenea, considerăm secvențele de intrare $u_1 = [1 \ 2 \ 3 \ 3 \ 1 \ 3 \ 2 \ \dots]$ și $u_2 = [0 \ 2 \ 1 \ 1 \ 3 \ 0 \ 3 \ \dots]$. Prin construcție, din G , avem că: $G_L = [3 \ 1]$, $G_F = [3 \ 1]^T$, $G_T = [0 \ 2; 3 \ 2]$ și $T = [0 \ 3; 1 \ 1]$. Valorile vectorului de stare ale codorului se determină prin (6.1) pornind cu starea inițială $s^0 = [0 \ 0]^T$. Urmând operațiile de codare indicate prin (6.1) și (6.2) cât și tablele din Figura 6.2, rezultă recurențial că $s^1 = [0 \ 3]^T$ și $c^1 = 3$, $s^2 = [1 \ 1]^T$ și $c^2 = 0$, $s^3 = [1 \ 0]^T$ și $c^3 = 2$, $s^4 = [2 \ 1]^T$ și $c^4 = 3$, $s^5 = [2 \ 1]^T$ și $c^5 = 1$ și $s^6 = [3 \ 1]^T$ și $c^6 = 3$, etc.

După descrierea procesului de codare al fiecărui cod constituent, vom descrie în continuare procesul de turbo-codare pentru MNBTC. Figura 6.3 prezintă schema generală a unui MNBTC. Un bloc de date de intrare $(u^n)_{0 \leq n < N} = [u^0 \ u^1 \ \dots \ u^{N-1}]$ este constituit dintr-o secvență de $N+1$ cuvinte de intrare consecutive. Fiecare cuvânt de dimensiune $R \times 1$ este alcătuit din R

simboluri consecutive, adică $u^n = [u_{R+1}^n \ u_R^n \ \dots \ u_2^n]^T$, cu $0 \leq n < N$. La rândul său, fiecare simbol este compus din Q biți: $u_r^n = [u_{r,Q-1}^n \ u_{r,Q-2}^n \ \dots \ u_{r,0}^n]^T$, $2 \leq r \leq R+1$, cu: $u_{r,q}^n \in \{0, 1\} = GF(2)$, $0 \leq q < Q-1$.

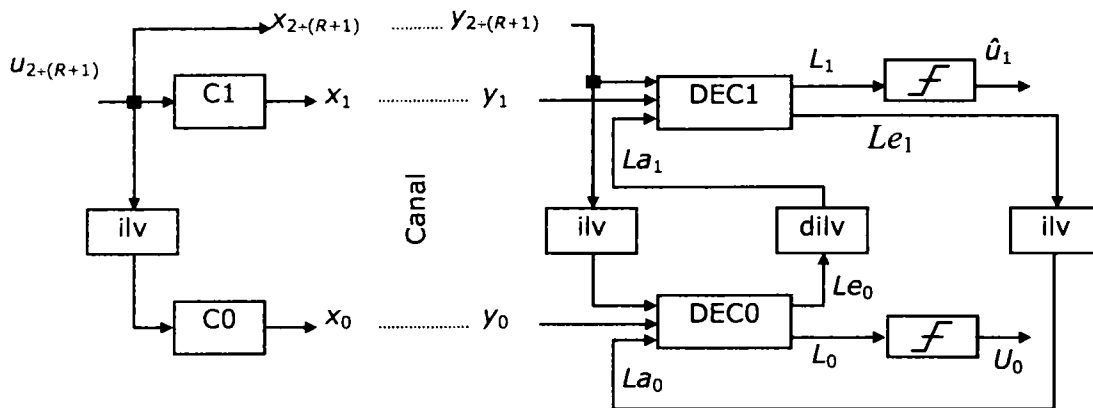


Figura 6.3 MNBTC – schema generală

La ieșirile codoarelor C1 și C0, se obține cuvântul de cod $(x^n)_{0 \leq n < N} = [x^0 \ x^1 \ \dots \ x^{N-1}]$, unde fiecare cuvânt $x^n = [x_{R+1}^n \ x_R^n \ \dots \ x_2^n \ x_1^n \ x_0^n]^T$ este compus din $R+2$ simboluri: primele R simboluri corespund simbolurilor de intrare iar ultimele două simboluri constituie redundanța. Asemeni simbolurilor de intrare, fiecare simbol din cuvântul de cod conține Q biți. Blocul codat, format din N cuvinte, care conțin $N \times (R+2)$ simboluri, ce corespund la $N \times (R+2) \times Q$ biți, este ulterior modulată (pentru simplitate, vom considera o modulație BPSK sau QPSK pe parcursul acestui capitol) și transmis către destinație. La destinație se recepționează o versiune alterată de zgomot $(y^n)_{0 \leq n < N}$ a semnalului emis. Blocul recepționat este format din N cuvinte: $y^n = [y_{R+1}^n \ y_R^n \ \dots \ y_2^n \ y_1^n \ y_0^n]^T$. Fiecare cuvânt y^n este un vector format din $R+2$ componente. Fiecare componentă poate fi reprezentată ea însăși printr-un vector de Q valori $y_r^n = [y_{r,Q-1}^n \ y_{r,Q-2}^n \ \dots \ y_{r,0}^n]^T$. În cazul transmisiei printr-un canal AWGN, fiecare valoare $y_{r,q}^n$ poate fi modelată ca: $y_{r,q}^n = x_{r,q}^n + w_{r,q}^n$, $0 \leq q < Q$, unde $w_{r,q}^n$ reprezintă zgomotul recepționat (variabilă aleatoare gaussiană, de medie nulă și dispersie σ^2).

6.3 Decodarea multi-non-binară

În acest paragraf sunt descrise modificările necesare a fi făcute asupra algoritmului de decodare MAP pentru a decoda MNBTC-uri.

Având în vedere cele două decodoare componente DEC1 și DEC0 din Figura 6.3, decodarea blocului recepționat, indiferent de algoritmul de decodare iterativ utilizat (MAP, LogMAP, MaxLogMAP sau SOVA), decodarea MNBTC-ilor poate fi făcută în trei maniere diferite: per cuvânt, per simbol sau per bit. Cele trei metode calculează în mod diferit probabilitățile a priori, extrinseci și a posteriori (APP). În cele ce urmează sunt descrise calculele survenite în cele trei metode:

Decodarea per cuvânt presupune calculul probabilităților a priori, extrinseci și a posteriori pentru fiecare valoare pe care poate să o ia cuvântul recepționat dintre cele $N_w = (2^Q)^R$ variante. Utilizând aceleași notații ca și în Figura 6.3, putem exprima APP-urile $L_j^{n,i}(d)$ și probabilitățile extrinseci $Le_j^{n,i}(d)$ pentru fiecare decodor j , cu $j = 0$ sau 1 , la iterația i astfel:

$$\begin{aligned} L_1^{n,i}(d) &= La_1^{n,i}(d) + Y_1^n + Le_1^{n,i}(d), \\ L_0^{n,i}(d) &= La_0^{n,i}(d) + Y_0^n + Le_0^{n,i}(d), \end{aligned} \quad (6.7)$$

unde $d \in \Delta = \{0, 1 \dots N_w - 1\}$ reprezintă indexul cuvântului candidat $u^n(d)$ dintre cele N_w cuvinte posibile. Y_1^n și Y_0^n sunt termeni calculați pe baza secvenței recepționate conform relației:

$$Y_j^n = \frac{1}{\sigma^2} \cdot \sum_{r=2q=0}^{R+1Q-1} x_{r,q}^n \cdot Y_{r,q}^n + \frac{1}{\sigma^2} \cdot \sum_{q=0}^{Q-1} x_{j,q}^n \cdot Y_{j,q}^n, \quad (6.8)$$

unde $j = 0$ sau 1 iar σ^2 reprezintă dispersia zgomotului.

Atunci, probabilitățile a priori pot fi exprimate ca:

$$\begin{aligned} La_1^{n,i}(d) &= \pi^{-1}(Le_0^{n,(i-1)}(d)), \\ La_0^{n,i}(d) &= \pi(Le_1^{n,(i-1)}(d)). \end{aligned} \quad (6.9)$$

Operațiile $\pi(\)$ și $\pi^{-1}(\)$ reprezintă întreteserea, respectiv deîntreteserea („ilv” și respectiv „dilv” în Figura 6.3).

Notăm că numărul componentelor probabilităților extrinseci este egal cu numărul ieșirilor din fiecare nod al trellisului deoarece fiecare ieșire corespunde unei anumite, posibile, valori ale cuvântului de informație.

După efectuarea iterațiilor necesare, cuvântul estimat \hat{u} al secvenței transmise poate fi determinat prin căutarea, pentru fiecare n , a celei mai mari valori APP dată de unul dintre cele două decodoare:

$$\hat{u}^n = \max_{d \in \Delta} L_j^{n,i}(d). \quad (6.10)$$

În cazul **decodării per simbol**, APP-urile și probabilitățile extrinseci sunt calculate pentru fiecare simbol u_r^n , $1 \leq r \leq R$, la fiecare moment de timp n . Deoarece un simbol corespunde la Q biți, există 2^Q valori posibile pentru estimatul lui u_r^n . Astfel, la fiecare iterație i , ambele decodoare calculează $R \cdot 2^Q$ valori pentru APP-uri și probabilitățile extrinseci. Această strategie de decodare este similară cu metoda propusă în [DOB05] pentru MBTC-uri.

În cazul **decodării per bit**, sunt posibile a fi utilizate atât APP-urile cât și rapoartele logaritmice de plauzibilitate (LLR-urile). În prima variantă, sunt calculate două valori ce corespund valorilor binare 0 sau 1, pentru fiecare bit $u_{r,q}^n$ al fiecărui

simbol u_i^n , din fiecare cuvânt u^n dintre cele N ale secvenței originale u . Astfel, ambele decodare, la fiecare iterație i , calculează $2 \cdot R \cdot Q$ valori pentru APP-uri și probabilitățile extrinseci. Alternativ, se pot calcula LLR-uri pentru cei $R \cdot Q$ biți din u^n ca și în [RHV97]. Algoritmul de decodare devine același cu cel utilizat în cazul BTC-lor.

Decodarea per cuvânt prezintă cea mai mare complexitate computațională prin comparație cu celelalte două metode de decodare, per simbol și per bit, deoarece complexitatea decodării per cuvânt crește exponențial cu ordinul câmpului Galois Q și cu numărul de intrări R în vreme ce, în cazurile decodării per simbol și respectiv per bit, complexitatea crește liniar cu R și exponențial, respectiv liniar cu Q . Însă, complexitatea mărită a decodării per cuvânt este cea care furnizează o mai bună performanță.

6.4 Închiderea trellisurilor MNBTC-lor

După descrierea manipulărilor probabilităților pe care le execută decodarea în continuare vom detalia problema închiderii trellisului. Închiderea trellisului MNBTC-ului este mult mai complicată decât pentru NBC deoarece trellisul este mult mai complicat. În acest paragraf sunt prezentate noi metode de terminație a trellisurilor pentru MNBTC-uri.

Pentru o secvență de intrare dată, $[u^0 u^1 \dots u^{N-1}]$, starea finală a trellisului, s^N , este:

$$s^N = \sum_{j=0}^{N-1} T^{N-j-1} \cdot G_T \cdot u^j + T^N \cdot s^0. \quad (6.11)$$

Închiderea trellisului poate fi făcută în două moduri: i) făcând trellisul *circular*; ii) prin *închidere la zero* ca în [WBRC98], [DOB05].

Trellisul este *circular* dacă $s^N = s^0$. Utilizând această egalitate în (6.11), obținem:

$$(I_M + T^N) \cdot s^0 = \sum_{j=0}^{N-1} T^{N-j-1} \cdot G_T \cdot u^j = s^x. \quad (6.12)$$

Dacă s^x este calculată printr-o prealabilă „precodare”, starea inițială corespunzătoare poate fi determinată astfel:

$$s^0 = (I_M + T^N)^{-1} \cdot s^x, \quad (6.13)$$

cu contrângerea ca N să nu fie multiplu de p , unde p este perioada lui T definită ca și cel mai mic întreg pentru care $T^p = I_M$. Datorită caracterului finit al câmpului, indiferent de M memoria codorului, există un astfel de p . Deoarece T conține M coeficienți din $GF(2^Q)$, rezultă că $p \leq 2^{M \cdot Q}$. Evident, dacă N este multiplu de p atunci $I_M + T^N = 0$ iar (6.13) nu poate fi satisfăcută pentru orice s^x , adică pentru orice date de intrare.

În cazul *închiderii trellisului la zero (zero padding)*, avem că $s^N = 0$ iar (6.11) devine:

$$\sum_{j=0}^{N-1} T^{N-j-1} \cdot G_T \cdot u^j = 0_{M \times 1}. \quad (6.14)$$

Rămâne să determinăm numărul de necunoscute $u^j \in GF(2^q)$ din (6.14) necesar a fi calculate pentru a închide trellisul. Vom arăta în continuare că sunt necesare și suficiente M necunoscute pentru a închide trellisul. Descompunând M ca:

$$M = a \cdot R + b, \quad a, b \text{ întregi}, \quad 0 \leq b < R, \quad (6.15)$$

atunci, (6.14) devine:

$$\begin{bmatrix} G_T & T^{a-1} \cdot G_T & T^a \cdot G_T \end{bmatrix} \cdot \begin{bmatrix} u^{N-1} \\ \dots \\ u^{N-a} \\ u^{N-a-1} \end{bmatrix} = s^{N-a-1} = \sum_{j=0}^{N-a-2} T^{N-j-1} \cdot G_T \cdot u^j. \quad (6.16)$$

Deoarece matricele $T^k \cdot G_T$, $0 \leq k < p$, sunt de rang maxim, putem selecta un set de indici $\{i_1, i_2, \dots, i_b\} \subset \{0, 1, \dots, R\}$ astfel încât matricea trunchiată $(T^a \cdot G_T)^b$ formată de coloanele $\{i_1, i_2, \dots, i_b\}$ matricei $T^a \cdot G_T$ este de rang maxim. Ținând cont că matricea T este periodică de perioadă $p > a$, matricea $A = [G_T \dots T^{p-1} \cdot G_T \dots (T^a \cdot G_T)^b]$ este inversabilă. Putem astfel să introducem $a+1$ simboluri redundante u^j cu scopul de a închide trellisul:

$$\begin{bmatrix} u^{N-1} \\ \dots \\ u^{N-a} \\ (u^{N-a-1})^b \end{bmatrix} = A^{-1} \cdot s^{N-a-1} + (T^a \cdot G_T)^{R-b} \cdot (u^{N-a-1})^{R-b}, \quad (6.17)$$

unde fragmentul de cuvânt $(u^{N-a-1})^b$ corespunde componentelor $\{i_1, i_2, \dots, i_b\}$ cuvântului u^{N-a-1} . $(u^{N-a-1})^{R-b}$ este fragmentul de cuvânt din u^{N-a-1} construit din cele $(R-b)$ componente rămase. Fragmentul $(u^{N-a-1})^{R-b}$ poate fi setat la zero pentru simplitate sau poate fi dedicat unor biți de informație, pentru a mări rata de codare.

6.5 Turbo-cod Reed-Solomon

Am văzut în secțiunile anterioare că familia MBTC-lor include pe cea a MBTC-lor care la rândul ei include TC-le clasice. În această secțiune vom vedea că această nouă familie de turbo-coduri include, de asemenea, o foarte interesantă familie de turbo-coduri pe care le vom numi RSTC deoarece conțin, ca și coduri constituente, coduri Reed-Solomon.

Propoziția 6.1: Un cod MNB este echivalent cu un cod RS dacă următoarele condiții sunt satisfăcute:

- 1) polinomul $g_0(D)$ corespunde polinomului generator al codului RS dorit [RES60]. Implicit, rezultă $M = \text{grad}(g_0)$;

- 2) Codorul are o singură intrare: $R = 1$ și $g_1(D) = 1$. Astfel, fiecare cuvânt corespunde unui singur simbol;
- 3) trellisul este închis la zero. Nu este generată secvență de cuvinte (simboluri) de control. Redundanța este constituită doar de simbolurile ce servesc închiderii trellisului din (6.17) [12].
- 4) lungimea secvenței de simboluri, care este egală cu $N = 2^q - 1$, trebuie să fie totodată egală cu lungimea cuvântului de cod RS.

