Universitatea “Politehnica” din Bucureşti

Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei

Modul de simulare de coduri corectoare de erori LDPC în canale magnetice

Lucrare de dizertaţie

prezentată ca cerinţă parţială pentru obţinerea titlului de

Master în domeniul Inginerie, Electronică si Telecomunicaţii

programul de studii de masterat Ingineria Informaţiei si a Sistemelor de Calcul

Conducător ştiinţific — *Conf. Dr. Ing.* Ştefan Stăncescu

Absolvent — Ichimoaei Ştefan-Cosmin

2012

CUPRINS

Capitolul IV. Codarea LDPC

Capitolul V. Decodarea LDPC

Capitolul VI. Aplicaţii ale codurilor LDPC

Capitolul VII. Structura aplicaţiei de simulare

Capitolul I. Introducere

1.1 Scopul lucrării

Dispozitivele de stocare a informaţiei pe discuri magnetice reprezintă o colecţie impresionantă de subsisteme mecanice şi electronice, care operează la limitele tehnologiei moderne. Proiectarea şi realizarea acestor unităţi, care poartă denumirea specifică de **hard-discuri,** necesită o experienţă vastă într-un număr impresionant de domenii, cum ar fi: câmpurile electrice şi magnetice, materialele, aerodinamica, mecanica, electromecanica, comunicaţiile digitale, codarea, procesarea de semnal şi proiectarea circuitelor integrate. Împreună, experţii în aceste discipline au reuşit să menţină rate exponenţiale de creştere a capacităţii de stocare a datelor şi a ratelor de transfer în ultimii ani, evoluţie care se prevede şi pentru viitorul apropiat.

Creşterea rapidă a capacităţilor de stocare şi a vitezelor de transfer presupune şi creşterea densităţii de înregistrare a informaţiei pe disc, ceea ce impune dezvoltarea unor metode din ce în ce mai performante de codare şi detecţie a informaţiei în canalele magnetice, lucru necesar pentru a păstra integritatea datelor înregistrate. În cadrul acestei lucrări va fi prezentată metoda de codare a semnalului prin codarea LDPC, una dintre cele mai performante metode de codare existente la ora actuală.

1.2. Principii de înregistrare a informaţiei pe suport magnetic

1.2.1. Structura fizică a unui hard-disc

În această secţiune vom analiza pe scurt componentele de bază ale unui hard-disc şi vom evidenţia influenţele acestora asupra semnalului înregistrat acolo unde este necesar. Figurile următoare prezintă o imagine simplificată a componentelor mecanice care alcătuiesc sistemul de înregistrare magnetică al unui hard-disc.

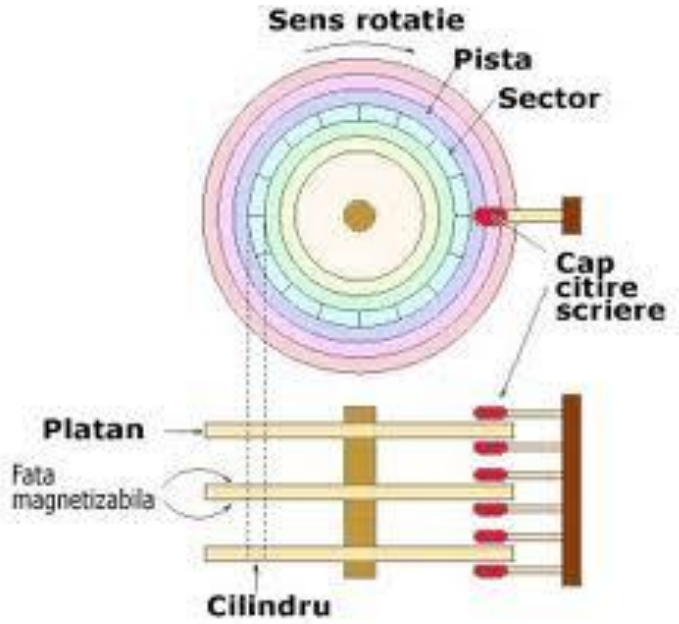


Figura 1.1: Componentele mecanice esenţiale ale unui hard-disc.

Există câte un cap de citire/scriere suspendat deasupra fiecărei suprafeţe (platan) al unui disc rotativ, pe care sunt stocate datele. Pe măsură ce discul se roteşte, capul va trece peste o regiune inelară a discului numită **pistă** unde va fi făcută înregistrarea propriu-zisă. Dacă discul este constituit din mai multe platane, atunci pistele cu aceeaşi rază de pe fiecare platan vor forma un **cilindru**. Discul este învelit într-o peliculă de material magnetic dur, care va reţine magnetizarea creată de capul de citire/scriere. Datele sunt înregistrate sub forma unor regiuni magnetizate sau celule bit, pe fiecare pistă. Fiecare bit înregistrat va avea una din cele două direcţii posibile de magnetizare ale câmpului magnetic. Trecerea de la o direcţie de magnetizare la cealaltă poartă denumirea de **tranziţie magnetică.**

Figura 1.2 prezintă detaliat secţiunea transversală a interfeţei cap magnetic – mediu, în timpul unui proces de scriere pe disc şi respectiv citire de pe disc.

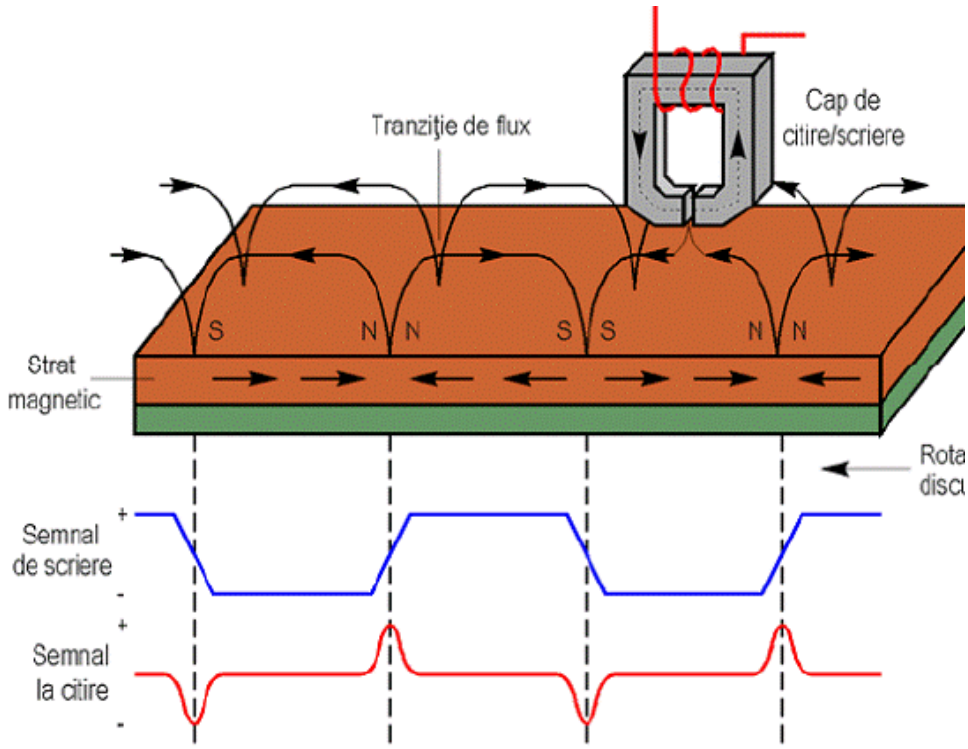


Figura 1.2: Secţiune transversală a interfeţei cap magnetic-mediu.