Turbo-codul RS va genera $2 \cdot M$ cuvinte (simboluri) redundante (câte M de la fiecare cod constituent) pentru $k = N - M$ simboluri de intrare. Dacă cele patru condiții anterioare sunt satisfăcute, atunci turbo-codul RS poate fi privit ca și un caz special de MNBTC.

6.6 Performanțele MNBTC-lor

În Figura 6.4 sunt prezentate performanțele BER(SNR), respectiv FER(SNR) ale MNBTC-lui de memorie 3 cu două intrări, definit prin matricea generatoare $G = [2 \ 1 \ 2; 3 \ 2 \ 2; 0 \ 0 \ 2; 3 \ 1 \ 1]$. Blocul de date este compus din $N = 376$ cuvinte, fiecare cuvânt conține două simboluri iar fiecare simbol doi biți. Algoritmul de decodare utilizat a fost varianta MaxLogMAP a algoritmului MAP cu decodare per cuvânt descris în paragraful 6.3.

Întreșeserea intersimbol a fost realizată cu un interleaver de tip S , cu $S = 21$ [DOD95]. De asemenea, s-a utilizat o întreșesere intrasimbol, care permută simbolurile cuvintelor cu număr de ordine par. Am considerat un canal de transmisie AWGN și o modulație BPSK. Trellisul primului cod component a fost închis la starea nulă, în vreme ce trellisul celui de-al doilea cod component nu a fost terminat. S-a utilizat criteriul de oprire descris în paragraful 5.6. Numărul maxim de iterații a fost ales 15.

Performanțele prezentate în Figura 6.4 sunt mai slabe decât ale MBTC-lui de memorie 4 propus în [DOB05]. Aceasta este o consecință a faptului că nu s-a făcut (deocamdată) o optimizare/căutare asupra mulțimii codurilor componente și asupra interleaverelor, asemeni cazului multi-binar. Cu toate acestea, MNBTC-le prezintă câteva avantaje intrinseci: o cădere abruptă a curbelor BER/FER (fapt ce sugerează că MNBTC-le au o distanță minimă de cod superioară MBTC-lor), un efect „error-floor” foarte scăzut chiar și pentru lungimi de cod relativ mici (pentru un FER sub 10^{-4}) și o convergență foarte bună. Astfel, la același SNR, numărul mediu de iterații efectuate de MNBTC a fost mai mic decât pentru MBTC-ul propus în [DOB05] și având de asemenea memorie 4). Aceste observații ne încurajează să continuăm studiul MNBTC-lor și proiectarea lor printr-o alegere optimă a codurilor componente și a metodei de întreșesere.

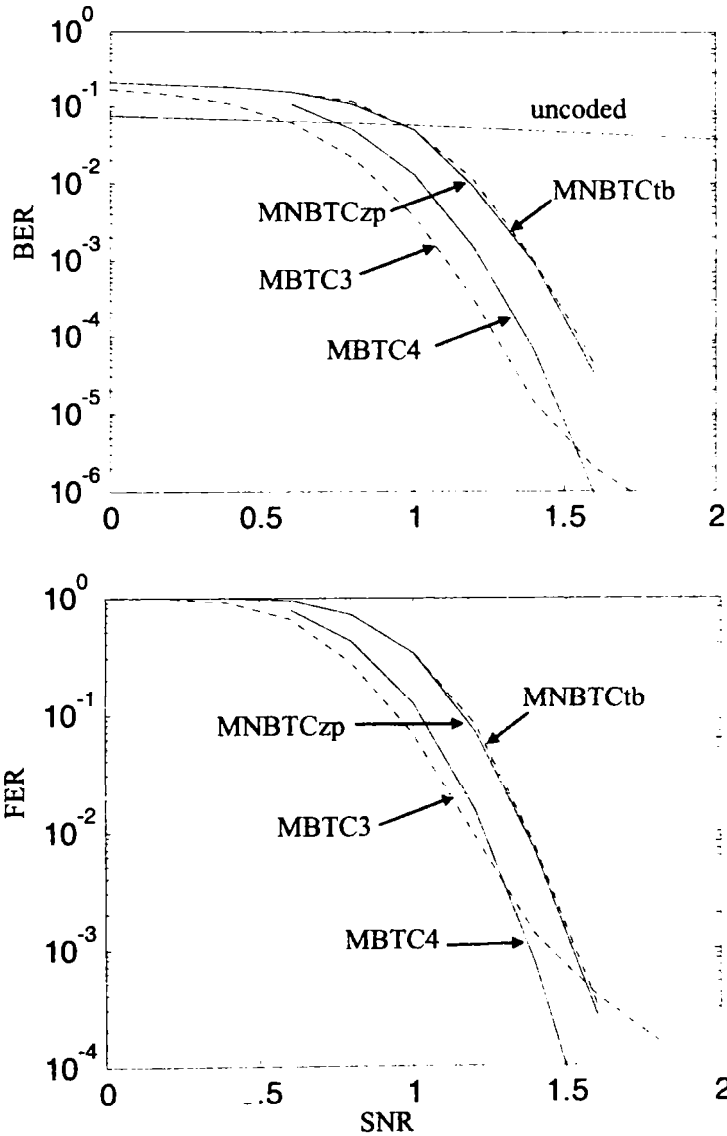


Figura 6.4. Comparație între MNBTC-ul cu matricea generatoare $G=[2 \ 1 \ 2; 3 \ 2 \ 2; 0 \ 0 \ 2; 3 \ 1 \ 1]$ (tb: tail biting, zp: zero padding) și MBTC-urile (de memorie 3 și 4, notate MBTC3, respectiv MBTC4) propuse în [DOB05] peste un canal AWGN

7. Contribuții originale

În acest capitol sunt prezentate succint contribuțiile originale cuprinse în lucrarea de față.

- În paragraful 3.4 este propusă **o nouă metodă de generare a numerelor aleatoare cu distribuție Nakagami**. Metoda se bazează pe schimbarea de variabilă dată de Ecuația (3.21) și pe inversarea pe cale numerică a funcției dată de (3.23). Această nouă metodă a permis, în premieră, simularea funcționării TC-lor în canalele (radio) cu fading Nakagami. Metoda este foarte precisă, relativ simplă, însă necesită circa 4 pași algoritmici pentru fiecare număr generat;
- În capitolul 4 am făcut o **analiză exhaustivă și o verificare**, prin simulări proprii, a influenței tuturor parametrilor **asupra performanțele unui TC**: codul component (§4.2), configurația TC-lui (§4.3), metoda de întrețesere și lungimea de întrețesere (§4.4), algoritmul de decodare (§4.5), metoda de închidere a trellisului (§4.6), metoda de puncturare și rata de codare (§4.7), numărul de iterații și criteriul de oprire utilizat (§4.8). De asemenea, am investigat modul în care estimarea canalului (mai precis a SNR-ului canalului) influențează performanța TC-lui (§4.5.1). Această analiză mi-a permis, pe de o parte, să reconstitui multe dintre rezultatele afirmate în literatură însă mai detaliat, iar pe de altă parte, să pun în evidență unele aspecte neelucidate (influența estimării canalului asupra performanței TC-lui, spre exemplu);
- În capitolul 4, subparagraful 4.4.3 am **definit** noțiunea de **spectru al distanțelor de întrețesere** (IDS). IDS-ul este un instrument pentru analiza proprietăților intrinseci ale unui interleaver. În Figura 4.13 sunt date IDS-le pentru câteva tipuri de interleavere. De asemenea, IDS-ul poate fi util în proiectarea TC-lor. Pe baza IDS-ului am **definit gradul de împrăștiere** care, alături de distanța minimă de întrețesere, sunt principalii indicatori pentru „calitatea” intrinsecă a unui interleaver.
- În subparagraful 4.4.4 s-a **calculat spectrul teoretic al distanțelor de întrețesere pentru un interleaver pur aleator**. Acest spectru, prezentat în Figura 4.15, poate fi privit ca o limită a IDS-lor prezentate în Figura 4.13. Gradul de împrăștiere a unui interleaver aleator (sau aleator de tip S), așa cum o dovedește Tabelul 4.8, este cel mai mare peste toate tipurile de interleavere. Acest fapt este o explicație pentru performanțele foarte bune ale interleaverelor aleatoare (de tip S) în TC-uri. Interleaverile de tip S vor putea fi depășite în performanță (a TC-lui) doar de acele interleavere care sunt dedicate unui anumit cod. Ca și familie de interleavere însă vor rămâne probabil cea mai performantă (în sensul că orice S interleaver va genera performanțe bune pentru orice cod, comparativ cu interleavere de alte tipuri).
- În subparagraful 4.4.6 s-a **calculat IDS-ul pentru un interleaver bloc-pătrat**. Deși are cea mai mare distanță minimă de întrețesere, interleaverul bloc sau

rectangular are cel mai mic grad de împrăștiere lucru care explică performanțele sale slabe;

- În paragraful 4.6 se prezintă o **analiză a influenței metodelor de închidere a trellisului codurilor componente ale unui TC asupra performanțelor acestuia**. Sunt investigate, de asemenea, și posibilitățile de inițializare a coeficienților recurențelor „înainte” și „înapoi” din algoritmul MAP. Simulările au arătat că metoda optimă este de a închide trellisul primului cod și de a inițializa echiprobabil coeficienții beta pentru cel de-al doilea cod;
- În paragraful 5.2 sunt **comparate** cele **două forme canonice de bază ale codoarelor RSC** (codurilor convoluționale recursive și sistematice) de rată $R/(R+1)$, unde R reprezintă numărul de intrări al codorului. Calculele matematice făcute arată **superioritatea formei canonice G** (observator). Mulțimea codoarelor generată cu această formă, pentru cazul $R>1$, este reprezentativă pentru un M –memoria codorului– și un R date, în sensul că generează toate codoarele ce pot fi generate. În plus prin forma canonică G se pot genera codoare ce nu pot fi generate cu H;
- În paragraful 5.6 se prezintă un **nou criteriu de oprire a iterațiilor pentru MBTC-uri**. Acest criteriu, bazat pe distribuția secvenței APP (A Posteriori Probability), este prezentat în două forme adaptate algoritmilor de decodare MAP și MaxLogMAP. Simulările arată că utilizarea criteriului de stop nu diminuează performanța MBTC-lui, dar reduce considerabil numărul de iterații efectuate și, implicit, timpul de decodare;
- În paragraful 5.7 se propune un **nou criteriu de căutare a codurilor componente și a interleaverelor** în scopul construcției (proiectării) MBTC-lor. Acest criteriu are la bază convergența TC-lui definită de relația (5.49);
- Utilizând rezultatele prezentate în paragraful 5.2 (superioritatea formei canonice G) și criteriul de căutare bazat pe convergența TC-lor, în subparagraful 5.7.3 sunt propuse **două DBTC-uri noi de memorie 3, ale căror performanțe FER** (prezentate în Figura 5.15) **sunt superioare codului propus în [DOB05] cu aceeași memorie**. Astfel, la un $FER=10^{-4}$, câștigul de codare este de aproximativ 0,2 dB în favoarea codurilor propuse în lucrarea de față, în comparație cu cel propus în [DOB05];
- În Capitolul 6 se propune o **nouă familie de turbo-coduri**. Codurile componente acestor MNBTC-uri prezintă R intrări non-binare. Noua clasă include MBTC-le, care la rândul lor includ BTC-le și de asemenea turbo-codurile Reed-Solomon. Capitolul 6 propune modul de construcție al acestor MNBTC-uri, adaptarea algoritmului MAP pentru decodarea lor, modurile de închidere precum și performanțele obținute prin aceste TC-uri. Deși deocamdată nu s-au obținut performanțe superioare MBTC-lor, totuși câteva indicii (spre exemplu convergența mai rapidă a MNBTC-lor) încurajează căutarea unor coduri MNB și a interleaverelor dedicate acestor coduri;
- În paragraful 6.5 sunt date condiții impuse asupra parametrilor MNBTC-lor pentru a se genera RSTC –**turbo-coduri având ca și coduri componente coduri Reed-Solomon**.

Anexa A

Performanțele BER ale BTC-lor

Tabelul A.1 Performanțele BER ale BTC-lor obținute cu interleaverul pseudo-aleator CCSDS

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate când este egal cu 0.

	1	3	5	7	11	13	15	17
1	0	2954035	3021300	1935725	2864349	1188340	1177130	1093049
3	3612356	0	0	1358637	0	527925	498583	0
5	4516335	0	0	100348	0	61305	62825	0
7	1262331	919632	5710	0	0	5163	6663	378923
11	4684497	0	0	0	0	37148	37295	0
13	4684497	33594	6755	7043	1720	0	312	292
15	108671	34188	9633	6457	1294	235	0	403
17	445882	0	0	230633	0	9302	9040	0
21	4886023	0	0	13579	0	11594	11975	0
23	178171	20128	1102	521	2061	1629	230	691
25	1420029	5267	966445	0	0	751	1159	926
27	21498	0	0	6201	0	12060	0	0
31	429482	6548	1855	600	2888	418	1823	344
33	10949	0	0	0	0	2602	2471	0
35	10609	0	0	6446	0	0	9675	0
37	135016	123269	1379	114297	1445	236	449	104996
41	4107943	0	0	2430	0	1991	1651	0
43	424030	8621	7204	0	0	456	0	2488
45	299338	13935	5521	1638	3316	901	555	6442
47	38916	0	0	10547	0	0	3771	0
51	205797	7216	6101	1502	3468	678	630	2024
53	9846	0	0	599	0	2125	6769	0
55	16898	0	0	0	0	3796	3708	0
57	5160	1513	2592	2879	913	748	4673	879
61	510340	12737	2421	0	0	0	607	721
63	125731	0	0	38990	0	29225	38412	0
65	28245	0	0	698	0	2550	3709	0
67	2075	8583	2910	2383	8190	355	2117	607
71	21641	0	0	7941	0	18221	0	0
73	2587	6523	3030	4425	12835	5061	471	313
75	9336	4653	11238	1249	968	12662	961	2685
77	94043	0	0	0	0	1404	2169	0

Tabelul 4.3 (continuare) Performanțele BER ale BTC-lor obținute cu interleaverul pseudo-aleator CCSDS

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
 Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate
 când este egal cu 0.

	21	23	25	27	31	33	35	37
1	3157698	1073326	1821748	710482	1242932	585061	560538	704876
3	0	243060	156083	0	318487	0	0	392376
5	0	5311	1471412	0	5259	0	0	20042
7	1419	1445	0	8216	1916	0	5840	289992
11	0	47924	0	0	37591	0	0	437
13	220	3144	243	8839	265	1192	0	155
15	204	579	313	0	2910	1077	7302	229
17	0	8749	1741	0	10009	0	0	249017
21	0	176919	159130	0	173697	0	0	8037
23	6165	0	578	476	1032	1283	3005	586
25	86082	34388	0	3327	1676	0	1904	1999
27	0	345	1105	0	373	0	0	546
31	10420	3162	207	828	0	776	534	1443
33	0	417	0	0	450	0	0	1374
35	0	241	558	0	368	0	0	270
37	956	757	10480	1418	737	901	731	0
41	0	2242	2031	0	1643	0	0	0
43	6217	551	0	0	2140	0	6606	3117
45	1873	12670	4607	854	3971	5999	4711	13557
47	0	1679	9612	0	13089	0	0	6179
51	1006	3910	5735	3119	17711	5515	3467	15763
53	0	962	681	0	0	0	0	1890
55	0	1940	0	0	13015	0	0	16663
57	4271	1351	1768	8903	30764	20257	3587	2084
61	6592	4919	0	18487	744	0	0	5890
63	0	39576	43539	0	36760	0	0	38388
65	0	0	2436	0	2649	0	0	1497
67	22152	26281	20200	1009	2018	1191	3883	11581
71	0	4869	8983	0	945	0	0	2437
73	114070	997	16231	1540	12697	977	424	3386
75	6851	104145	2325	7577	781	16290	1835	450
77	0	1751	0	0	2123	0	0	73523

Tabelul 4.3 (continuare) Performanțele BER ale BTC-lor obținute cu interleaveul pseudo-aleator CCSDS

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate
când este egal cu 0.

	41	43	45	47	51	53	55	57
1	2558092	1291785	1193479	523168	1207492	480732	644618	277521
3	0	145046	50884	0	41916	0	0	31688
5	0	6347	50865	0	52682	0	0	20946
7	348	0	262	13007	216	329	0	737
11	0	0	43879	0	44683	0	0	401
13	7029	416	199	0	250	1679	1008	195
15	1396	0	198	122	151	482	1639	499
17	0	8663	1597	0	5878	0	0	8822
21	0	155602	9816	0	9225	0	0	1372
23	1917	254	616	140	17637	261	604	368
25	1267	0	865	15090	359	2107	0	2429
27	0	0	586	0	1273	0	0	4623
31	6477	3924	21692	5640	332	0	490	27704
33	0	0	13682	0	16959	0	0	6718
35	0	12497	820	0	265	0	0	15250
37	0	617	6125	912	13870	3399	21406	885
41	0	52386	51113	0	54221	0	0	3930
43	12883	0	7156	5074	4128	6676	0	2928
45	7550	7623	0	4591	23549	16302	4950	16167
47	0	29280	13511	0	24500	0	0	13531
51	8407	4568	39701	11210	0	525	12873	24937
53	0	11421	16531	0	4511	0	0	23816
55	0	0	8543	0	6700	0	0	21371
57	17131	27777	26594	1752	14839	10324	8576	0
61	10064	0	5965	0	2505	23247	0	35109
63	0	7429	35644	0	29862	0	0	19276
65	0	113080	5937	0	20680	0	0	14321
67	12965	25181	13423	15669	12916	7387	7400	13738
71	0	0	15047	0	13372	0	0	14385
73	12240	1979	40387	16590	24624	14869	5216	19117
75	15321	20015	23252	40156	9457	106484	6020	112287
77	0	0	33231	0	26997	0	0	1144

Tabelul 4.3 (continuare) Performanțele BER ale BTC-lor obținute cu interleaverul pseudo-aleator CCSDS

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
 Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate
 când este egal cu 0.

	61	63	65	67	71	73	75	77
1	1286689	589732	508488	246833	586229	251741	312032	409436
3	119502	0	0	137957	0	139590	32654	0
5	5037	0	0	1560	0	1861	23048	0
7	0	257	363	18076	11557	14284	351	0
11	0	0	0	416	0	445	662	0
13	0	392	242	153	215	1689	265	424
15	370	282	1975	2524	0	466	235	359
17	9446	0	0	8446	0	8800	9174	0
21	159927	0	0	14240	0	12664	3965	0
23	3818	1717	0	1465	13100	471	14498	2881
25	0	4654	2191	17575	15184	9580	1931	0
27	23549	0	0	297	0	585	13015	0
31	250	1406	185	379	253	3520	365	14592
33	0	0	0	1994	0	2053	2180	0
35	0	0	0	547	0	251	3434	0
37	723	229	5265	1037	837	972	909	90250
41	44754	0	0	74738	0	11696	4972	0
43	0	8412	18832	18168	0	14370	14586	0
45	4498	37402	2142	19979	28922	17354	20076	24917
47	0	0	0	13803	0	16293	113221	0
51	3149	14515	129624	24140	3039	15473	27847	69808
53	149866	0	0	13064	0	9364	7155	0
55	0	0	0	11158	0	14395	11214	0
57	23236	15364	10750	15603	6537	6747	23012	6439
61	0	3933	3802	9639	3637	89620	14001	0
63	19300	0	0	6196	0	14240	18842	0
65	7637	0	0	19730	0	14989	4491	0
67	3109	13093	9074	0	6612	2022	20297	30407
71	5929	0	0	14642	0	12603	7323	0
73	35211	7683	12892	3100	4207	0	6536	93671
75	78940	18715	2690	26072	2081	9212	0	18490
77	0	0	0	1291	0	1185	1035	0

Tabelul 4.4 Performanțele BER ale BTC-lor obținute cu interleaverul aleator de tip S
($S=29$)

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate
când este egal cu 0.

	1	3	5	7	11	13	15	17
1	0	2782307	2943825	1786715	3066143	1207492	1339953	1123232
3	3713565	0	0	1513452	0	480592	586229	0
5	4532351	0	0	38628	0	54530	45274	0
7	1040109	1022982	3436	0	0	2994	2956	305910
11	3979820	0	0	0	0	36253	41956	0
13	3979820	24035	5447	4684	3999	0	143	242
15	118335	36556	7559	4703	1166	267	0	185
17	394618	0	0	162101	0	332	365	0
21	4155989	0	0	594	0	781	830	0
23	186127	5812	539	658	1333	752	144	333
25	1198150	2598	1069026	0	0	539	162	640
27	7770	0	0	2034	0	1874	0	0
31	180019	6177	1006	568	2092	220	939	501
33	5135	0	0	0	0	1273	1560	0
35	9702	0	0	2503	0	0	3214	0
37	102566	114577	322	76369	275	385	317	81253
41	4212043	0	0	1934	0	1139	824	0
43	186387	5332	2908	0	0	1305	0	2480
45	359028	10573	6894	2182	7325	413	737	2858
47	9265	0	0	13877	0	0	2067	0
51	228619	14099	4668	2885	2697	1039	1665	9634
53	12047	0	0	1625	0	8100	2027	0
55	29230	0	0	0	0	4696	6396	0
57	8908	4020	1333	2419	612	1843	19325	949
61	227980	4904	2968	0	0	0	846	703
63	10090	0	0	263	0	417	384	0
65	8205	0	0	817	0	2784	2383	0
67	3536	4241	1694	2789	12315	190	2025	287
71	6549	0	0	2588	0	2093	0	0
73	3793	12398	1442	43816	15414	2714	672	788
75	12376	2481	3854	2155	1844	10375	687	377
77	70523	0	0	0	0	2842	2491	0

Tabelul 4.4 (continuare) Performanțele BER ale BTC-lor obținute cu interleaveverul aleator de tip S ($S=29$)

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate când este egal cu 0.

	21	23	25	27	31	33	35	37
1	2853139	1250243	1811238	611821	1159192	575409	691787	657002
3	0	188714	158108	0	235426	0	0	515695
5	0	2482	1401345	0	3080	0	0	11095
7	832	817	0	6151	850	0	4025	241462
11	0	41470	0	0	35644	0	0	477
13	167	763	185	1311	479	182	0	100
15	266	738	145	0	707	401	1421	228
17	0	302	3075	0	307	0	0	193675
21	0	42699	46773	0	48925	0	0	10501
23	24686	0	313	743	6509	1428	2622	794
25	2893	4516	0	731	2887	0	908	758
27	0	258	732	0	506	0	0	432
31	5303	5826	437	1721	0	945	439	1285
33	0	302	0	0	252	0	0	1161
35	0	460	487	0	439	0	0	324
37	543	397	4446	565	421	570	404	0
41	0	639	1355	0	903	0	0	0
43	9810	550	0	0	1105	0	15923	6397
45	1180	2040	3133	2808	4826	15890	3769	14232
47	0	992	9972	0	7480	0	0	9044
51	1735	8191	3568	3668	8333	13754	1056	13976
53	0	777	1633	0	0	0	0	2386
55	0	3125	0	0	1994	0	0	6292
57	4723	739	16047	1396	71654	16720	4934	4592
61	6697	2943	0	10200	291	0	0	3813
63	0	307	167	0	190	0	0	387
65	0	0	1867	0	2581	0	0	1138
67	28519	130035	26804	1066	648	1183	1525	2540
71	0	2634	2889	0	1310	0	0	24800
73	33805	1276	113225	12719	19350	2428	714	3802
75	11232	19005	8164	5724	329	15031	934	2206
77	0	5198	0	0	3437	0	0	48073

Tabelul 4.4 (continuare) Performanțele BER ale BTC-lor obținute cu interleaverul aleator de tip S ($S=29$)

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate când este egal cu 0.

	41	43	45	47	51	53	55	57
1	2993273	1207492	1223435	556141	1188807	549547	717488	262103
3	0	131135	31141	0	28824	0	0	39378
5	0	3172	41831	0	40679	0	0	9642
7	276	0	127	9802	74	197	0	373
11	0	0	41854	0	49803	0	0	508
13	1649	83	80	0	61	386	473	127
15	1299	0	286	94	73	1313	499	208
17	0	333	532	0	533	0	0	335
21	0	45040	1133	0	588	0	0	1887
23	11315	233	1940	136	25743	437	1122	391
25	1331	0	538	13856	204	1010	0	1071
27	0	0	202	0	752	0	0	982
31	13931	2866	13021	16983	425	0	600	69932
33	0	0	7457	0	14444	0	0	17526
35	0	23136	258	0	336	0	0	7477
37	0	731	5433	390	22609	3707	14937	376
41	0	36471	47788	0	40759	0	0	30444
43	9174	0	2550	2094	1518	4200	0	4059
45	10600	14247	0	23325	26952	22341	4008	22238
47	0	2094	5453	0	16476	0	0	3385
51	20778	1979	25027	13750	0	2564	4858	21800
53	0	28429	44149	0	3700	0	0	4375
55	0	0	16959	0	14568	0	0	5918
57	22967	36629	28408	616	8852	1526	3734	0
61	13068	0	3898	0	2653	23247	0	32727
63	0	19832	53766	0	30534	0	0	8299
65	0	65910	14435	0	26699	0	0	8136
67	61580	63321	111671	5047	66312	7881	24005	29670
71	0	0	27579	0	12537	0	0	192134
73	27606	3694	31173	17325	108604	14879	4281	17458
75	24360	10319	17105	15974	18931	12671	2340	119031
77	0	0	27526	0	27316	0	0	64

Tabelul 4.4 (continuare) Performanțele BER ale BTC-lor obținute cu interleaverul aleator de tip S ($S=29$)

Pe orizontală – polinomul de ieșire; pe verticală – polinomul de reacție;
Numărul din tabel semnifică $BER \times 10^9$ când este diferit de 0 și incompatibilitate când este egal cu 0.

	61	63	65	67	71	73	75	77
1	1134529	583417	583417	260213	494650	250240	333025	367943
3	117391	0	0	104040	0	102492	33592	0
5	3278	0	0	358	0	727	8419	0
7	0	258	125	8786	8039	11305	363	0
11	0	0	0	485	0	519	569	0
13	0	147	935	196	121	396	193	287
15	186	128	518	350	0	342	122	575
17	335	0	0	488	0	782	402	0
21	35638	0	0	1074	0	2472	2761	0
23	3591	2305	0	2710	4540	330	15135	2778
25	0	15702	878	13606	19005	9179	708	0
27	16096	0	0	520	0	249	152923	0
31	269	1091	217	286	167	13009	207	4345
33	0	0	0	919	0	1416	1690	0
35	0	0	0	218	0	335	2291	0
37	331	384	4299	1593	534	923	798	58997
41	46317	0	0	18534	0	401227	6071	0
43	0	5968	20458	20065	0	444	15899	0
45	5201	31271	3309	16184	8513	98094	24113	10032
47	0	0	0	4066	0	30534	26549	0
51	8835	52297	15758	99546	3659	14286	16219	68429
53	206608	0	0	14589	0	15480	13301	0
55	0	0	0	15839	0	923983	6148	0
57	6688	29861	39126	15876	27662	17594	32804	18244
61	0	2570	3929	3518	686	64571	4821	0
63	3021	0	0	7494	0	13151	30265	0
65	10792	0	0	33623	0	39829	3974	0
67	21168	10619	13226	0	18438	1958	19668	16879
71	7742	0	0	22430	0	20300	5443	0
73	122750	99527	2136	3568	14395	0	61241	25116
75	27355	24284	3416	22923	1731	27310	0	48372
77	0	0	0	1055	0	1720	582	0

Anexa B

MBCC-le de memorie 4 selectate după criteriul convergenței

Nr.	Codul	Nr.It/Nb				Simulare SNR=1.8dB		
		SNR=1.6 dB		SNR=1.8 dB		Nst_bl	BER *1e-9	FER *1e-6
		Căutare	Simulare	Căutare	Simulare			
0	21 23 25				3.0912	22	0	0
1	25 23 17	3.671		3.192	3.194	1381	297	206
2	29 14 17	3.624		3.189	3.197	341	93	34
3	29 19 17	3.676		3.254	3.191	1362	369	266
4	31 19 17	3.670		3.248	3.221	1054	409	256
5	31 25 17	3.679		3.231	3.221	1136	292	196
6	17 11 19	3.635		3.188	3.199	96	6	4
7	17 13 19	3.613		3.186	3.203	176	25	10
8	17 14 19	3.514		3.116	3.104	26	2	2
9	21 07 19	3.583		3.182	3.165	57	0	0
10	21 09 19	3.644		3.194	3.205	103	6	4
11	21 14 19	3.504		3.097	3.086	30	0	0
12	21 18 19	3.653		3.227	3.206	71	178	8
13	22 13 19	3.678		3.250	3.208	179	13	8
14	23 05 19	3.614		3.202	3.206	97	191	12
15	23 10 19	3.546		3.158	3.145	97	7	6
16	23 14 19	3.632		3.189	3.173	45	14	10
17	23 15 19	3.711		3.232	3.234	171	10	4
18	25 15 19	3.599		3.190	3.193	294	41	16
19	25 23 19	3.419		3.044	3.045	46	9	6
20	26 11 19	3.608		3.207	3.195	209	203	30
21	26 17 19	3.628		3.174	3.164	112	14	2
22	26 25 19	3.569		3.146	3.132	63	19	8
23	27 05 19	3.616		3.202	3.227	109	15	6
24	27 10 19	3.596		3.188	3.188	62	7	4
25	27 13 19	3.596		3.182	3.178	116	3	2
26	27 14 19	3.605		3.217	3.174	45	5	4
27	27 15 19	3.638		3.197	3.191	122	7	6
28	27 17 19	3.646		3.238	3.232	130	3	2
29	27 20 19	3.637		3.207	3.239	200	37	24
30	27 21 19	3.363		3.019	3.013	34	3	2
31	27 22 19	3.695		3.220	3.222	132	19	8
32	28 17 19	3.676		3.210	3.221	178	43	16
33	28 21 19	3.590		3.193	3.176	163	65	38
34	29 04 19	3.689		3.257	3.241	26	9	6
35	29 06 19	3.643		3.226	3.229	46	0	0
36	29 09 19	3.586		3.172	3.163	242	15	8
37	29 10 19	3.689		3.274	3.264	132	10	4