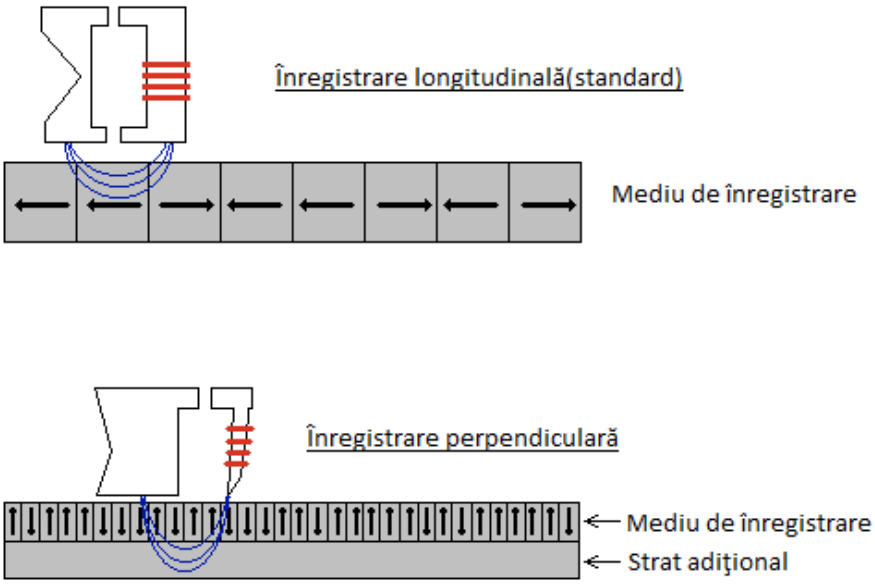


Figura 1.3: Diferenţele dintre tehnologiile de înregistrare longitudinală şi transversală.

O formă de undă tipică a tensiunii induse în capul magnetic de citire, este prezentată în figura 1.4. Trebuie menţionat că în figură s-a considerat semnalul ca fiind neafectat de zgomot şi că de obicei înainte de a fi scrise, datele sunt precodate, astfel că în timpul procesului de citire un puls de tensiune pozitiv sau negativ va corespunde unui bit de „1”, în timp ce absenţa unui puls va corespunde unui bit de „0” (codare **NRZI**).

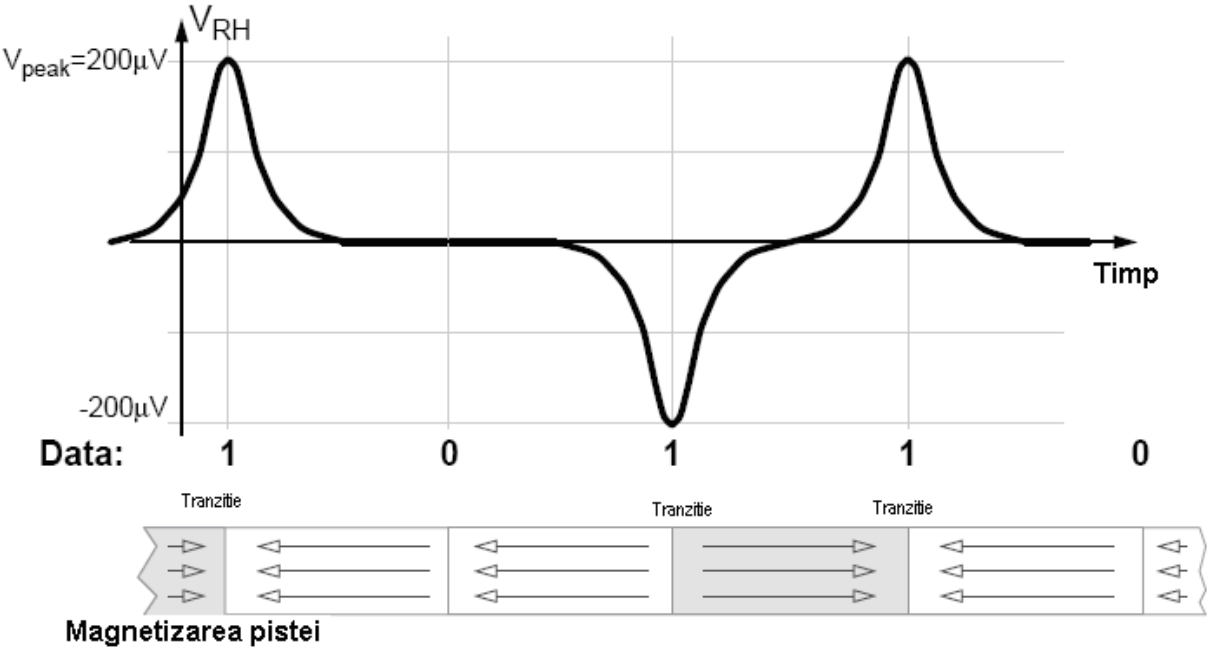


Figura 1.4: Forma tensiunii induse în capul de citire (la densitate de înregistrare scăzută).

Din figura anterioară sunt importante de reţinut două aspecte: unui bit de „1” îi va corespunde întotdeauna o tranziţie magnetică şi două pulsuri de tensiune consecutive vor avea întotdeauna polarităţi diferite. Perioada de timp în care amplitudinea unui puls de tensiune depăşeşte 50% din valoarea sa de vârf este considerată ca reprezentând 50% din lăţimea impulsului şi se notează cu **PW50.** Această noţiune este importantă deoarece densitatea de înregistrare a canalului magnetic se defineşte ca fiind raportul dintre PW50 şi perioada de bit a canalului (**T**): **D=PW50/T.** La densităţi de înregistrare foarte mari spaţiul dintre impulsurile de tensiune din figura 1.4 începe să scadă până când acestea vor începe să se suprapună, fenomen care poartă denumirea de **interferenţă intersimbol (ISI).** Prezenţa ISI împreună cu cea a zgomotului poate conduce la nivele foarte ridicate ale ratei de erori în timpul procesului de citire/recuperare a datelor.

1.2.2. Metode de codare a informaţiei în canale magnetice

Codarea informaţiei are un rol esenţial în structura canalului magnetic de înregistrare, în acest mod prevenindu-se probleme majore care pot afecta sincronizarea capului de citire-scriere cu discul magnetic şi integritatea datelor.

În figura următoare sursa de informaţii este de obicei un sistem de calcul sau un alt sistem capabil să genereze date în format binar. Informaţia este codată cu ajutorul unui cod corector de erori (de exemplu cod Reed-Solomon) cu rol de protecţie a informaţiei stocate în cazul unor defecte de material, zgomote, etc. Semnalul binar rezultat este în continuare aplicat unui circuit de scriere care realizează prelucrarea acestuia în vederea adaptării sale la mediul magnetic de stocare. Aceste prelucrări constau în codări suplimentare (de exemplu codare RLL, etc), amplificare, etc. Semnalul obţinut este apoi aplicat capului de scriere, care realizează înregistrarea fizică a informaţiei pe suportul magnetic, sub forma unor tranziţii în magnetizarea acestuia.

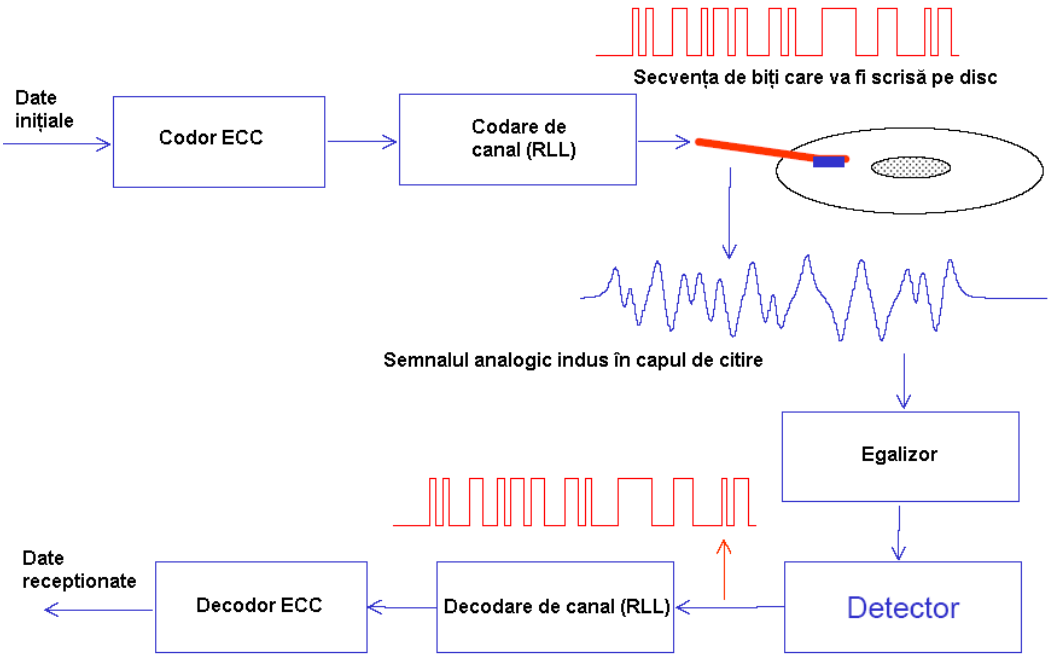


Figura 1.5: Schema tipică a unui canal de înregistrare pe suport magnetic.

La citire, capul magnetic de citire furnizează în exterior un semnal obţinut prin inducerea în circuitul magnetic al acestuia a unei tensiuni prin inducţie electromagnetică. Acest semnal este aplicat unui circuit care realizează o amplificare şi eventual alte prelucrări (de exemplu, filtrarea în vederea eliminării interferenţei între biţii vecini). Semnalul (analogic) obţinut la ieşirea circuitului amintit mai sus este aplicat unui circuit de detecţie, care realizează refacerea informaţiei digitale. Informaţia digitală obţinută este decodată, corectându-se eventualele erori apărute, şi furnizată sistemului de calcul.

În continuare vom face o serie de precizări suplimentare legate de codurile RLL. Am precizat anterior faptul că înainte de a fi scrişi, biţii de informaţie sunt precodaţi, sau codaţi NRZI. Codul NRZI este folosit pentru că dispozitivul de citire (capul magnetic) are o caracteristică de derivator (funcţionează pe baza legii inducţiei electromagnetice), deci în semnalul de ieşire tranziţiile de magnetizare vor corespunde unor vârfuri, care pot fi detectate cu uşurinţă ca biţi de ―1‖. Principalul dezavantaj al acestui cod este faptul că la apariţia unor şiruri lungi de biţi ―0‖ este posibilă pierderea sincronizării circuitului de citire. Mai precis, datorită unor fluctuaţii inerente în viteza de rotaţie a

discului, capul magnetic ajunge să fie poziţionat incorect în raport cu locaţia de pe disc, care trebuie citită/scrisă. Aceste fluctuaţii sunt compensate de un circuit specializat, o buclă PLL, care însă are la rândul ei nevoie de un semnal de tact pe care îl preia chiar din semnalul indus în capul de citire de secvenţa înregistrată. Un şir lung de biţi de „0‖ va duce la absenţa acestui semnal, aşa cum se poate observa în figura 1.6, şi la imposibilitatea compensării fluctuaţiilor de viteză amintite mai sus. Codurile RLL previn această problemă introducând câte un bit de „1‖ după un anumit număr de biţi „0‖.

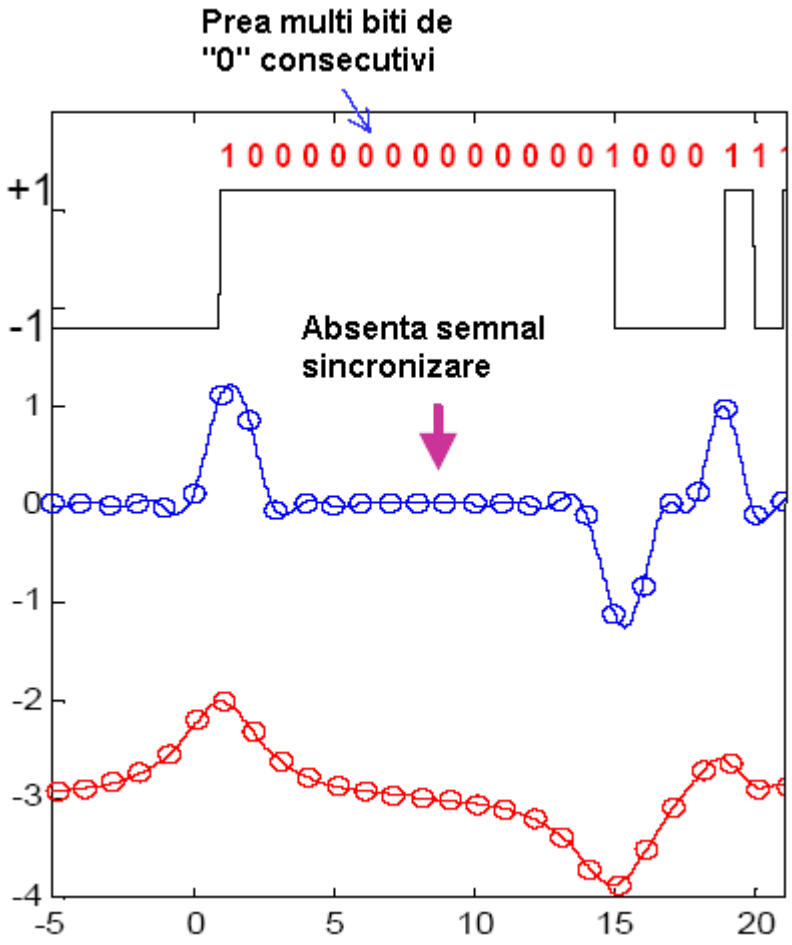


Figura 1.6: Forma de undă asociată unei secvenţe lungi de biţi „0”.

Codurile RLL clasice, (de exemplu **RLL** (1,7) şi **RLL** (2,7)) au ca principal dezavantaj introducerea unei redundanţe semnificative a secvenţei codate în raport cu secvenţa originală (între 50%-100%). Acest fenomen implică necesitatea creşterii densităţii de înregistrare pentru a compensa spaţiul pierdut prin codare şi astfel se pierde câştigul iniţial obţinut prin micşorarea ratei de erori de bit. Codurile **MTR (Maximum Transition Run)** RLL, rezolvă această problemă prin adaptarea tehnicii de codare la tipul canalului. Astfel, deoarece canalele PRML şi DFE permit nativ densităţi de înregistrare mai ridicate, sau altfel spus sunt mai puţin sensibile (DFE) sau chiar utilizează efectele ISI (PR), codul nu mai este obligat să separe perechile de biţi de ‚1’ prin biţi de ’0’, ci poate permite

secvenţe de tranziţii magnetice de lungime maximă fixată [12]. Pentru codurile din această lucrare, lungimea maximă este de 3. În acest mod redundanţele efective ale codurilor implementate variază între: 25% şi 6,25%, reducând spaţiul pierdut pe disc prin codare.

Scopul principal al lucrării este de a prezenta performanţele codării LDPC.

Capitolul II. Coduri corectoare de erori.

Concepte de bază

Toate codurile corectoare de erori se bazează pe acelaşi principiu: Informaţiei îi este adăugată redundanţă, pentru a încerca corectarea eventualelor erori care pot apărea în procesul de stocare sau de transmisie. În principiu, simboluri redundante sunt ataşate simbolurilor de informaţie, pentru a se obţine o secvenţă codată sau un *cuvânt de cod.* În figura 2.1 este prezentat un cuvânt de cod obţinut prin codare cu un bloc cod. O astfel de codare se numeşte *sistematică*. Asta înseamnă că simbolurile de informaţie vor apărea întotdeauna în primele k poziţii ale cuvântului de cod. Cele (n-k) simboluri rămase din cuvântul de cod sunt anumite funcţii de simboluri de informaţie si asigură redundanţă, care poate fi folosită pentru detectare/corectare erorilor. Mulţimea tuturor secvenţelor de cod de acest fel se numeşte cod corector de erori, şi va fi notat cu C.