Nr.	Codul	Nr.It/Nb				Simulare SNR=1.8dB		
		SNR=1.6 dB		SNR=1.8 dB		Nst_bl	BER *1e-9	FER *1e-6
		Căutare	Simulare	Căutare	Simulare			
38	29 12 19	3.654		3.229	3.218	66	12	6
39	29 13 19	3.725		3.249	3.246	275	7	4
40	29 17 19	3.508		3.076	3.090	30	11	6
41	29 18 19	3.685		3.232	3.234	309	67	26
42	29 21 19	3.458		3.067	3.077	33	0	0
43	29 23 19	3.399		3.060	3.033	39	5	4
44	29 25 19	3.403		3.041	3.030	31	5	2
45	29 27 19	3.461		3.098	3.083	62	0	0
46	30 21 19	3.580		3.165	3.164	208	30	14
47	30 27 19	3.607		3.191	3.192	110	0	0
48	31 05 19	3.584		3.167	3.163	191	0	0
49	31 10 19	3.689		3.229	3.240	251	10	6
50	31 14 19	3.653		3.227	3.184	24	0	0
51	31 17 19	3.409		3.055	3.048	27	0	0
52	31 21 19	3.646		3.224	3.231	842	51	18
53	31 26 19	3.506		3.112	3.090	137	21	14
54	31 29 19	3.478		3.107	3.078	38	3	2
55	17 13 21	3.675		3.186	3.621	701	2119	94
56	19 14 21	3.678		3.252	3.241	244	101	6
57	19 15 21	3.646		3.229	3.213	585	63	24
58	22 17 21	3.630		3.193	3.171	240	248	48
59	23 11 21	3.658		3.202	3.198	308	101	66
60	23 12 21	3.559		3.176	3.155	91	23	18
61	25 14 21	3.671		3.260	3.242	237	29	14
62	25 19 21	3.715		3.228	3.254	961	420	150
63	25 23 21	3.440		3.047	3.045	250	79	48
64	27 06 21	3.680		3.234	3.241	35	50	10
65	27 15 21	3.660		3.212	3.225	986	465	174
66	27 19 21	3.569		3.147	3.137	938	406	146
67	27 23 21	3.359		2.996	2.989	83	39	26
68	27 25 21	3.534		3.097	3.115	615	394	142
69	27 26 21	3.619		3.191	3.198	595	462	168
70	29 06 21	3.607		3.182	3.153	124	54	40
71	29 15 21	3.519		3.123	3.112	290	127	54
72	29 19 21	3.448		3.052	3.045	28	90	40
73	29 26 21	3.592		3.189	3.196	246	103	46
74	29 27 21	3.352		3.004	2.992	106	31	24
75	30 23 21	3.503		3.105	3.105	221	74	46
76	30 25 21	3.657		3.243	3.224	700	119	40
77	30 27 21	3.640		3.223	3.205	602	417	142
78	31 13 21	3.500		3.121	3.112	307	90	42
79	31 22 21	3.500		3.109	3.109	268	111	60
80	19 10 23	3.488		3.112	3.110	68	0	0
81	21 06 23	3.679		3.234	3.230	137	268	4
82	21 12 23	3.601		3.181	3.164	68	6	4
83	25 06 23	3.623		3.195	3.188	127	9	4
84	25 10 23	3.528		3.122	3.120	46	3	2

Nr.	Codul	Nr.It/Nb				Simulare SNR=1.8dB		
		SNR=1.6 dB		SNR=1.8 dB		Nst_bl	BER *1e-9	FER *1e-6
		Căutare	Simulare	Căutare	Simulare			
85	25 12 23	3.611		3.186	3.177	64	3	2
86	25 19 23	3.487		3.117	3.074	305	15	10
87	25 21 23	3.468		3.081	3.075	243	23	16
88	27 21 23	3.606		3.181	3.171	513	35	12
89	27 25 23	3.636		3.223	3.177	366	41	18
90	29 19 23	3.586		3.180	3.154	320	14	6
91	29 25 23	3.634		3.213	3.169	405	53	14
92	31 10 23	3.628		3.209	3.193	287	31	10
93	31 25 23	3.536		3.133	3.114	458	23	16
95	17 07 25	3.695		3.240	3.222	213	21	12
96	17 11 25	3.611		3.205	3.169	132	3	2
97	17 14 25	3.523		3.121	3.105	27	11	8
98	19 11 25	3.557		3.164	3.136	74	2	2
99	20 11 25	3.696		3.252	3.258	110	31	12
100	21 07 25	3.556		3.152	3.170	89	15	12
101	21 09 25	3.665		3.232	3.215	87	19	6
102	21 14 25	3.440		3.085	3.084	18	2	2
103	21 15 25	3.544		3.142	3.157	195	22	14
104	21 18 25	3.671		3.254	3.217	161	34	20
105	22 13 25	3.642		3.203	3.202	150	26	12
106	22 17 25	3.726		3.242	3.215	221	17	8
107	23 04 25	3.690		3.261	3.245	34	17	10
108	23 06 25	3.667		3.221	3.210	37	3	2
109	23 09 25	3.675		3.258	3.240	428	18	8
110	23 10 25	3.699		3.257	3.264	213	2	2
111	23 12 25	3.669		3.254	3.226	47	131	4
112	23 15 25	3.671		3.250	3.265	251	5	2
113	23 17 25	3.477		3.077	3.089	56	15	8
114	23 18 25	3.562		3.175	3.167	174	13	8
115	23 19 25	3.390		3.032	3.030	62	0	0
116	23 21 25	3.436		3.081	3.071	46	3	2
117	23 22 25	3.693		3.232	3.238	127	9	4
118	26 11 25	3.673		3.216	3.197	210	31	24
119	26 17 25	3.612		3.200	3.209	156	5	4
120	27 05 25	3.637		3.213	3.223	178	26	16
121	27 10 25	3.644		3.206	3.189	69	2	2
122	27 13 25	3.685		3.213	3.209	86	21	8
123	27 14 25	3.586		3.184	3.169	55	7	6
124	27 15 25	3.605		3.207	3.186	99	23	10
125	27 20 25	3.616		3.227	3.231	123	3	2
126	27 21 25	3.381		3.018	3.011	28	0	0
127	27 22 25	3.648		3.207	3.179	99	5	2
128	27 23 25	3.481		3.077	3.082	60	6	4
129	28 17 25	3.706		3.228	3.252	244	29	12
130	28 21 25	3.607		3.170	3.180	72	3	2
131	29 10 25	3.538		3.147	3.151	132	17	8
132	29 14 25	3.588		3.170	3.171	43	7	4

Nr.	Codul	Nr.It/Nb				Simulare SNR=1.8dB		
		SNR=1.6 dB		SNR=1.8 dB		Nst_bl	BER *1e-9	FER *1e-6
		Căutare	Simulare	Căutare	Simulare			
133	29 19 25	3.419		3.047	3.046	55	13	8
134	29 20 25	3.706		3.248	3.228	290	11	6
135	29 23 25	3.383		3.029	3.035	52	7	4
136	30 19 25	3.668		3.207	3.189	251	27	12
137	30 27 25	3.591		3.183	3.180	124	23	10
138	30 29 25	3.648		3.215	3.217	157	26	8
139	31 10 25	3.676		3.245	3.232	211	45	18
140	31 11 25	3.480		3.109	3.088	121	42	24
141	31 14 25	3.591		3.178	3.182	16	2	2
142	31 17 25	3.422		3.055	3.048	22	0	0
143	31 20 25	3.582		3.178	3.166	78	0	0
144	31 21 25	3.644		3.221	3.207	248	27	10
145	31 23 25	3.476		3.093	3.085	40	3	2
146	21 17 27	3.588		3.171	3.164	432	31	16
147	31 21 27	3.576		3.152	3.112	555	82	36
148	19 06 29	3.577		3.174	3.174	64	14	4
149	19 10 29	3.514		3.135	3.121	58	29	6
150	19 12 29	3.627		3.198	3.190	99	6	4
151	19 17 29	3.710		3.229	3.230	903	41	18
152	21 06 29	3.629		3.167	3.161	69	2	2
153	21 12 29	3.703		3.231	3.246	345	18	10
154	21 19 29	3.514		3.109	3.080	334	21	12
155	23 19 29	3.590		3.162	3.167	339	45	14
156	25 10 29	3.536		3.153	3.113	60	3	2
157	25 19 29	3.467		3.084	3.070	222	47	32
158	25 23 29	3.594		3.176	3.155	420	43	14
159	27 19 29	3.634		3.197	3.186	468	63	18
160	27 21 29	3.580		3.183	3.164	339	19	8
161	31 10 29	3.627		3.210	3.198	292	17	8
162	31 17 29	3.637		3.197	3.201	1194	83	40
163	31 19 29	3.485		3.112	3.104	346	18	14
164	17 10 31	3.627		3.202	3.179	27	22	12
165	19 10 31	3.690		3.221	3.136	1001	13885	206
166	19 13 31	3.532		3.157	3.041	1119	7792	160
167	19 18 31	3.664		3.208	3.203	256	66	24
168	21 17 31	3.589		3.192	3.172	438	125	44
169	23 11 31	3.706		3.244	3.220	472	187	60
170	23 20 31	3.669		3.245	3.213	285	166	52
171	25 10 31	3.652		3.215	3.216	395	59	28
172	25 22 31	3.556		3.136	3.133	357	203	70
173	27 10 31	3.492		3.106	3.092	36	5	2
174	27 17 31	3.344		2.989	2.983	60	15	12
175	27 21 31	3.495		3.087	3.075	366	109	42
176	29 05 31	3.690		3.243	3.230	533	123	46
177	29 10 31	3.665		3.236	3.229	562	47	22
178	29 26 31	3.631		3.227	3.202	298	122	46

Anexa C

Calculul inversei matricii $(I_M + D \cdot T)$

Pornind de la definiția matricii de trecere T , relația (5.4), rezultă că:

$$I_M + D \cdot T = \begin{bmatrix} 1 & D & 0 & \dots & 0 & 0 \\ 0 & 1 & D & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & D \\ h_{0,M} \cdot D & h_{0,M-1} \cdot D & h_{0,M-2} \cdot D & \dots & h_{0,2} \cdot D & h_{0,1} \cdot D + 1 \end{bmatrix}. \quad (C.1)$$

Calculul matricii inverse, $(I_M + D \cdot T)^{-1}$, se va face prin metoda transformărilor liniare. Astfel, prin transformări liniare asupra liniilor matricii $(I_M + D \cdot T)$ se caută să se obțină I_M . Utilizând aceleași transformări asupra liniilor lui I_M , se va obține $(I_M + D \cdot T)^{-1}$. Transformările liniare prezentate în continuare, încearcă să obțină pe fiecare coloană din $(I_M + D \cdot T)$ câte un singur „1”. Aceleași transformări sunt imediat aplicate matricii I_M :

Pasul 1

$L_1 \times h_{0,M} \cdot D + L_M \rightarrow L_M$ -înmulțește linia „1” cu termenul $\Sigma_1 = h_{0,M} \cdot D$ și adună la linia „M”;

$L_2 \times D + L_1 \rightarrow L_1$ -înmulțește linia „2” cu D și adună la linia „1”:

$$(I_M + D \cdot T) \rightarrow \begin{bmatrix} 1 & 0 & D^2 & \dots & 0 & 0 \\ 0 & 1 & D & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & D \\ 0 & h_{0,M} \cdot D^2 + h_{0,M-1} \cdot D & h_{0,M-2} \cdot D & \dots & h_{0,2} \cdot D & h_{0,1} \cdot D + 1 \end{bmatrix}$$

$$I_M \rightarrow \begin{bmatrix} 1 & D & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ h_{0,M} \cdot D & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Pasul $m=2, 3, \dots, M-1$

$L_m \times \sum_{k=0}^{m-1} h_{0,M-k} \cdot D^{m-k} + L_M \rightarrow L_m$ -înmulțește linia „m” cu termenul

$$\Sigma_m = \sum_{k=0}^{m-1} h_{0, M-k} \cdot D^{m-k} \text{ și adună la linia „}M\text{”};$$

$L_{m+1} \times D + L_1 \rightarrow L_1$ -înmulțește linia „ $m+1$ ” cu D și adună la linia „ 1 ”:

$$(I_M + D \cdot T) \rightarrow \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & D^{M-1} \\ 0 & 1 & 0 & \dots & 0 & D^{M-2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & D \\ 0 & 0 & 0 & \dots & 0 & h_0(D) \end{bmatrix}, \text{ unde } h_0(D) = \sum_{m=1}^M h_{0,m} \cdot D^m + 1.$$

$$I_M \rightarrow \begin{bmatrix} 1 & D & \dots & D^{M-3} & D^{M-2} & 0 \\ 0 & 1 & \dots & D^{M-4} & D^{M-3} & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 1 & 0 \\ \Sigma_1 & \Sigma_2 & \dots & \Sigma_{M-2} & \Sigma_{M-1} & 1 \end{bmatrix}$$

Pasul M

$L_M / h_0(D) \rightarrow L_1$ -împarte linia „ M ” cu $h_0(D)$:

$$(I_M + D \cdot T) \rightarrow \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & D^{M-1} \\ 0 & 1 & 0 & \dots & 0 & D^{M-2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & D \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

$$I_M \rightarrow \begin{bmatrix} 1 & D & \dots & D^{M-3} & D^{M-2} & 0 \\ 0 & 1 & \dots & D^{M-4} & D^{M-3} & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 1 & 0 \\ \Sigma_1 / h_0(D) & \Sigma_2 / h_0(D) & \dots & \Sigma_{M-2} / h_0(D) & \Sigma_{M-1} / h_0(D) & 1 / h_0(D) \end{bmatrix}$$

Pasul $M+m=M+1, M+2, \dots, 2 \cdot M-1$

$L_M \times D^m + L_{M-m} \rightarrow L_{M-m}$ -înmulțește linia „ M ” cu D^m și adună la linia „ $M-m$ ”;

$$(I_M + D \cdot T) \rightarrow I_M$$

$$\begin{bmatrix} \frac{D^{M-1} \cdot \Sigma_1}{h_0(D)} + 1 & \frac{D^{M-1} \cdot \Sigma_2}{h_0(D)} + D & \dots & \frac{D^{M-1} \cdot \Sigma_{M-1}}{h_0(D)} + D^{M-2} & \frac{D^{M-1}}{h_0(D)} \\ \frac{D^{M-2} \cdot \Sigma_1}{h_0(D)} & \frac{D^{M-2} \cdot \Sigma_2}{h_0(D)} + 1 & \dots & \frac{D^{M-2} \cdot \Sigma_{M-1}}{h_0(D)} + D^{M-3} & \frac{D^{M-2}}{h_0(D)} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{D \cdot \Sigma_1}{h_0(D)} & \frac{D \cdot \Sigma_2}{h_0(D)} & \dots & \frac{D \cdot \Sigma_{M-1}}{h_0(D)} + 1 & \frac{D}{h_0(D)} \\ \frac{\Sigma_1}{h_0(D)} & \frac{\Sigma_2}{h_0(D)} & \dots & \frac{\Sigma_{M-1}}{h_0(D)} & \frac{1}{h_0(D)} \end{bmatrix} = (I_M + D \cdot T)^{-1}. \quad (C.2)$$

Forma (C.2) obținută pentru matricea $(I_M + D \cdot T)^{-1}$ poate fi simplificată utilizând notațiile:

$$P_M(D) = [D^{M-1} \dots D \ 1] \quad (C.3)$$

$$\Delta_M(D) = \begin{bmatrix} D & D^2 & \dots & D^{M-1} & D^M \\ 0 & D & \dots & D^{M-2} & D^{M-1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & D & D^2 \\ 0 & 0 & \dots & 0 & D \end{bmatrix}. \quad (C.4)$$

Astfel:

$$\begin{aligned} (I_M + D \cdot T)^{-1} &= \\ &= \begin{bmatrix} \frac{D^{M-1} \cdot \Sigma_1}{h_0(D)} & \frac{D^{M-1} \cdot \Sigma_2}{h_0(D)} & \dots & \frac{D^{M-1} \cdot \Sigma_{M-1}}{h_0(D)} & \frac{D^{M-1}}{h_0(D)} + D^{M-1} \\ \frac{D^{M-2} \cdot \Sigma_1}{h_0(D)} & \frac{D^{M-2} \cdot \Sigma_2}{h_0(D)} & \dots & \frac{D^{M-2} \cdot \Sigma_{M-1}}{h_0(D)} & \frac{D^{M-2}}{h_0(D)} + D^{M-2} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{D \cdot \Sigma_1}{h_0(D)} & \frac{D \cdot \Sigma_2}{h_0(D)} & \dots & \frac{D \cdot \Sigma_{M-1}}{h_0(D)} & \frac{D}{h_0(D)} + D \\ \frac{\Sigma_1}{h_0(D)} & \frac{\Sigma_2}{h_0(D)} & \dots & \frac{\Sigma_{M-1}}{h_0(D)} & \frac{1}{h_0(D)} + 1 \end{bmatrix} + D^{-1} \cdot \Delta_M(D) = \\ &= \frac{1}{h_0(D)} \cdot \begin{bmatrix} D^{M-1} \cdot \Sigma_1 & D^{M-1} \cdot \Sigma_2 & \dots & D^{M-1} \cdot \Sigma_{M-1} & D^{M-1} \cdot \Sigma_M \\ D^{M-2} \cdot \Sigma_1 & D^{M-2} \cdot \Sigma_2 & \dots & D^{M-2} \cdot \Sigma_{M-1} & D^{M-2} \cdot \Sigma_M \\ \dots & \dots & \dots & \dots & \dots \\ D \cdot \Sigma_1 & D \cdot \Sigma_2 & \dots & D \cdot \Sigma_{M-1} & D \cdot \Sigma_M \\ \Sigma_1 & \Sigma_2 & \dots & \Sigma_{M-1} & \Sigma_M \end{bmatrix} + D^{-1} \cdot \Delta_M(D) = \\ &= \frac{1}{h_0(D)} \cdot P_M^T(D) \cdot [\Sigma_1 \ \Sigma_2 \ \dots \ \Sigma_M] + D^{-1} \cdot \Delta_M(D), \end{aligned}$$

de unde se obține:

$$(I_M + D \cdot T)^{-1} = \frac{1}{h_0(D)} \cdot P_M^T(D) \cdot H_R \cdot \Delta_M(D) + D^{-1} \cdot \Delta_M(D), \quad (C.5)$$

relație care este identică cu (5.9).