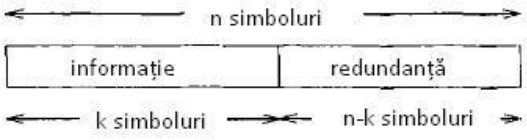


Figura 2.1 Codare bloc sistematică pentru corecţia erorilor

2.1 Coduri bloc şi coduri convoluţionale

În funcţie de metoda prin care redundanţa este adaugată mesajului, codurile corectoare de erori se împart în două clase: coduri bloc şi coduri convoluţionale. Ambele tipuri de codări au aplicaţii practice. De-a lungul timpului, codurile convoluţionale au fost preferate, aparent datorită posibilităţii implementării algoritmului de decodare Viterbi, şi datorită părerii conform căreia codurile bloc nu vor putea fi decodate eficient cu decizii soft. Totuşi, descoperirile recente în teoria şi dezvoltarea algoritmilor de decodare cu decizii soft pentru codurile bloc liniare au înlăturat această neîncredere. Mai mult, cele mai bune coduri corectoare de erori cunoscute astăzi sunt blocurile cod (coduri neregulate cu densitate mică şi control al parităţii).

2.1.1 Coduri convoluţionale

Codurile convoluţionale diferă de codurile bloc atât prin faptul că există memorie pentru codor, cât si prin dependenţă celor *n* ieşiri, la fiecare unitate de timp, nu numai de cele *k* intrări, ci şi de cele *m* blocuri anterioare de intrare.

Un cod convoluţional *(n,k,m)* poate fi implementat cu un circuit secvenţial care are *n* ieşiri liniare, *k* intrări şi *m* intrări de memorie. Uzual, *n* şi *k* sunt valori întregi cu $k < n$, dar valoarea lui *m* trebuie să fie mai mare pentru a avea probabilitate de eroare mică. În cazul în care k=1, secvenţa de informaţie nu mai este divizată în blocuri şi poate fi procesată în mod continuu. Codurile convoluţionale au fost introduse în 1955 de către Elias ca o alternativă a codurilor bloc. La scurt timp după aceea, Wozencraft a propus *decodarea secvenţială* ca fiind o metodă eficientă pentru codurile convoluţionale. În 1963, Massey a propus o metodă de decodare mai puţin eficientă, dar mai simplu de implementat , numită *decodarea cu prag.* Aceasta a dat naştere numeroaselor aplicaţii ale codurilor convoluţionale în cadrul transmisiilor digitale prin cablu sau unde radio. În 1967, Viterbi a propus o metodă de decodare de plauzibilitate maximă, uşor de implementat pentru codurile cu memorie mică. Această schemă, denumită *decodorul Viterbi,* împreună cu versiunea îmbunătăţită a decodării secvenţiale, au lărgit şi mai mult domeniul de aplicaţie al codurilor convoluţionale spre comunicaţiile prin satelit, la începutul anilor 1970.

2.1.2 Coduri bloc

Codurile bloc procesează informaţia bloc-cu-bloc, tratând fiecare bloc de biţi de informaţie separat de celelalte. Cu alte cuvinte, codarea bloc este o operaţie fără memorie, în sensul că toate cuvintele de cod sunt independente unul de celălalt. În schimb, rezultatele codărilor convoluţionale depind nu numai de informaţia de intrare în momentul actual, dar şi de intrări si rezultate precedente, fie într-un mod bloc-cu-blo, fie bit-cu-bit. Trebuie menţionat că şi codurile bloc au memorie, când sunt proiectate ca un proces bit-cu-bit, în cadrul cuvântului de cod. În ultima perioadă, în schimb, diferenţele dintre codurile bloc şi cele convoluţionale sunt din ce în ce mai mici, mai ales în urma recentelor avansuri în înţelegerea structurilor trellis şi a unor structuri circulare convoluţionale. Într-adevăr, unii cercetători care lucrează cu coduri convoluţionale le numesc uneori ―coduri cu structură trellis, variabilă în timp‖, în timp ce unii cercetători care lucrează cu coduri bloc le denumesc pe cele din urmă - coduri cu structură trellis regulată.

2.2 Coduri Turbo

2.2.1 Scurt istoric

Codarea turbo a fost introdusă în 1993 de Berrou, Glavieux şi Thitimajasima[1], care raportau rezultate impresionante pentru coduri cu lungime mare a cadrelor. Din momentul apariţiei, turbocodarea a evoluat într-un ritm fără precedent, datorită eforturilor intense depuse de cercetătorii din domeniu. Ca urmare, turbo codurile au fost introduse şi în standarde, ca de exemplu standardul pentru comunicaţii mobile de generaţia a treia (3G). În sistemele video de difuzare, unde întârzierea asociată sistemului e mai puţin critică decât în sistemele interactive sensibile la întârzieri, câştigurile în performanţă care pot fi atinse sunt şi mai impresionante.

In lucrarea lor, autorii au folosit concatenarea paralelă a două coduri convoluţionale recursive sistematice, **RSC** (Recursive Systematic Convolutional codes), cu un dispozitiv de aleatorizare (**interleaver**) plasat intre cele doua codoare. Pentru decodare, au folosit o structură iterativă având implementată o variantă modificată a algoritmului MAP (Maximum A Posteriori probability), propus de Bahl, Cocke, Jelinek şi Raviv. Pentru a fi eficient, procesul de decodare este iterativ, în sensul că fiecare decodor va genera, la fiecare iteraţie, o informaţie extrinsecă utilizată de celălalt decodor pentru a-si îmbunătăţi performanţele. Dacă numărul de iteraţii creşte atunci creşte şi performanţa decodorului, ajungând la limita capacităţiii Shannon.

De la apariţia turbocodurilor s-a depus un efort enorm în domeniu, pentru a reduce complexitatea decodorului, de către diverşi cercetatori, ca Robertson cu Villebrun şi Hoeher, Berrou cu Adde Angui şi Faudeuil, Bataille ş.a. S-a mai propus şi utilizarea turbocodurilor împreuna cu scheme de modulaţie care folosesc eficient banda de frecvenţe. Lucrările lui Benedetto şi Montorsi respectiv Perez ş.a.[3] au facilitat înţelegerea cauzelor performanţelor excelente ale turbocodurilor. Hagenauer ş.a. a extins conceptul şi pentru codurile bloc concatenate. Jung şi Nasshan[4] au analizat performanţele sistemelor codate în care se folosesc cadre scurte, ca de exemplu sistemele de transmisie a vocii. Ei au aplicat turbocodurile şi la sistemele CDMA (Code Division Multiple Acces), folosind detecţia combinată cu diversitatea antenelor. Barbulescu şi Pietrobon s-au ocupat de proiectarea interleaverelor. In lucrarea sa, Sklar a descris decodorul iterativ şi componentele acestuia.

2.2.2 Configuraţii de turbo coduri

Cel mai întâlnit model al unui codor turbo este reprezentat prin utilizarea în paralel asupra aceleiaşi informaţii a două, trei sau n coduri convoluţionale recursive sistematice (RSC -Recursive Sistematic Codes).

Deşi codurile nerecursive, nesistematice au o distanţă liberă d_{free}, mai mare decât cele recursive şi sistematice, şi astfel utilizate în varianta clasică (fără a fi concatenate) oferă rezultate mai bune din punct de vedere al ratei erorii, totuşi în componenţa turbocodurilor se dovedesc a avea performanţe superioare codurile recursive şi sistematice .

Structura paralelă foloseşte două sau mai multe coduri RSC fiecare cu un întreţesător propriu diferit de celelalte. Datorită întreţeserii, chiar dacă prelucrează aceeaşi informaţie, codoarele componente vor genera biţi de paritate diferiţi.

O structură generală a unui codor turbo convoluţional concatenat în paralel (**PCCC – Parallel Concatenated Convolutional Code**)[2] este prezentată în Figura 2.1

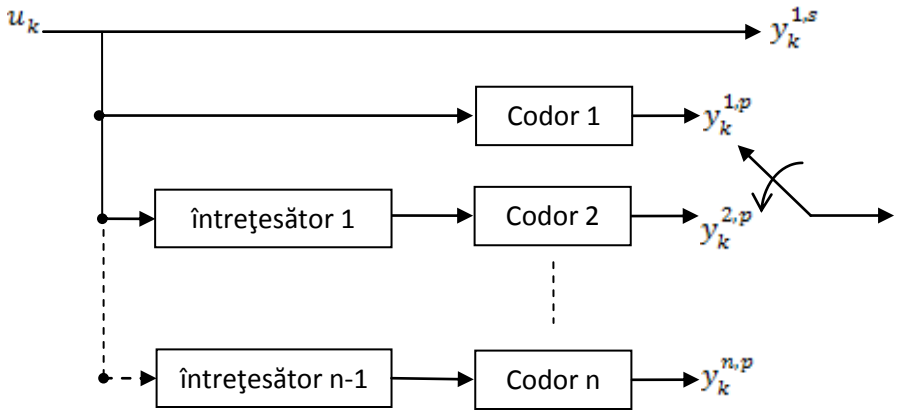


Figura 2.1 - Codor turbo convoluţional concatenat în paralel cu rata de 1/(n+1)

Codurile convoluţionale folosite de codurile turbo au de obicei lungimi de constrângere mici, în jur de 3, 5. Dacă în cazul codurilor convoluţionale simple lungimea de constrîngere mare repezintă un avantaj, în cazul codurilor turbo nu duce la performanţe mai bune, în shimb creşte complexitatea calculelor şi întârzierea decodării.

Dacă concatenăm două coduri obţinem un semnal cu rata 1/3. Dacă folosim trei coduri rata va fi 1/4, şi tot aşa. De obicei două codoare sunt suficiente întrucât creşterea numărului acestora reduce eficienţa benzii fără să ducă la îmbunătăţiri proporţionale ale performanţelor.

Un alt model de codor turbo este realizat prin concatenarea în serie a codoarelor convoluţionale (SCCC –Serial Concatenated Convolutional Code). Codurile SCC au performanţe mai bune în cazul rapoartelor semnal zgomot (SNR) mari. Dacă în cazul codurilor PCC era necesar ca toate codurile componente să fie RSC, în cazul codurilor SCC doar codul de la intrare trebuie să fie RSC. De asemenea ratele codurilor componente ale unui SCCC pot fi diferite. Codul de ieşire poate fi chiar un cod bloc.

Un codor turbo realizat prin concatenarea în serie a altor coduri cu rate diferite poate arăta ca în Figura

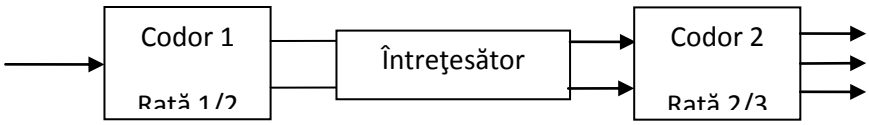


Figura 2.2 - Codor turbo realizat prin concatenare serie

De asemenea există şi modele hibride care folosesc atât concatenarea serie cât şi cea paralelă. Un astfel de exemplu este ilustrat în Figura .3

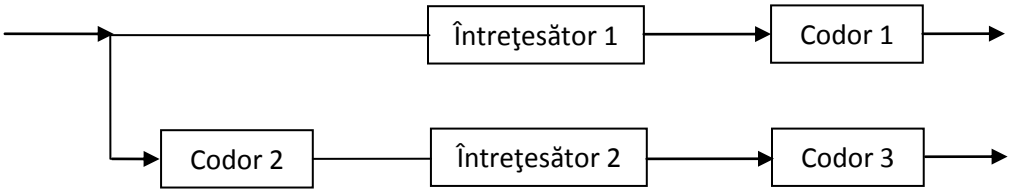


Figura 2.3 - Codor turbo hibrid

Un alt tip de codor numit codor turbo produs (TPC –Turbo Product Code) are o structură foarte diferită de cea paralelă sau serie. Codul TP foloseşte coduri bloc în locul celor convoluţionale. Două coduri bloc diferite (de obicei coduri Hamming) sunt concatenate serial fără a mai exita vreun întreţesător între ele. Funcţia de întreţesere este asigurată de aplicaţia rând-coloană de la un codor către celălalt. Întrucât codurile sunt independente şi operează pe rânduri şi coloane,se obţine o aleatorizare suficientă şi de aceea interleaver-ul nu mai este necesar.

La fel ca şi codurile PCC, codurile TP funcţionează bine la raport semnal zgomot scăzut şi pot fi formate prin concatenarea oricăror tipuri de coduri bloc.

Nu structura codurilor prezentate mai sus duce la denumirea de „Turbo” ci tipul decodării folosite. Este vorba de o decodare iterativă cu reacţie (conexiune inversă) care aminteşte de principiul de funcţionare a motoarelor turbo.

2.2.3 Explicarea principiului Turbo

În Figura 2.4 se prezintă schema unui TC în configuraţie paralelă. Secvenţa de informaţie, notată u, este codată de către codorul C1 rezultând secvenţa de paritate x_1. Aceeaşi secvenţă de biţi, u, este furnizată, însă în altă ordine prin întreţesere cu ajutorul dispozitivului de întreţesere „I”, codorului C2, care generează la rândul său, secvenţa de paritate x_2. Secvenţele rezultate x_0=u, x_1 şi x_2, prin multiplexare şi modulare (operaţii omise în Fig.2.4) constituie ieşirea TC-ului, semnal ce va fi emis în canal. La ieşirea acestuia, prin demodulare şi demultiplexare rezultă secvenţele (soft) recepţionate, y_0, y_1 şi y_2.

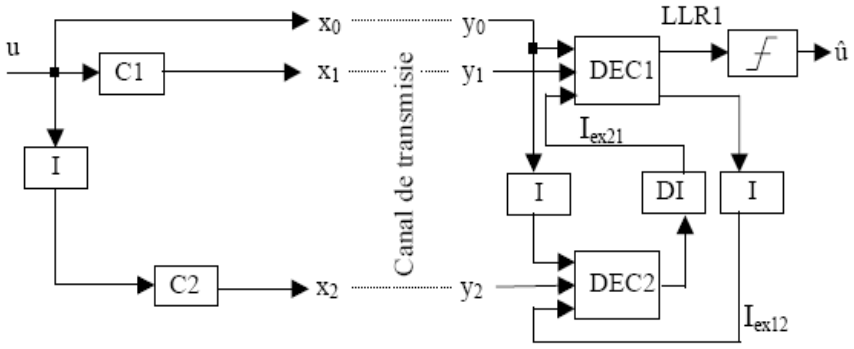


Figura 2.4 - Schema unui turbocod (cod concatenat paralel)

Fiecare decodor calculează logaritmul raportului de plauzibilitate (Log Likelihood Ratio –LLR) pentru fiecare bit din u*:* $LLR(u_i)=ln\frac{p(u_i=1/y)}{p(u_i=0/y)}$ (1)

(în figură este prezentat doar logaritmul raportului de plauzibilitate al primului decodor, notat LLR1) şi de asemenea informaţia extrinsecă destinată celuilalt decodor. Fiecare decodor primeşte informaţie extrinsecă şi, pe baza ei şi pe baza secvenţelor venite din canal (y_0 şi y_1 respectiv y_0 întreţesută şi y_2) furnizează la rându-i informaţie extrinsecă. Acest proces se repetă iterativ de un anume număr de ori (impus sau calculat, funcţie de tipul TC-ului). După efectuarea tuturor iteraţiilor impuse, se face o decizie hard asupra logaritmului raportului de plauzibilitate generat după ultima iteraţie de unul din cele două decodoare. Secvenţa rezultată prin operaţia de decizie hard constituie ieşirea TC-ului. Codoarele utilizate în TC-uri sunt în special cele recursive şi sistematice. În Figura 5 este prezentat un codor convoluţional recursiv, sistematic, având matricea generatoare:

$G(D)=[1, (1+D^2)/(1+D+D^2)]$ sau, în octal: G = [1, 5/7].

În contrast cu codorul convoluţional, care are o implementare hard simplă, după cum se vede în Fig. 2.5, un decodor, pentru a putea fi component al unui TC, trebuie să accepte intrare soft şi, de asemenea, să ofere ieşire soft. *Dispozitivul de întreţesere* (interleaver) realizează o permutare a secvenţei de intrare. Altfel spus, implementează o funcţie bijectivă de forma: $\pi : I \rightarrow I$, cu $I = \{1,2, ... N\}$ (2)

Rolul acestei rearanjări este de a produce o decorelare între diferitele intrări ale unui decodor component.