Deoarece:

$$(I_M+D \cdot T^T)^{-1} = [(I_M+D \cdot T)^T]^{-1} = [(I_M+D \cdot T)^{-1}]^T \tag{C.6}$$

rezultă că pentru a calcula $(I_M+T^T)^{-1}$ este suficient să setăm în (C.5) pe $D=1$ și să transpunem matricea rezultată. De remarcat că există și impunerea ca $h_0(1)=1$, în caz contrar matricea (I_M+T) nu este inversabilă. Această condiție este îndeplinită de polinoamele ireductibile, mulțime ce include pe cea a polinoamelor primitive.

Se obține astfel:

$$(I_M + T^T)^{-1} = [(I_M + T)^{-1}]^T = I_{\Delta M} + \Sigma_g \tag{C.7}$$

unde:

$$I_{\Delta M} \stackrel{def}{=} \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}_{M \times M}, \tag{C.8}$$

$$\Sigma_g \stackrel{def}{=} \begin{bmatrix} \sum_{m=M}^M g_{0,m} & \sum_{m=M}^M g_{0,m} & \sum_{m=M}^M g_{0,m} \\ \sum_{m=M-1}^M g_{0,m} & \sum_{m=M-1}^M g_{0,m} & \sum_{m=M-1}^M g_{0,m} \\ \dots & \dots & \dots \\ \sum_{m=1}^M g_{0,m} = 0 & \sum_{m=1}^M g_{0,m} = 0 & \dots \sum_{m=1}^M g_{0,m} = 0 \end{bmatrix}_{M \times M} \tag{C.9}$$

Anexa D

Densitatea de probabilitate Nakagami

Diagramele din această anexă prezintă histogramme obținute pe secvențe de numere generate prin algoritmul propus în Capitolul 3 (desenate cu culoarea roșie) versus densitatea de probabilitate Nakagami (desenată cu albastru). S-au avut în vedere doar densități de probabilitate normalizate în sensul că valoarea pătratică medie a fost setată la valoarea 1. Toate graficele au fost construite din 100 de puncte (implicit și histogrammele), echidistante între 0 și 3. Peste valoarea 3, valorile tuturor densităților de probabilitate investigate au valori neglijabile.

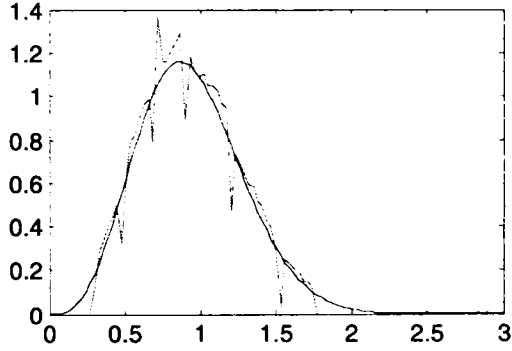
Primele 24 de histogramme (diagrame) au fost construite pentru secvențe de $N=100000$ numere, iar ultimele 16 pentru secvențe de $N=1000000$ numere. Pentru fiecare valoare a lui m din setul $\{2, 3, 4, 5\}$ s-au construit câte 6 histogramme la $N=100000$, corespunzătoare valorilor 10^{-2} , $5 \cdot 10^{-3}$, $2 \cdot 10^{-3}$, 10^{-3} , 10^{-4} și 10^{-5} pentru precizia c din algoritmul de generare și câte 4 histogramme la $N=1000000$, corespunzătoare valorilor 10^{-2} , 10^{-3} , 10^{-4} și 10^{-5} pentru precizia c .

Deasupra fiecărei diagrame au fost notate:

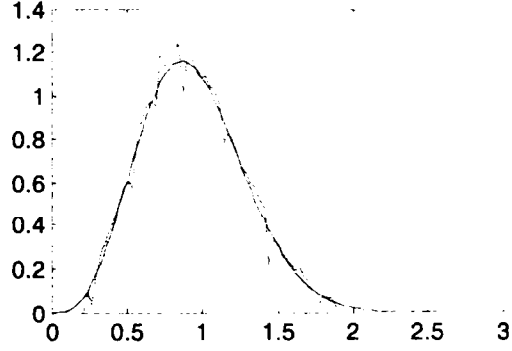
- precizia c , definită în paragraful 3.4.2;
- eroarea ε , definită prin relația (3.31);
- numărul de iterații (notat it), necesar algoritmului de generare propus pentru a atinge precizia c .

$N=100000$; $m=2$;

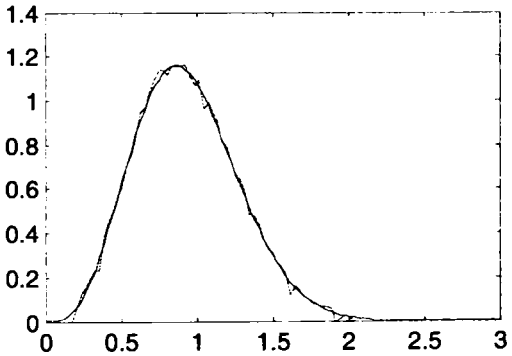
$c = 1e-2$; $\varepsilon = 2.2435$; $it = 2.81736$;



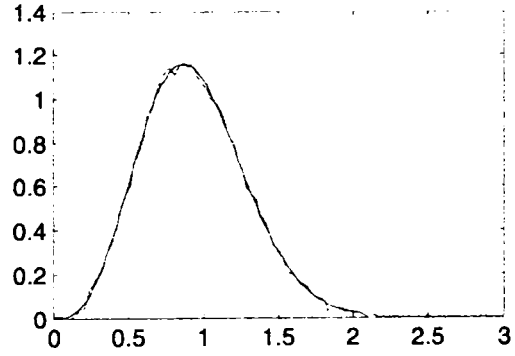
$c = 5e-3$; $\varepsilon = 0.6584$; $it = 3.02757$;



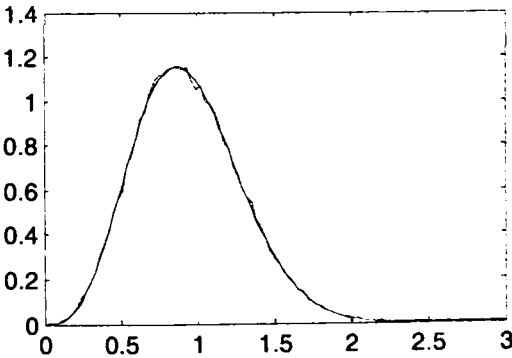
$c = 2e-3$; $\varepsilon = 0.1203$; $it = 3.27864$;



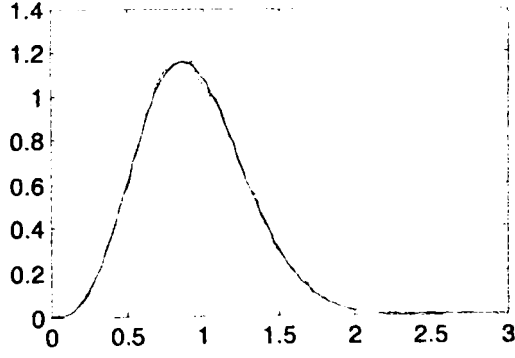
$c = 1e-3$; $\varepsilon = 0.0627$; $it = 3.44159$;

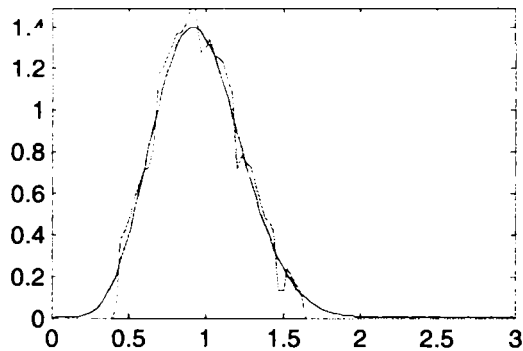
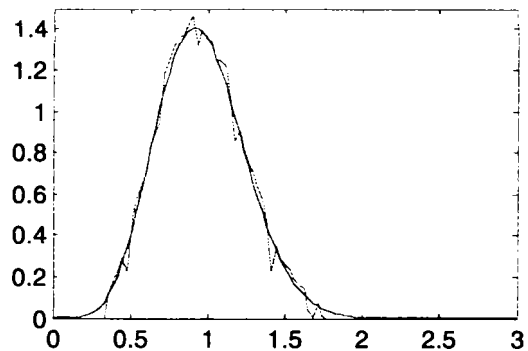
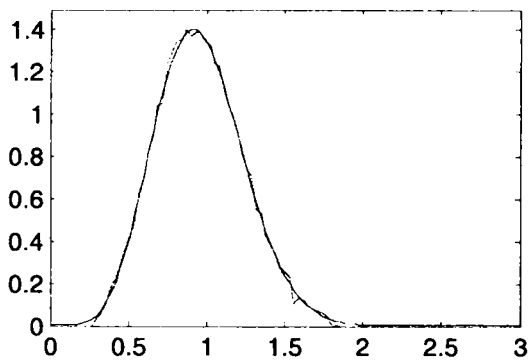
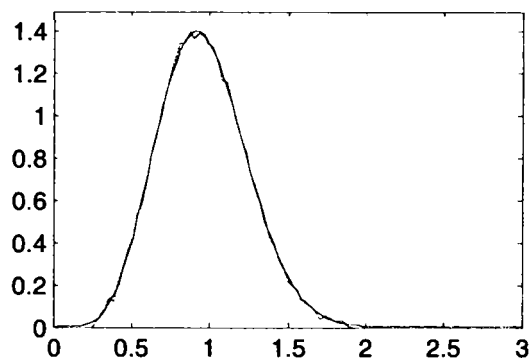
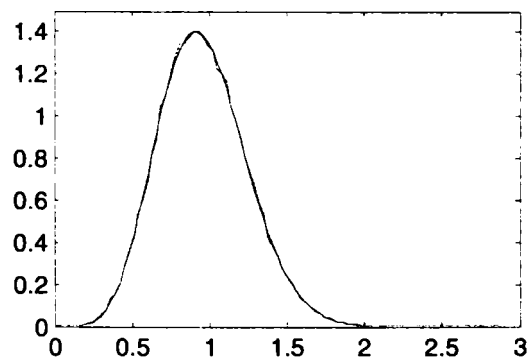
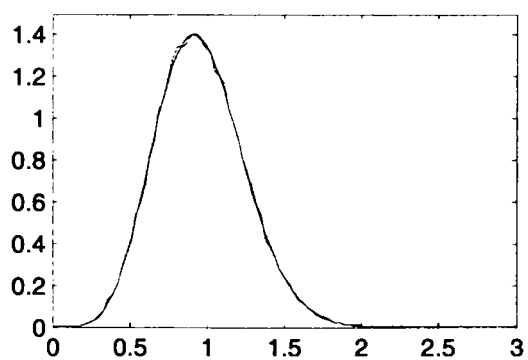


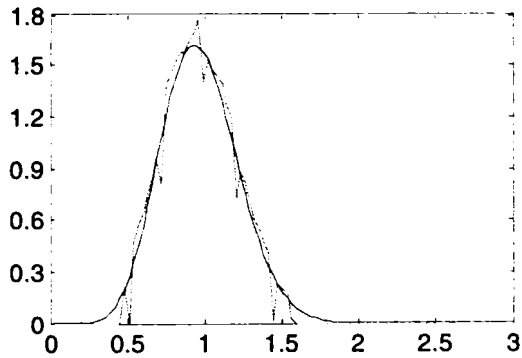
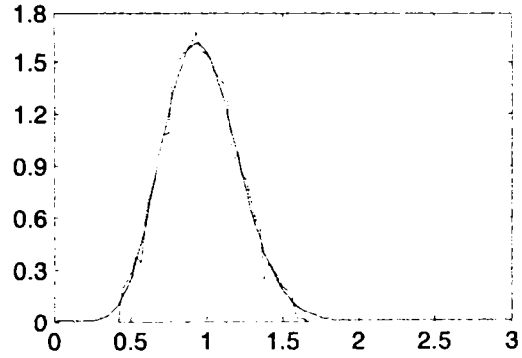
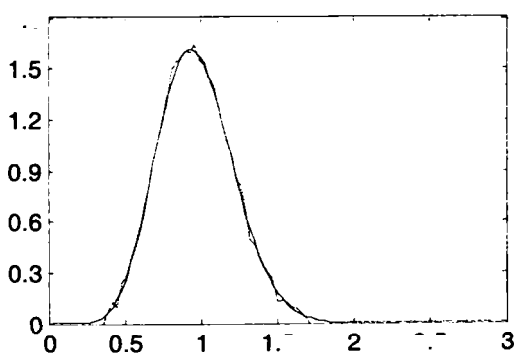
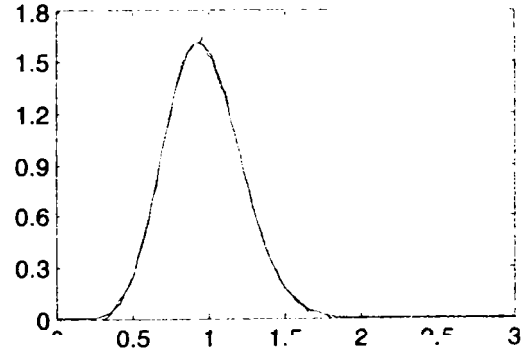
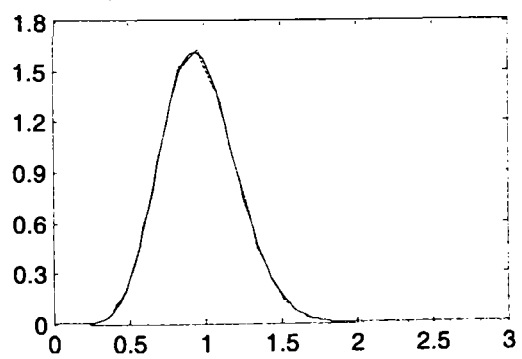
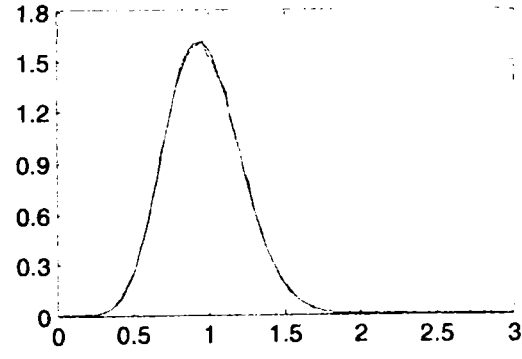
$c = 1e-4$; $\varepsilon = 0.0364$; $it = 3.88006$;