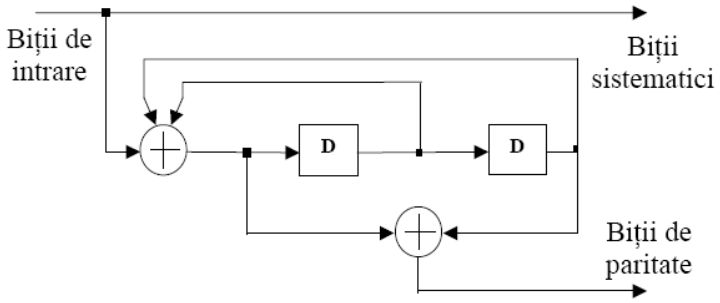


Figura 2.5 – Codor convoluţional recursiv,sistematic, cu $G=[1,(1+D^2)/(1+D+D^2)]$

2.2.4 Procesarea Turbo

Un sistem de comunicaţie tipic constă în general din o cascadă de subsisteme cu diferite funcţionalităţi de procesare a semnalului. Fie, de exemplu, un sistem simplu de comunicaţii folosind codare de canal şi semnalizare peste un canal afectat de interferenţă inter-simbol (IIS), precum in figura2.6. Într-un receptor convenţional, demodulatorul ia decizii hard referitor la biţii transmişi {b[i]} pe baza semnalului recepţionat r(t), care sunt apoi transmişi la decodorul canalului care reface informaţia originală. Problema cu această abordare este că deciziile hard asupra biţilor se face cu pierdere de informaţie în fiecare subsistem (demodulator şi decodor). Aceasta deoarece în timp ce subsistemul indică numai dacă bitul recepţionat este 0 sau 1, el are totuşi suficientă informaţie să estimeze gradul de încredere asupra deciziilor sale. O soluţie ce elimină pierderea de informaţie şi implicit scăderea performanţei este transmiterea nivelului de încredere cu deciziile făcute (adică să facă decizii soft). Aceasta se face când se transmite informaţia de la demodulator la decodor, ceea ce duce la un câştig de 2dB pentru canalul cu zgomot alb gaussian aditiv (AWGN). Totuşi, chiar dacă se fac decizii soft optimale pentru fiecare bit pentru transmiterea de informaţii între oricare subsisteme, performanţa este încă departe de cea optimă. Cauza este că deşi blocurile mai depărtate în lanţul de procesare (decodorul) folosesc informaţia de la stagiile anterioare (demodulatorul), inversul nu se întâmplă. Performanţa optimă este atinsă în cazul efectuării detecţiei în comun, ţinându-se cont simultan de toate procesările din receptor (detecţie a probabilităţii maxime bazată pe supertrellis-ul atât a codului de canal cât şi a canalului IIS), complexitatea acestei abordări fiind foarte prohibitivă. De aceea s-a propus o abordare iterativă care permite stagiilor incipiente (demodulare) să-şi rafineze rezultatele pe baza informaţiilor de la stagiile avansate (decodorul).

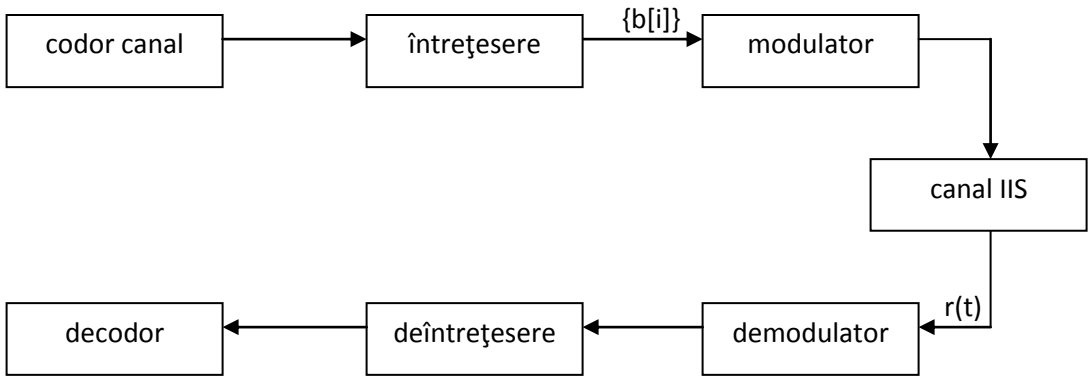


Figura 2.6 Comunicare codată pe un canal ce provoacă interferenţă inter-simbol

Pentru folosirea de procesare turbo în sistemul din figura 2.6, atât demodulatorul cât şi decodorul trebuie să fie de tipul probabilitate *a posteriori* maximă (MAP). Funcţia unui demodulator MAP este de scoate la ieşire decizii soft ce reflectă probabilitatea ca un bit să fie 0 sau 1. La a *l*-a iteraţie, informaţia folosită de demodulatorul MAP constă în semnalul recepţionat r(t) şi probabilităţile *a priori* a biţilor de intrare obţinute de la decodorul MAP care sunt calculate pe baza ieşirilor demodulatorului de la iteraţia l-1. Demodulatorul MAP foloseşte această informaţie, combinată cu informaţia despre modulaţia şi structura canalului ales, pentru a produce probabilităţile a posteriori (APP) a biţilor.

$$P^l(b[i]=1\,|\,r(t)) = \frac{p(r(t)\,|\,b[i]=1)P^{l-1}(b[i]=1)}{p(r(t))} \tag{16}$$

$$P^l(b[i]=0\,|\,r(t)) = \frac{p(r(t)\,|\,b[i]=0)P^{l-1}(b[i]=0)}{p(r(t))} \tag{17}$$

Considerând raportul de probabilitate logaritmică (LLR) format de probabilităţile a posteriori din (16) şi (17):

$$\Lambda_1^l(b[i]) = \log\frac{P^l(b[i]=1\,|\,r(t))}{P^l(b[i]=0\,|\,r(t))} = \log\frac{p(r(t)\,|\,b[i]=1)}{p(r(t)\,|\,b[i]=0)} + \log\frac{P^{l-1}(b[i]=1)}{P^{l-1}(b[i]=0)}$$

$$\lambda_1^l(b[i]) \qquad\qquad \lambda_2^{l-1}(b[i]) \tag{18}$$

Se vede din (18) că LLR-ul este suma a două cantităţi distincte, primul termen $\lambda_1^l(b[i])$, este informaţia extrinsecă produsă de subsistemul din primul stagiu din receptor (demodulatorul MAP), care este informaţie despre b[i] extrasă de demodulator din semnalul recepţionat r(t) şi probabilităţile a priori

a celorlalţi biţi transmişi, fără a folosi probabilităţile a priori b[i]. Al doilea termen, $\lambda_2^{l-1}(b[i])$, conţine probabilităţile a priori b[i]. Pentru prima iteraţie (l=1), se consideră $P^0(b[i]=1)= P^0(b[i]=0)=1/2$ ($\lambda_2^{l-1}(b[i])=0$ pentru orice i). Informaţia extrinsecă $\{\lambda_1^l(b[i])\}$ produsă de demodulatorul MAP este trimisă subsistemului din stagiul următor (decodorul MAP) ca informaţie a priori pentru decodare.

Bazat pe informaţia a priori primită de la demodulatorul MAP şi pe constrângerile codului, decodorul MAP calculează LLR-ul a posteriori pentru fiecare bit:

$$\Lambda_2^l(b[i]) = \log \frac{P(b[i]=1 \mid \{\lambda_1^l(b[i])\}; structură\ cod)}{P(b[i]=0 \mid \{\lambda_2^l(b[i])\}; structură\ cod)} = \lambda_2^l(b[i]) + \lambda_1^l(b[i]) \tag{19}$$

Ieşirea decodorului este suma dintre informaţia extrinsecă $\lambda_2^l(b[i])$ determinată de subsistemul din stagiul al doilea (decodorul MAP) şi informaţia anterioară $\lambda_1^l(b[i])$ de la stagiul anterior (demodulatorul MAP). Informaţia extrinsecă $\lambda_2^l(b[i])$ este apoi retrimisă la demodulatorul MAP ca informaţie a priori pentru următoarea (l+1) iteraţie. Trebuie remarcat că (18) şi (19) sunt adevărate doar dacă intrările demodulatorului sau decodorului sunt independente. Cum atât canalul IIS şi codorul sunt cu memorie, această presupunere de independenţă este invalidată, deci trebuie făcută întreţeserea (permutare în timp) între demodulator şi decodor pentru a asigura independenţa aproximativă. În sfârşit, structura receptorului turbo pentru un sistem IIS este ilustrată în figura 3.8. Această schemă poartă numele de egalizor turbo. Numele *turbo* este justificat deoarece demodulatorul şi decodorul folosesc pentru a scoate date la ieşire informaţii a priori de la iteraţia anterioară, similar funcţionării unui motor turbo.

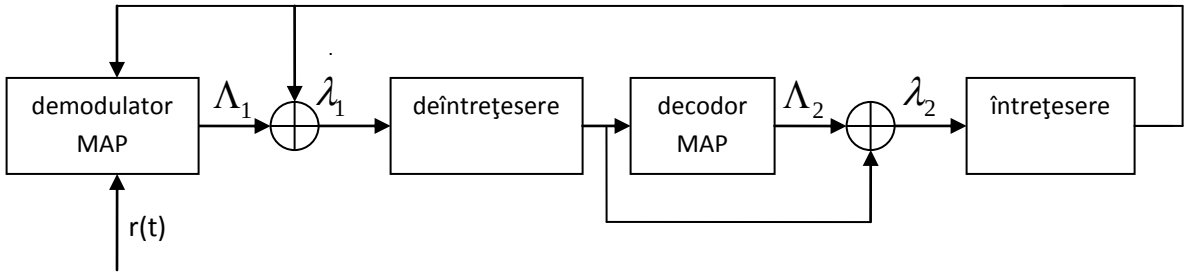


Figura 2.7 Receptor turbo pentru comunicaţie codată peste canal IIS

Principiul turbo poate fi deasemenea aplicat canalelor cu acces multiplu, rezultând detectoare turbo multiutilizator.

2.2.5 Decodarea codurilor Turbo

Pentru orice cod tradiţional simplu, ultimul pas de la decodor constă în a produce decizia hard asupra biţilor decodaţi (sau mai general a simbolurilor decodate). Pentru ca intr-o schemă concatenată un cod turbo să funcţioneze bine, algoritmul de decodare nu ar trebui să se limiteze la furnizare deciziilor

hard. Pentru a exploata cât mai bine informaţiile învăţate de la fiecare decodor, algoritmul de decodare trebuie sa efectueze un schimb de decizii soft decât unele hard. Pentru un sistem cu 2 componente de cod, conceptul care stă în spatele decodării turbo constă în a transmite decizii soft de la ieşirea unui decodor la intrarea celuilalt, şi să itereze acest proces de mai multe ori pentru a avea decizii mai bune. Aşadar un decodor pentru a putea fi component al unui turbo cod, trebuie să accepte intrare soft şi, de asemenea, să ofere ieşire soft. În continuare vor fi prezentaţi principalii algoritmi SISO (soft-input soft-output).

Algoritmii de decodare bazaţi pe trellis reprezintă metode recursive prin care se estimează secvenţa de date pe baza secvenţei recepţionate, pe baza unui criteriu de evaluare a distanţei dintre secvenţa recepţionată si toate secvenţele posibile prin diagrama trellis, distanţă ce trebuie minimizată. Sunt cunoscute două mari clase de algoritmi bazaţi pe trellis:

- familia de algoritmi de tip **MAP** (Maximum Aposteriori Probability). El maximizează logaritmul raportului de plauzibilitate condiţionată de o anumită secvenţă recepţiontă şi o anumită succesiune de stări prin trellis. Acest algoritm este optim din punct de vedere al minimizării probabilităţii de eroare de bit, fiind, din acest punct de vedere superior algoritmului Viterbi, oferind nu numai secvenţa estimată cît şi probabilităţiile ca fiecare bit din secvenţă să fie corect recepţionat. Dezavantajul său constă în faptul că complexitatea sa este mult mai mare decât a algoritmului Viterbi convenţional, deoarece trebuie să examineze toate căile posibile prin trellis, fără eliminările pe care algoritmul Viterbi le face la anumiţi paşi. Din această cauză au fost deduse un număr de versiuni simplificate care duc la performanţe mai bune şi la o complexitate mai mică (Max-Log-MAP, Log-MAP,etc).

- algoritmul **Viterbi** – minimizează probabilitatea ca o secvenţă incorectă prin trellis să fie aleasă ca fiind cea corectă în raport cu secvenţa recepţionată. Faţă de algoritmul Viterbi clasic, cel folosit în cazul decodării codurilor turbo prezintă două modificări, şi anume:

- modalitatea de calcul a metricilor căilor a fost modificată astfel încât să ţină seama de informaţia apriori obţinută de fiecare decodor de la perechea sa;
- algoritmul a fost modificat astfel încât să se obţină atât un estimat al secvenţei prin trellis cât şi o ieşire „soft" care să ofere celuilalt decodor o informaţie asupra metricii calculate;

A nu se confunda decodarea MAP sau SOVA (Soft Output Viterbi Algoritm) cu noţiunea de decodare iterativă. Algoritmii MAP şi SOVA sunt utilizaţi cu sistemele de codare bazate pe trellis în timp ce procesul iterativ se poate aplica oricărui tip de cod inclusiv codurilor bloc care nu sunt bazate pe trellis. În cazul codurilor turbo, implementarea decodorului este foarte costisitoare din punct de vedere al efortului de calcul; de aceea se preferă structuri de codare cu memorie mică (folosind registre de deplasare cu maxim 4 celule) şi dimensiuni relativ reduse ale circuitului de întreţesere, pentru a nu lucra cu blocuri de date foarte lungi (lungimea trellisului pentru care se determină parametrii fiind dictată de dimensiunea blocului de intrare)

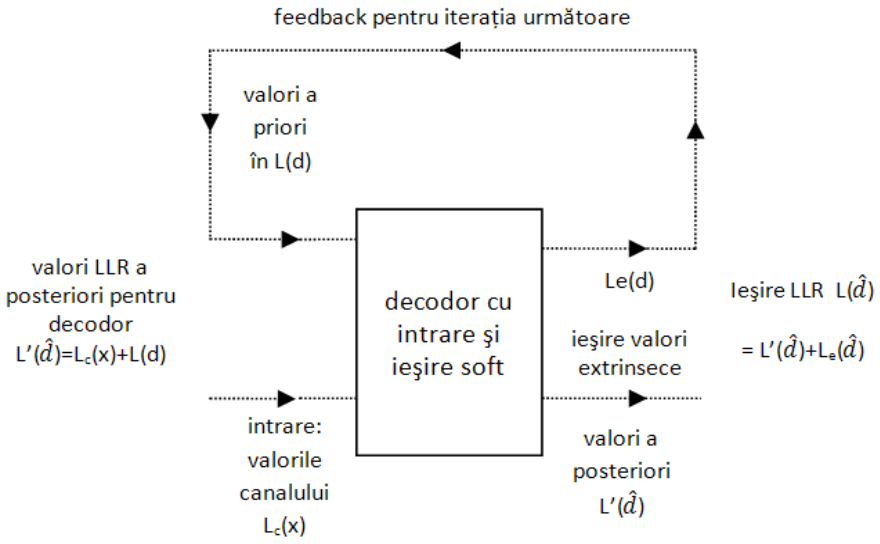


Figura 2.8 Decodor SISO (pentru un cod sistematic)

2.2.6 Principiile care stau la baza decodării iterative (Turbo)

Pentru prima iteraţie de decodare a decodorului SISO din figura 2.8, se presupune că datele sunt echiprobabile, generând o valoare apriori iniţială a LLR-ului egală cu 0 (L(d)=0). Ieşirea $L(\hat{d})$ a decodorului din fig 2.8 este formată din LLR-ul detectorului, L`(d) şi din LLR-ul extrinsec de la ieşire, $L_e(\hat{d})$, reprezentând informaţia învăţată din procesul de decodare. După cum este ilustrat în fig 2.8, pentru o decodare iterativă plauzibilitatea extrinsecă este întoarsă în intrarea decodorului, pentru a servi la îmbunătăţirea valorii apriori pentru următoarea iteraţie.