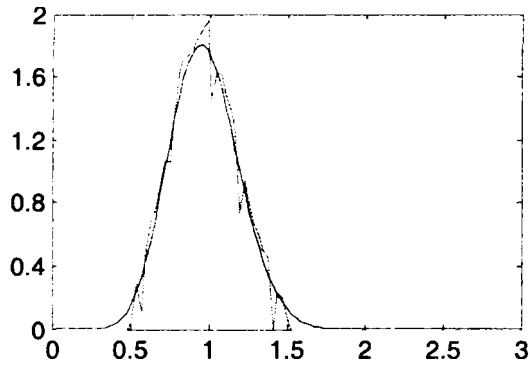
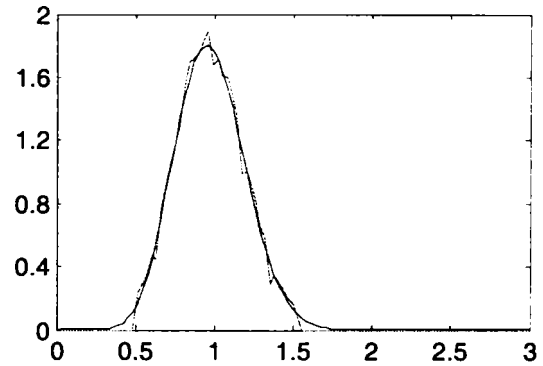
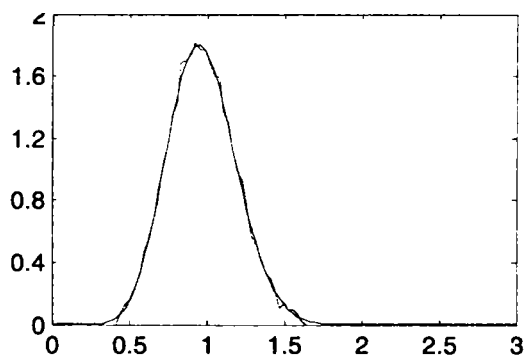
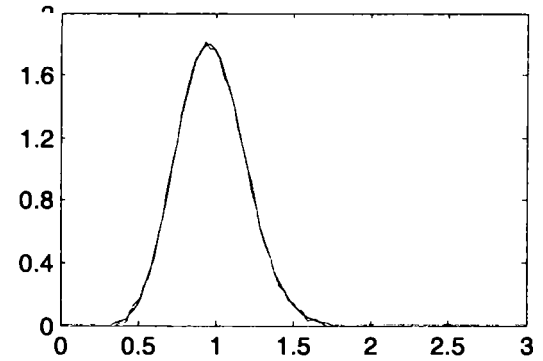
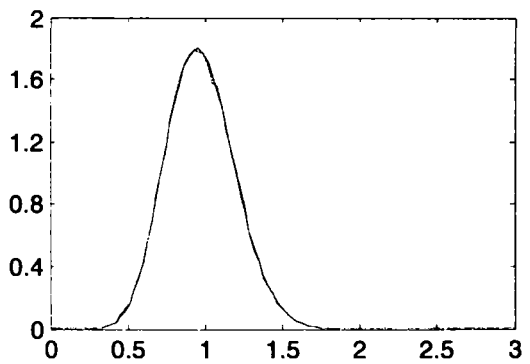
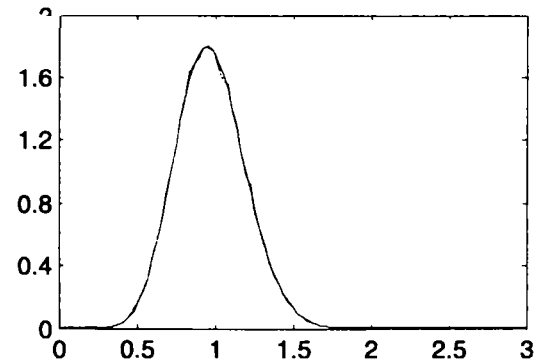


$c = 1e-5$; $\varepsilon = 0.0351$; $it = 4.21683$;



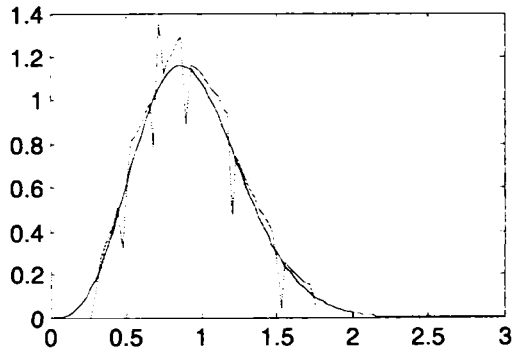
$N=100000$; $m=3$; $c = 1e-2$; $\varepsilon = 1.0857$; $it = 2.75084$; $c = 5e-3$; $\varepsilon = 0.4346$; $it = 2.97208$; $c = 2e-3$; $\varepsilon = 0.0957$; $it = 3.22220$; $c = 1e-3$; $\varepsilon = 0.0470$; $it = 3.39009$; $c = 1e-4$; $\varepsilon = 0.0229$; $it = 3.82794$; $c = 1e-5$; $\varepsilon = 0.0323$; $it = 4.16411$;

$N=100000$; $m=4$; $c = 1e-2$; $\varepsilon = 1.4745$; $it = 2.72789$; $c = 5e-3$; $\varepsilon = 0.4605$; $it = 2.94957$; $c = 2e-3$; $\varepsilon = 0.1088$; $it = 3.20147$; $c = 1e-3$; $\varepsilon = 0.0357$; $it = 3.36941$; $c = 1e-4$; $\varepsilon = 0.0310$; $it = 3.81144$; $c = 1e-5$; $\varepsilon = 0.0308$; $it = 4.14580$;

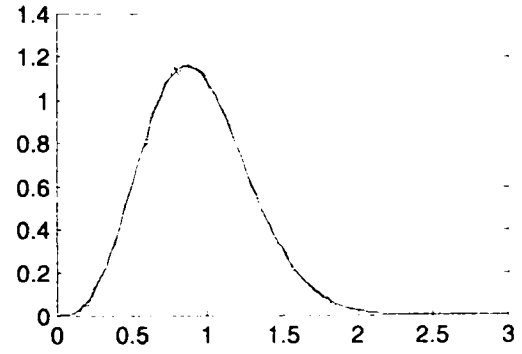
$N=100000$; $m=5$; $c = 1e-2$; $\varepsilon = 1.2736$; $it = 2.71750$; $c = 5e-3$; $\varepsilon = 0.4133$; $it = 2.93946$; $c = 2e-3$; $\varepsilon = 0.1097$; $it = 3.19151$; $c = 1e-3$; $\varepsilon = 0.0426$; $it = 3.35882$; $c = 1e-4$; $\varepsilon = 0.0268$; $it = 3.80281$; $c = 1e-5$; $\varepsilon = 0.0263$; $it = 4.13630$;

$N=1000000; m=2;$

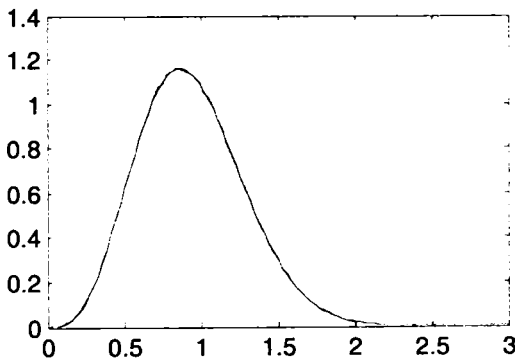
$c = 1e-2 \quad \varepsilon = 2.2548\% \quad it = 2.818566;$



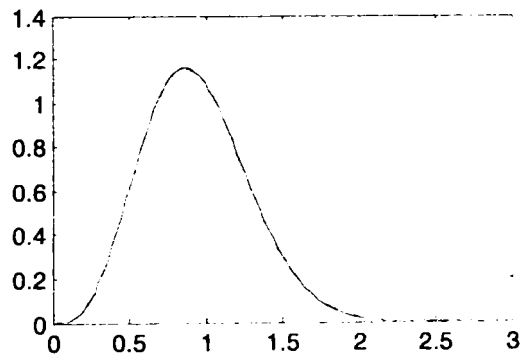
$c = 1e-3 \quad \varepsilon = 0.0408\% \quad it = 3.443234;$



$c = 1e-4 \quad \varepsilon = 0.0046\% \quad it = 3.883406;$

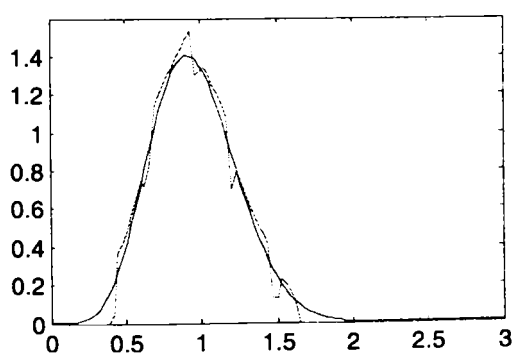


$c = 1e-5 \quad \varepsilon = 0.0040\% \quad it = 4.216796;$

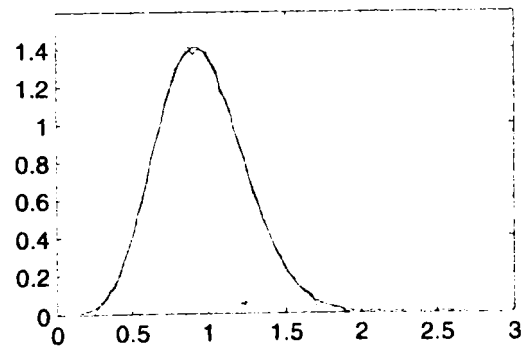


$N=1000000; m=3;$

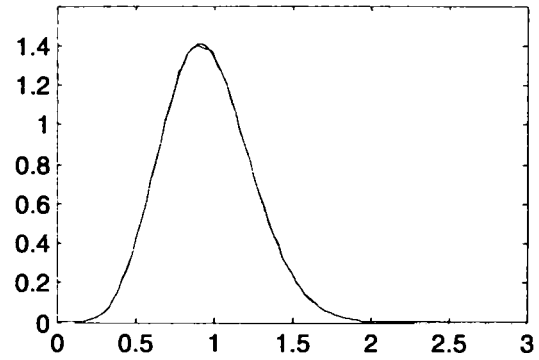
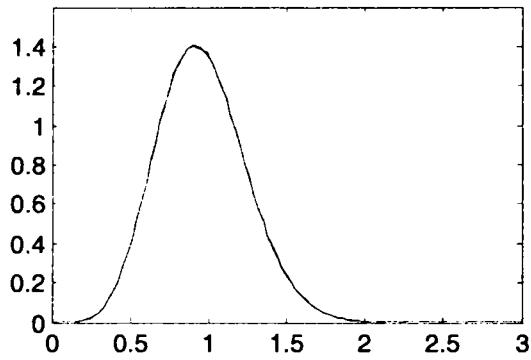
$c = 1e-2 \quad \varepsilon = 0.9888\% \quad it = 2.753021;$



$c = 1e-3 \quad \varepsilon = 0.0316\% \quad it = 3.388850;$

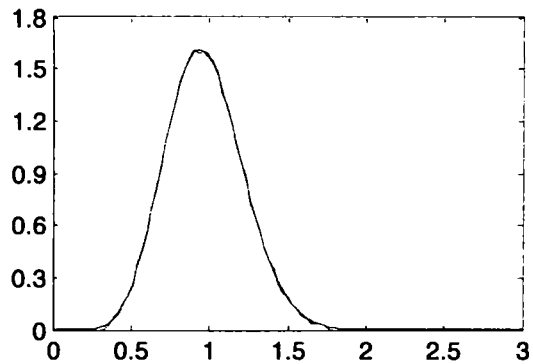
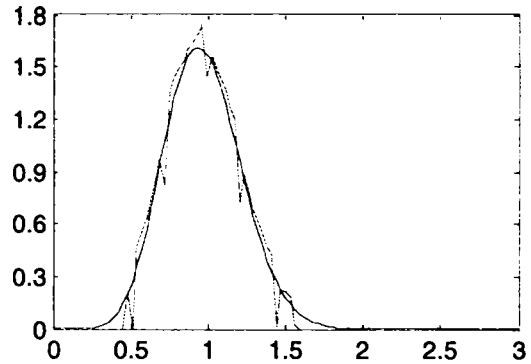


$c = 1e-4$ $\varepsilon = 0.0035\%$ $it = 3.830562$; $c = 1e-5$ $\varepsilon = 0.0037\%$ $it = 4.164949$;

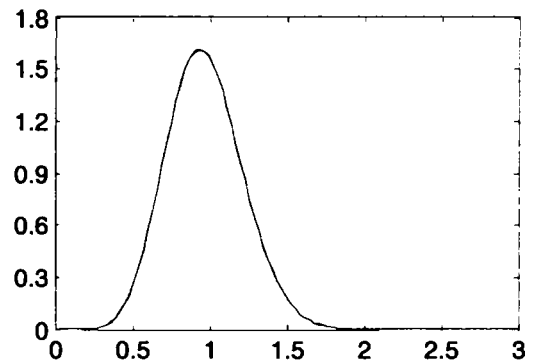
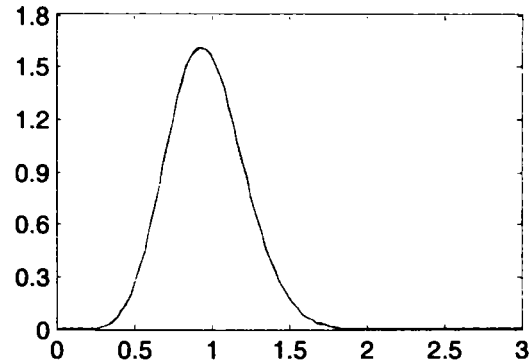


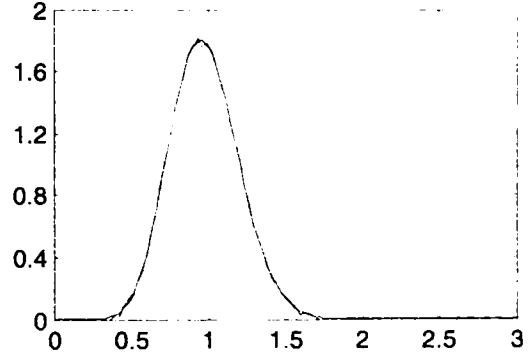
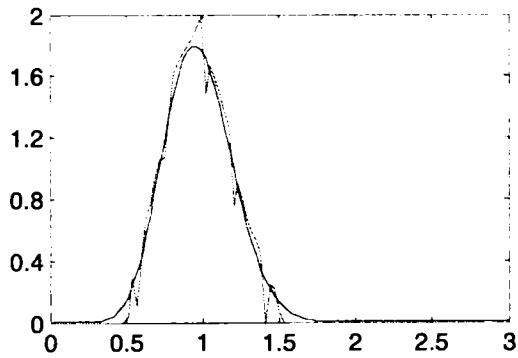
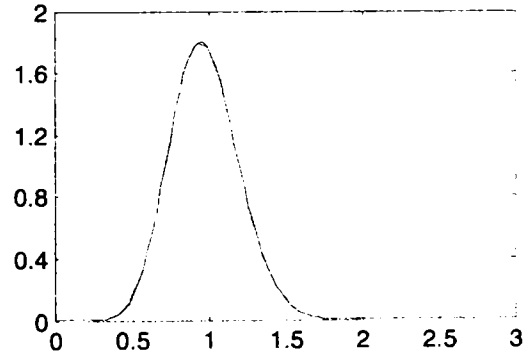
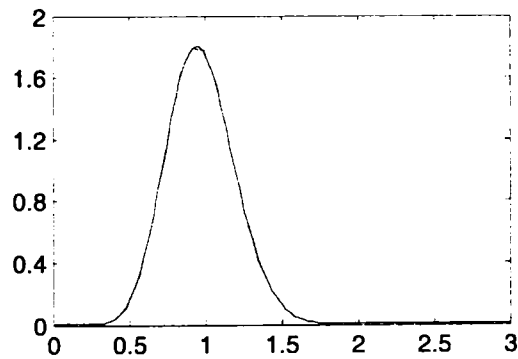
$N=1000000$; $m=4$;

$c = 1e-2$ $\varepsilon = 1.3581\%$ $it = 2.728952$; $c = 1e-3$ $\varepsilon = 0.0152\%$ $it = 3.368606$;



$c = 1e-4$ $\varepsilon = 0.0029\%$ $it = 3.811379$; $c = 1e-5$ $\varepsilon = 0.0025\%$ $it = 4.146293$;



$N=1000000$; $m=5$; $c = 1e-2$ $\varepsilon = 1.2016\%$ $it = 2.717728$; $c = 1e-3$ $\varepsilon = 0.0210\%$ $it = 3.358548$; $c = 1e-4$ $\varepsilon = 0.0022\%$ $it = 3.801103$; $c = 1e-5$ $\varepsilon = 0.0017\%$ $it = 4.136770$;

Bibliografie

- [ACR99] O.F.Acikel, W.E.Ryan, „Punctured Turbo-Codes for BPSK/QPSK Channels”, IEEE Transactions on Communications, sept.1999,
www.ece.arizona.edu/~ryan/New%20Folder/punctured.pdf
- [AHK97] K.S. Andrews, C. Heegard, D. Kozen, “A theory of interleavers”, Technical Report TR97-1634, Department of Computer Science, June 1997
- [ALA98] S. M. Alamouti, “A Simple Transmit Diversity Technique for Wireless Communications”, IEEE Journal on Selected Areas in Communications”, vol. 16, pag. 1451-1458, octombrie 1998
- [APR97] P.Adde, R.Pyndiah, O.Raoul, J.R.Inisan, “Block Turbo Decoder Design”, International Symposium on Turbo Codes – Brest – France, 1997, pag. 166-169
- [BAK00] M.H.Baligh, A.K.Khandani, „Asymptotic Effect of Interleaver Structure on the performance of Turbo-Codes”, 2000,
www.cst.uwaterloo.ca/c/MHBCISS02.pdf
- [BAP99] S. A. Barbulescu and S. S. Pietrobon, “Turbo Codes: A Tutorial on a New Class of Poerful Error Correcting Coding Schemes”, *Journal of Electrical and Electronics Engineering, Australia*, Vol.19, No.3, pp.129-152, Sep. 1999.
- [BAT97] Gérard Battail, “A conceptual framework for understanding turbo-codes”, International Symposium on Turbo Codes – Brest – France, 1997, pag. 55-62
- [BBB01] M. Benchrifa, M. Belkasmi, A. Benouna, „Amélioration des Performances des Codes Convolutifs Concaténés en Parallèle avec un Turbo Décodage Utilisant APRI-SOVA”, ICSIP'2001, Agadir, Morocco, 2001
- [BCJR74] L. R. Bahl, J. Cocke, F. Jelinek, J. Raviv, “Optimal Decoding of Linear Codes for Minimising Symbol Error Rate”, IEEE Transactions on Information Theory, vol. 20, pag. 284-287, martie 1974
- [BDM97] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, „Design of Serially Concatenated Interleaved Codes”, ICC 1997, pag. 710-714
- [BDT03] M.Bickerstaff, L.Davis, C.Thomas, D.Garrett, C Nicol, „A 24 Mb/s Radix-4 LogMAP Turbo Decoder for 3GPP-HSDPA Mobile Wireless”, IEEE International Solid-State Circuits Conference”, februarie 2003,
www.bell-labs.com/issccpaper
- [BEC05] N. Beaulieu, C. Cheng, “Efficient Nakagami- m Fading Channel Simulation”, IEEE journal on Vehicular Technology, vol. 54, no. 2, March 2005.

- [BEJ99] C. Berrou and M. Jézéquel, "Nonbinary convolutional codes for turbo coding," *Electron. Lett.*, vol. 35, no. 1, pp. 39-40, Jan. 1999.
- [BEM97] S. Benedetto, G. Montorsi, "Generalized Concatenated Codes With Interleavers", *International Symposium on Turbo Codes - Brest - France, 1997*, pag. 32-39
- [BEM96a] S. Benedetto, G. Montorsi, "Design of parallel concatenated convolutional codes", *IEEE Transactions on Communications*, vol. 44, pag. 591-600, mai 1996
- [BEM96b] S. Benedetto, G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes", *IEEE Transactions on Information Theory*, vol. 42, pag. 409-428, martie 1996
- [BER97] Claude Berrou, "Some clinical aspects of Turbo Codes", *International Symposium on Turbo Codes - Brest - France, 1997*, pag. 26-31
- [BER65] E. Berlekamp, "On decoding binary Bose-Chaudhuri-Hocquenghem", *IEEE Transactions on Information Theory*, vol. 11, pag. 577-579, 1965
- [BER68] E. Berlekamp, "Algebraic Coding Theory", New York, USA: McGraw-Hill, 1968
- [BGT93] 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
- [BGM98] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Transactions on Communications*, vol. 46, no. 9, pp. 1101-1105, Sep. 1998.
- [BH1] **H. Balta**, "Coduri utilizate în sistemele de transmisiuni cu spectru împrăștiat", Referat Nr.1, Dep. Comunicații, Fac. ETC, UPTimișoara, martie 2003
- [BH2] **H. Balta**, "Coduri utilizate în sistemele de transmisiuni cu spectru împrăștiat", Referat Nr.2, Dep. Comunicații, Fac. ETC, UPTimișoara, martie 2005
- [BH3] **H. Balta**, "Turbocoduri", Referat Nr.3, Dep. Comunicații, Fac. ETC, UPTimișoara, martie 2004
- [BH4] **H. Balta**, Maria Kovaci, "A Comparison between Weight Spectrum of Different Convolutional Code Types", *Analele Universității din Oradea, Fascicula Electrotehnică, secțiunea Electronică, Oradea, România, 2004*
- [BH5] **H. Balta**, Maria Kovaci, "Turbo-coduri: construcție, performanțe, perspective", *Conferința științifică internațională: „Eficiență și calitate în învățământul superior”*, ediția a X-a ediție, "Nicolae Bălcescu" Land Forces Academy - Sibiu, România, iunie 11, 2004

- [BH6] **H. Balta**, Maria Kovaci, "Studiu asupra preciziei curbelor BER construite prin simularea Funcționării turbo-codurilor", Conferința științifică internațională: „Eficiență și calitate în învățământul superior”, ediția a X-a ediție, "Nicolae Bălcescu" Land Forces Academy - Sibiu, România, iunie 11, 2004
- [BH7] **H. Balta**, Maria Kovaci, "A study on turbo decoding iterative algorithms", Buletinul Științific al Universității "Politehnica" din Timișoara, Transactions on Electronics and Communications, Tom 49(63), Fascicola 2, 2004, Oct. 22-23, Timișoara, România
- [BH8] **H. Balta**, Maria Kovaci, "The Performances of Convolutional Codes used in Turbo Codes", Buletinul Științific al Universității "Politehnica" din Timișoara, Transactions on Electronics and Communications, Tom 49(63), Fascicola 2, 2004, Oct. 22-23, Timișoara, România
- [BH9] L.Trifina, **H. Balta**, Anca Rușinariu, "Decreasing of the Turbo MAP Decoding Time Using an Iterations Stopping Criterion", International Symposium on Signals, Circuits and Systems, July 14-15, 2005, Iași, România
- [BH10] Maria Kovaci, **H. Balta**, Miranda Naforniță, "The Performances of Interleavers used in Turbo Codes", International Symposium on Signals, Circuits and Systems, July 14-15, 2005, Iași, România
- [BH11] **H. Balta**, Maria Kovaci, Miranda Naforniță, "A Study on Turbo Coding Systems with $\pi/4$ Shifted DQPSK Modulation", International Symposium on Signals, Circuits and Systems, July 14-15, 2005, Iași, România
- [BH12] **H. Balta**, A. Miloș, M. Rășinar "Simularea canalelor de transmisie (radio) cu fading plat de tip Rayleigh și Rice", Buletinul Universității Petrol-Gaze din Ploiești, Volum LVII, Seria Tehnică, Nr. 2/2005, pag.51-56, 2005, 11-13 mai, Ploiești, România
- [BH13] **H. Balta**, I. Polverea, I. Gheorghită " Performanțele codurilor convoluționale în configurație turbo", Buletinul Universității Petrol-Gaze din Ploiești, Volum LVII, Seria Tehnică, Nr. 2/2005, pag. 57-62, 2005, 11-13 mai, Ploiești, România
- [BH14] **H. Balta**, Maria Kovaci, " A Study on Non-Binary Turbo Codes", The 36-th International Scientific Symposium of Military Equipment & Technologies Research Agency, pp. 214-219, 2005, mai 26-27, București, România
- [BH15] Maria Kovaci, **H. Balta**, "Non-Binary Turbo Codes Interleavers", The 36-th International Scientific Symposium of Military Equipment & Technologies Research Agency, pp. 208-213, 2005, mai 26-27, București, România
- [BH16] **H. Balta**, L.Trifina, Anca Rușinariu, " The Effect of Puncturing on the Convolutional Turbo-codes Performances", The 36-th International Scientific Symposium of Military Equipment & Technologies Research Agency, pp. 288-293, 2005, mai 26-27, București, România

- [BH17] **H. Balta**, M. Răşinar, A. Miloş, "Influenţa estimării canalului asupra performanţelor turbo-codurilor", A XIX-a sesiune de comunicări ştiinţifice cu participare internaţională „Nav-Mar-Edu 2005”, 02-04 2005, Constanţa, România
- [BH18] **H. Balta**, I. Gheorghişă, I. Polvereu, " Utilizarea turbo-codurilor în canalele de transmisie cu fading plat", A XIX-a sesiune de comunicări ştiinţifice cu participare internaţională „Nav-Mar-Edu 2005”, 02-04 2005, Constanţa, România
- [BH19] **H. Balta**, Maria Kovaci, Miranda Nafornită, „A New Method for the Simulation of the Nakagami Flat Fading (Radio) Transmission Channels”, Analele Universităţii din Oradea, Fascicola Electrotehnică, secţiunea Electronică, 2005, Oradea, România
- [BH20] **H. Balta**, Maria Kovaci, "The Turbo-Codes Performances In The (Radio) Rice Flat Fading Channels", The 31st Internationally Attended Scientific Conference "Modern Technologies In The Xxi Century", Military Technical Academy, 2005, nov.03-04, Bucureşti
- [BH21] **H. Balta**, Maria Kovaci, Al. de Baynast, " Performance of Turbo-Codes on Nakagami Flat Fading (Radio) Transmission Channels", Proceedings of IEEE ASILOMAR Conference, pag.606-610, 25-27 octombrie 2005, USA
- [BH22] **H. Balta**, Maria Kovaci, L. Trifina "A Reduced Memory MAP Algorithm for Turbo Codes", Proceedings of the 8th International Conference on Development and Application Systems, Faculty of Electrical Engineering and Computer Science "Stefan cel Mare" University of Suceava, 25-27 May, 2006, pag. 164-168, Suceava, România
- [BH23] L. Trifina, **H. Balta**, „Proposal Of A Cost Function For The Iterative Growth Of Interleaver's Length", Proceedings of the 8th International Conference on Development and Application Systems, Faculty of Electrical Engineering and Computer Science "Stefan cel Mare" University of Suceava, 25-27 May, 2006, Suceava, România
- [BH24] Maria Kovaci, **H. Balta**, Miranda Nafornita "Performance of Multi Binary Turbo Codes on Rayleigh Flat Fading Transmission Channels", Analele Universităţii din Oradea, Fascicola Electrotehnică, Secţiunea Electronică, 2006, pp. 64-67.
- [BH25] **H. Balta**, Maria Kovaci, Miranda Nafornita "Performance of Multi Binary Turbo Codes on Rice Flat Fading Channels", Analele Universităţii din Oradea, Fascicola Electrotehnică, Secţiunea Electronică, 2006, pp. 17-20.
- [BH26] L. Trifina, V. Munteanu, **H. Balta**, „New types of interleavers based on the Welch-Costas permutation", Proceedings of the Second European Conference on the Use of Modern Information and Communication Technologies ECUMICT 2006, pag.107-117, mai 2006, Belgia

- [BH27] L. Trifina, V. Munteanu, **H. Balta**, "Fast Generation of Dividable S-Random Interleavers and their Performance Analysis", The 1st International Symposium on Electrical and Electronics Engineering ISEEE 2006, pag. 62-65, October 13-14, 2006 Galați, România
- [BH28] **H. Balta**, S. El Assad, „Interleaving Distances Spectrum Comparison between Turbo-codes Interleavers”, European Microwave Week, 10-15 septembrie 2006, Manchester, UK
- [BH29] Maria Kovaci, Al. De Baynast, **H. Balta**, Miranda Nafornita "Performance of Multi Binary Turbo-Codes on Nakagami Flat Fading Channels", Buletinul Științific al Universității „Politehnica” Timișoara, Seria Electronică și Telecomunicații, Tom 51-65 Electronică și Telecomunicații, Fascicola 2, 2006, pag.140-145, Timișoara, România
- [BH30] **H. Balta**, Catherine Douillard, Maria Kovaci "The Minimum Likelihood APP Based Early Stopping Criterion for Multi-Binary Turbo Codes", Buletinul Științific al Universității „Politehnica” Timișoara, Seria Electronică și Telecomunicații, Tom 51-65 Electronică și Telecomunicații, Fascicola 2, 2006, pag.199-203, Timișoara, România
- [BH31] **H. Balta**, M. Kovaci, Al. de Baynast, C. Vlădeanu, R. Lucaciu, "Multi-Non-Binary Turbo-Codes From Convolutional to Reed-Solomon Codes", Buletinul Științific al Universității „Politehnica” Timișoara, Seria Electronică și Telecomunicații, Tom 51-65 Electronică și Telecomunicații, Fascicola 2, 2006, pag.113-118, Timișoara, România
- [BH32] Maria Kovaci, **H. Balta**, A. De Baynast, Miranda Nafornita, "Performance Comparison of Punctured Turbo Codes and Multi Binary Turbo Codes", Proceedings of International Symposium on Signal, Circuits and Systems, ISSCS 2007, Iasi, Romania, July, 12-13, 2007, pp. 485-488
- [BHW95] W.J. Blackert, E.K. Hall, S.G. Wilson, "Turbo code termination and interleaver conditions", Electronics Letters 23rd November 1995 Vol. 31 No. 24
- [BJDK01] C. Berrou, M. Jézéquel, Catherine Douillard and Sylvie Kerouédan, „The Advantages of Non-Binary Turbo Codes”, ITW2001, Cairns, Australia, Sept. 2-7, 2001
- [BOC60] R. Bose, D. Ray-Chaudhuri, "On a class of error correcting binary group codes", Information and Control, vol. 3, pag. 68-79, martie 1960
- [BRW] Spencer Brown, „The Turbocharging guide”,
www.turbomustangs.com/turbotech/main.htm
- [BSD04] C. Berrou, Y. Saouter, Catherine Douillard, Sylvie Kerouédan, and M. Jézéquel, "Designing good permutations for turbo codes: towards a single model", International Conference on Communications, IEEE 2004, Paris, France, 20-24 June, vol. 1, pag. 341-345

- [CAB95] G.Caire, E.Biglieri, "Linear Block Codes Over Cyclic Groups", IEEE Transactions on Information Theory, vol. 41, pp. 1246-1256, September 1995
- [CCS02] Consultative Committee for Space Data Systems, "Recommendation for Space Data System Standards", Telemetry Channel Coding, Blue Book, October 2002, <http://www.ccsds.org/CCSDS/documents/101x0b6.pdf>
- [CHA72] D. Chase, "A class of algorithms for decoding block codes with channel measurement information", IEEE Transactions on Information Theory, vol. IT-18, pag. 170-182, January 1972
- [CRO00] S.N. Crozier, "New High-Spread High-Distance Interleavers for Turbo-Codes", Proc. 20th Biennial Symp. Commun., pp. 3-7, May. 2000., (Queen's University, Kingston, Canada).
- [DAM99] F.Daneshgaran, M.Mondin, "Design of Interleavers for Turbo Codes: Iterative Interleavers Growth Algorithms of Polynomial Complexity", IEEE Transactions on Information Theory, vol.45 pp.1845-11859, September 1999
- [DDP98] C. S. Dolinar, D. Divsalar, and F. Pollara, Code Performance as a Function of Block Size, Technical Report TMO Progress Report 42-133, Jet Propulsion Laboratory, California Institute of Technology, May 1998.
- [DIP97] D.Divsalar, F.Pollara, "Serial and Hybrid Concatenated Codes with Applications", International Symposium on Turbo Codes - Brest - France, 1997, pag. 80-87
- [DOB05] Catherine Douillard, C. Berrou, "Turbo Codes With Rate- $m/(m+1)$ Constituent Convolutional Codes, IEEE Transactions on Communications, Vol. 53, No. 10, Oct. 2005, pp.1630-1638
- [DOD95] S. Dolinar, D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations", TDA Progress Report 42-122, August 15, 1995
- [ELI50] P. Elias, "Coding for noisy channels", IRE Convention Record, cap. 4, pag. 37-47, 1950
- [FAN63] R. Fano, "A heuristic discussion of probabilistic coding", IEEE Transactions on Information Theory", vol. IT-9, pag. 64-74, April 1963
- [FEV97] W.Feng, B.Vucetic, "A List Bidirectional Soft Output Decoder of Turbo Codes", International Symposium on Turbo Codes - Brest - France, 1997, pag. 288-292
- [FOJ97] G. David Forney Jr., "On iterative decoding and the two-way algorithm", pp.12-25, International Symposium on Turbo Codes - Brest - France, 1997

- [FOR88] G. Forney, "Coset Codes-Part II: Binary Lattices and Related Codes", IEEE Transactions on Information Theory, vol. 34, pag. 1152-1187, september 1988
- [FOR70] G.D. Forney Jr, "Convolutional Codes I: Algebraic Structure", IEEE Transaction on Information Theory, nov.1970, pp.720-738
- [GAC04] R. Garelo and A. V. Casado, "The all-zero iterative decoding algorithm for turbo code minimum distance computation," in Proc. Int. Conf. Commun., Paris, France, Jun. 2004, pp. 361-364.
- [GAD03] H.El Gamal, M.O.Damen, " Universal Space-Time Coding", IEEE Transactions on Information Theory, vol.49 pp.1097-1119, May 2003
- [GBN01] D.Garrett, Bing Xu, C.Nicol, „Energy Efficient Turbo Decoding for 3G Mobile”, www.acm.org/.../Archives/ProceedingArchives/Compendiums/
- [GHA04] A. Ghrayeb, T. Abualrub, „Asymptotic performance comparison of concatenated (turbo) codes over GF(4)", Int. J. Commun. Syst. 2004, 17, 479-490.
- [GHF95] M.Ghinea, V.Fireșteanu, „MATLAB Calcul numeric – Grafică – Aplicații", Ed. Teora, 1995
- [GOK02] M. Goresky and A. Klapper, Fibonacci and Galois representations of feedback-with-carry shift registers. IEEE Transactions on Information Theory Vol. 48 No. 11:Nov 2002, pp.2826-2836
- [GOP97] A.Goalic, R.Pyndiah, "Real-Time Turbo-Decoding of Product Codes on a Digital Signal Processor", International Symposium on Turbo Codes – Brest – France, 1997, pag. 267-270
- [GOZ61] D. Gorenstein, N. Zierler, "A class of cyclic linear error-correcting codes in pm symbols", Journal of the Society of Industrial and Applied Mathematics, vol. 9, pag. 107-214, iunie 1961
- [HAG97] J.Hagenauer, „The Turbo Principle: Tutorial Introduction and State of the Art", International Symposium on Turbo Codes – Brest – France, 1997
- [HAH99] P. Ha and B. Honary, „A Fast Algorithm for Generating Random Interleavers", In 5th International Symposium on Communications Theory and Applications. Ambleside, UK. 1999. pp. 21-23.
- [HAM50] R. Hamming, "Error detecting and error correcting codes", Bell System Technical Journal, vol. 29, pag. 147-160, 1950
- [HAP94] J. Hagenauer, and L. Papke, „Decoding „Turbo"-Codes with the Soft Output Viterbi Algorithm (SQVA), Proceedings of IEEE International Szmsposium on Information Theorz, pag. 164, June –July 1994

- [HAS71] T. Hasegawa, "On a Coder of Fibonacci Code for Undermiater Digital Data Transmission", IEEE `71 Eng. In the Ocean Environment Conf., pp.381-383
- [HEJ71] J.A. Heller and I.M. Jacobs, "Viterbi Decoding for Satellite and Space Communications", IEEE Transactions on Communication Technology, Vol. COM-19, No. 5, pp. 835-848, 1971.
- [HEW99] C. Heegard, S. B. Wicker, *Turbo Coding*, Kluwer Academic Publishers, 1999
- [HLY02] 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
- [HOE97] P. Hoeher, "New Iterative ("Turbo") Decoding Algorithms", International Symposium on Turbo Codes – Brest – France, 1997, pp. 63-70
- [HOK97] J.Hokfelt, T.Maseng, "Methodical Interleaver Design for Turbo Codes", International Symposium on Turbo Codes – Brest – France, 1997, pp. 212-215
- [HOM97] B. Honary, G. Markarian, "Trellis Decoding of Block Codes", Dordrecht, Olanda: Kluwer Academic, 1997
- [HOP96] 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
- [HOQ59] A. Hocquenghem, "Codes correcteurs d'erreurs", Chiffres (Paris), vol. 2, pag. 147-156, septembrie 1959
- [JBS00] M. C. Jeruchim, P. Balaban, K.S. Shanmugan, „Simulation of Communication Systems, Modeling, Methodology and Techniques”, second edition, Kluwer Academic, 2000
- [JEZ03] M.Jézéquel, „Turbo codes (convolutifs)”, seminar Timișoara, 18-20 martie 2003
<http://hermes.etc.utt.ro/docs/cercetare//carti/tccp2.pdf>
- [JOZ99] R. Johannesson, K. Sh. Zigangirov, „Fundamentals of Convolutional Coding”, IEEE Press, 1999
- [JOW93] R. Johannesson and Z. Wan. A linear algebra approach to minimal convolutional encoders. IEEE Transactions on Information Theory, Vol. 39 No 4 April 1993, pp.1219-1233.
- [JUN94] P. Jung, M. Nasshan, "Performance evaluation of turbo codes for short frame transmission system", IEEE Electronics Letters, vol. 30, pag. 111-112, ianuarie 1994

- [KAL95] G.Kalai, N.Linial, "On the Distance Distribution of Codes", IEEE Transactions on Information Theory, vol. 41, pp. 1467-1471, September 1995
- [KLS92a] H. Krishna, K. Y. Lin, J. D. Sun, "A coding theory approach to error control in redundant residue number systems – part I: Theory and single error correction", IEEE Transactions on Circuits and Systems, vol. 39, pag. 8-17, ianuarie 1992
- [KLS92b] J. D. Sun, H. Krishna, "A coding theory approach to error control in redundant residue number systems – part II: Multiple error detection and correction", IEEE Transactions on Circuits and Systems, vol. 39, pag. 18-34, ianuarie 1992
- [KOB90] W. Koch, A. Baier, Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference", IEEE Globecom, pag. 1679-1684, decembrie 1990
- [KOF97] K.Koora, A. Finger, "A New Scheme to Terminate all Trellis of Turbo-Decoder for Variable Block Length", International Symposium on Turbo Codes – Brest – France, 1997, pp. 174-179
- [KOV03] R.Koetter, A.Vardy, "Algebraic Soft-Decision Decoding of Reed-Solomon Codes", IEEE Transactions on Information Theory, vol.49 pp.2809-2825, November 2003
- [KRL95] I.Krasikov, S.Litsyn, "On Spectra of BCH Codes", IEEE Transactions on Information Theory, vol. 41, pp. 786-787, May 1995
- [KSS95] F.R.Kschischang, V.Sorokine, "On the Trellis Structure of Block Codes", IEEE Transactions on Information Theory, vol. 41, pp. 1924-1937, November 1995
- [LAZ99] A.Lapidoth, J.Ziv, "On the Decoding of Convolutional Codes on an Unknown Channel", IEEE Transactions on Information Theory, vol.45 pp.2321-2331, November 1999
- [MAS63] J. Massey, "Threshold Decoding", Cambridge, USA: MIT Press, 1963
- [MAS65] J. Massey, "Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes", IEEE Transactions on Information Theory, vol. 11, pag. 580-585, 1965
- [MAS69] J. Massey, "Shift-register synthesis and BCH decoding", IEEE Transactions on Information Theory, vol. IT-15, pag. 122-127, ianuarie 1969
- [NGT01] T.M.N. Ngatched și F. Takawira, „Simple stopping criterion for turbo decoding”, Electronics Letters 25th October 2001 Vol. 37 No. 22
- [NHB97] H. Nickl, J. Hagenauer, F. Burkett, "Approaching Shannon's capacity limit by 0.27 dB using simple Hamming codes", IEEE Communications Letters, vol. 1, pag. 130-132, September 1997

- [OBS97] M. Öberg, P. H. Siegel, "The Effect of Puncturing in Turbo Encoders", International Symposium on Turbo Codes – Brest – France, 1997, pp. 184-187
- [PET60] W. Peterson, "Encoding and error correction procedures for the Bose-Chaudhuri codes", IEEE Transactions on Information Theory, vol. IT-6, pag. 459-470, septembrie 1960
- [PKY99] Li Ping, Kwan L. Yeung, „Symbol-by-Symbol APP Decoding of Golay Code and Iterative Decoding of Concatenated Golay Codes”, IEEE Transactions on Information Theory”, Vol 45 No 7 November 1999
- [PLV03] L.Ping, WQ.K.Leung, K.Y.Wu, "Low-Rate Turbo-Hadamard Codes", IEEE Transactions on Information Theory, vol.49 pp.3213-3224, December 2003
- [PRO01] John G. Proakis, "Digital communications", McGraw-Hill Series in Electrical and Computer Engineering Stephen W., 2001.
- [PSC96] L. Perez, J. Seghers, D. Costello, "A distance spectrum interpretation of turbo codes", IEEE Transactions on Information Theory, vol. 42, pag. 1698-1709, noiembrie 1996
- [PSE99] M.Peleg, I.Sason, S.Shamai (Shitz), A.Elia, "On Interleaved, Differentially Encoded Convolutional Codes", IEEE Transactions on Information Theory, vol.45 pp.2572-2581, November 1999
- [PYN97] R. Pyndiah, "Iterative Decoding of Product Codes: Block Turbo Codes", International Symposium on Turbo Codes – Brest – France, 1997, pp. 71-79
- [RES60] I. Reed, G. Solomon, "Polynomial codes over certain finite fields", Journal of the Society of Industrial and Applied Mathematics, vol. 8, pag. 300-304, iunie 1960
- [RGT05] A. C. Reid, T. A. Gulliver, D. P. Taylor, "Rate-1/2 Component Codes for Nonbinary Turbo Codes", IEEE Transactions on Communications, vol. 53, No. 9, september 2005.
- [RHV97] P. Robertson, P. Hoeher and E. Villebrun, "Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding", European Trans. Telecommun., vol. 8, pp. 119-125, Mar-Apr. 1997.
- [RIC00] T.Richardson, "The Geometry of Turbo-Decoding Dynamics", IEEE Transactions on Information Theory, vol.46 pp.9-23, January 2000
- [ROB94] P. Robertson, „Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes”, IEEE GLOBECOM'94, San Francisco, December 1994, pag. 1298-1303

- [ROB97] P. Robertson, "An Overview of Bandwidth Efficient Turbo Schemes", International Symposium on Turbo Codes – Brest – France, 1997, pp. 103-110
- [ROY99] J. Rosenthal, E.V. York, "BCH Convolutional Codes", IEEE Transactions on Information Theory, vol.45 pp.1833-1844, September 1999
- [RVH95] 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
- [SCB95] G. Schnabl, M. Bossert, "Soft-Decision Decoding of Reed-Muller Codes as Generalized Multiple Concatenated Codes", IEEE Transactions on Information Theory, vol. 41, pp. 304-307, January 1995
- [SCG03] C. Schlegel, A. Grant, "Differential Space-Time Turbo Codes", IEEE Transactions on Information Theory, vol.49 pp.2298-2306, September 2003
- [SHA48] C.E. Shannon, "A Mathematical Theory of Communications", The Bell System Technical Journal, Vol.27, pp. 379-423, 623-656. October, 1948
- [SRI05] K. Sripimanwat, "Turbo Code Applications. A Journey from a Paper to Realization", Springer, P.O. Box 17, 3300 AA Dordrecht, The Netherlands, 2005
- [SZT67] N. Szabo, R. Tanaka, "Residue Arithmetic and Its Applications to Computer Technology", New York, USA: McGraw-Hill, 1967
- [SSS00] H.R. Sadjadpour, N.J.A. Sloane, M. Salehi, G. Nebe, "Interleaver Design for Short Block Length Turbo Codes," ICC2000, New Orleans, LA June 18-22 2000
- [SUT05] J. Sun, O. Y. Takeshita, "Interleavers for Turbo Codes Using Permutation Polynomials Over Integer Rings", IEEE Transactions on Information Theory, vol. 51, No. 1, January 2005.
- [TAC00] O.Y. Takeshita, D.J. Costello, "New Deterministic Interleaver Design for Turbo Codes", IEEE Transactions on Information Theory, vol.46, September 2000
- [TAJ99] Tarokh, H. Jafarkhani, "Space-Time Block Codes from Orthogonal Designs", IEEE Transactions on Information Theory, vol.45 pp.1456-1467, July 1999
- [TAS94] D.J. Taipale, M.J. Seo, "An Efficient Soft-Decision Reed-Solomon Decoding Algorithm", IEEE Transactions on Information Theory, vol. 40, pp. 1130-1139, July 1994
- [TJK99] V. Tarokh, H. Jafarhani, A. Calderbank, "Space-time block codes from orthogonal designs", IEEE transactions on Information Theory, vol. 45, pag. 1456-1467, mai 1999

- [TRK95] A.N.Trofimov, B.D.Kudryashov, "Distance Spectra and Upper Bounds on Error Probability for Trellis Codes", IEEE Transactions on Information Theory, vol. 41, pp. 561-571, March 1995
- [UNG87a] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets part I: Introduction", IEEE Communications Magazine, vol. 25, pag 5-11, februarie 1987
- [UNG87b] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets part II: State of the art", IEEE Communications Magazine, vol. 25, pag 12-21, februarie 1987
- [VHS] N.R.Veselinovic, P.Henttu, S.A.Siltala, D.Tujkovic, M.Juntti, "Wideband interference suppression in turbo coded DS/FH spread spectrum system" In: Proc. IEEE Int. Conf. Commun., 2001
- [VIT67] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE Transactions on Information Theory, vol. IT-13, pag. 260-269, aprilie 1967
- [VVN97] A.J. Viterbi, A.M.Viterbi, J.Nicolas, N.T.Sindhushayana, „Perspectives on Interleaved Concatenated Codes with Iterative Soft-Output Decoding”, International Symposium on Turbo Codes – Brest – France, 1997, pag.47-54
- [VUY00] B.Vucetic, J.Yuan, "Turbo Codes Principles and Applications", Kluwer Academic Publishers, USA, 2000
- [WAD00] G.Wade, "Coding Techniques, An Introduction to Compression and Error Control", Creative Print and Design, Ebbw Vale, Great Britain, 2000
- [WAH66] R. Watson, C. Hastings, "Self-checked computation using residue arithmetic", Proceedings of the IEEE, vol. 54, pag. 1920-1931, decembrie 1966
- [WBRC98] C. Weiss, C. Bettstetter, S. Riedel, and D.J. Costello, "Turbo decoding with tailbiting trellises", in Proc. IEEE Int. Symp.Signals, Syst., Electron., Pisa, Italy, Oct. 1998, pp. 343-348.
- [WHO00] J.P.Woodard, L.Hanzo, "Comparative Study of Turbo Decoding Techniques: An Overview", IEEE Transactions on Vehicular Technology, nov.2000
- [WOL78] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis", IEEE Transactions on Information Theory", vol. IT-24, pag. 76-80, ianuarie 1978
- [WOR61] J.Wozencraft, B. Reiffen, "Sequential Decoding", Cambridge, MA, USA: MIT Press, 1961

- [WOZ57] J.Wozencraft, "Sequential decoding for reliable communication", IRE NATL. Conv. Rec., vol. 5, cap. 2, pag. 11-25, 1957
- [ZMY97] S.Zhou, S.Mei, Y.YAO, "Turbo-Code without Chanel Estimator", International Symposiom on Turbo Codes – Brest – France, 1997, pag. 147-150