Să considerăm codul bidimensional reprezentat în fig 2.9. Configuraţia poate fi descrisă ca o matrice de date formată din k_1 linii si k_2 coloane. Fiecare din cele k_1 linii conţine un vector de cod format din k_2 biţi de informaţie şi n_2-k_2 biţi de paritate. În mod similar, fiecare din cele k_2 coloane conţine un vector de cod format din k_1 biţi de informaţie şi n_1-k_1 biţi de paritate. Diferitele părţi ale structurii sunt notate cu d pentru date, p_h pentru paritatea orizontală (de-a lungul liniilor) şi p_v pentru paritatea verticală (de-a lungul coloanelor). În plus, sunt blocuri notate cu L_{eh} şi L_{ev} care conţin valorile LLR extrinseci învăţate din paşii de decodare orizontală şi respectiv verticală. De reţinut că acest cod produs este un exemplu simplu de cod concatenat. Structura lui conţine 2 paşi de decodare diferiţi, orizontal şi vertical.

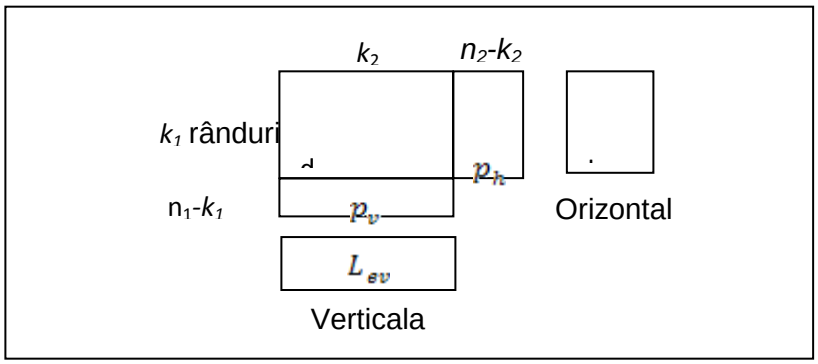


Figura 2.9 Cod produs bidimensional

Algoritmul de decodare iterativ al acestui cod produs acţionează după cum urmează:

1. Stabileşte informaţia apriori L(d)=0
2. Decodează orizontal şi folosind ecuaţia (15) se obţine informaţia extrinsecă orizontală după cum urmează:

$L_{eh}(\hat{d}) = L(\hat{d}) - L_c(x) - L(d)$

3. Stabileşte $L(d) = L_{eh}(\hat{d})$
4. Decodează vertical şi folosind ecuaţia (15) se obţine informaţia extrinsecă verticală după cum urmează:

$L_{ev}(\hat{d}) = L(\hat{d}) - L_c(x) - L(d)$

5. Stabileşte $L(d) = L_{ev}(\hat{d})$
6. Dacă au fost suficiente iteraţii pentru a genera o decizie de încredere, du-te la pasul 7; altfel du-te la pasul 2.
7. Ieşirea soft este:

$$L(\hat{d})=L_c(x)+L_{eh}(\hat{d})+L_{ev}(\hat{d}) \quad (20)$$

2.3 Coduri LDPC

2.3.1 Apariţia codurilor LDPC

Robert Gallager a introdus codurile de control al parităţii de mica densitate[5,6], la doar o decadă după ce opera lui Shannon a fost publicată[7]. Abordarea sa urma să formeze în cele din urmă baza pentru o clasă de coduri care se apropie foarte mult de legătura lui Shannon. Codurile lui Gallager, şi algoritmii iterativi de decodare, au fost oarecum trecuţi cu vederea de cei din domeniul codificărilor. În acea periodă, nu mulţi cercetătorii aveau la dispoziţia lor un calculator performant, aşa că munca lui nu a mai fost continuată. Ba mai mult, având o intuiţie puternică, abordarea lui Gallager era o provocare pentru paradigma de codificare într-o vreme în care

specialiştii puneau accent pe distanţa minimă. Adevăratul potenţial al controlului parităţii de mică densitate a fost redescoperit abia trei decade mai târziu, la jumătatea anilor 1990.

Un număr mic de cercetători au continuat să folosească codurile Gallager timp de câteva decade după publicarea tezei lui, vezi [8,9]. La începutul anilor 1980, Tanner a furnizat o reprezentare grafică a LDPC şi a altor scheme de codificare, care sunt cunoscute acum sub denumirea de diagrama Tanner[10]. El a propus ca problema decodificării să fie rezolvată prin descompunerea codurilor în coduri mai simple din structura lor şi a introdus noţiunea care este cunoscută în prezent sub denumirea de algoritm de decodificare a sumelor minime. Tanner a prevăzut de asemenea şi potenţialul enorm al paralelismului, permiţând codurilor mai simple sa fie procesate simultan datorită coplexităţii lor scăzute. De la descoperirea codurilor de control al parităţii de mică densitate au fost propuse mai multe variante de îmbunătăţire. Unele dintre aceste coduri, în special în cazul diagramelor foarte mari(între aproximativ 105 şi 107 biţi) pot să atingă performanţe până aproape de o sutime a unui decibel a legăturii lui Shannon[11].

2.3.2 Prezentarea unui model al sistemelor de comunicaţie

Un sistem de comunicaţie complet include numeroase părţi componente care pot ridica probleme interesante şi greu de rezolvat. Cele mai multe sisteme de comunicaţie moderne lucrează în domeniul digital, acesta oferind o mulţime de posibilităţi de procesare a semnalului. În general, pentru obţinerea de performanţe optime, codorul sursei, codorul de canal şi modulatorul ar trebui considerate ca făcând parte din aceeaşi funcţie a sistemului şi nu ca blocuri separate. Shannon[13] a demonstrat că partea de codare a sursei şi partea de codare de canal pot fi separate în unităţi individuale fără a se pierde la capitolul optimizare. Aceeaşi regulă însă nu se poate aplica şi pentru despărţirea funcţiilor de codare de canal şi modulare. De exemplu, modulaţia bazată pe codare trellis (TCM), care este o tehnică ce combină codarea de canal şi modulaţia, poate obţine performanţe mai ridicate decât metodele care separă codarea de canal şi modulaţia.

2.3.2.1 Modele de canal

Pentru studierea şi compararea codurilor de canal este folositor să considerăm modulatorul şi demodulatorul ca fiind părţi componente ale canalului. Rezultatul este un canal cu intrări şi ieşiri în timp discret, caracterizat de intrările şi ieşirile posibile şi de probabilităţile de tranziţie care fac legătura dintre intrări şi ieşiri. De asemenea, ieşirea poate să nu depindă numai de intrarea curentă ci şi de cele anterioare. În aceste cazuri vorbim despre canale cu memorie. Acest proiect are ca scop înţelegerea şi simularea codurilor LDPC. Îacest scop va fi folosit canalul cu zgomot aditiv Gaussian alb(additive white Gaussian noise - AWGN), acesta fiind un tip de canal cu o bună relevanţă practică. Aşa cum este sugerat şi de numele canalului ieşirea Y este modelată prin adăugarea unei variabile aleatoare distribuite Gaussian (G) la intrarea în canal X.

$$Y=X+G \tag{21}$$

G este o variabilă aleatoare distribuită Gaussian de medie zero şi varianţă a2. Pentru un anumit simbol X= x ieşirea din canal Y este o variabilă aleatoare distribuită Gaussian de medie x şi varianţă σ^2, dată de formula:

$$P_{Y|X}(y \mid x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y-x)^2}{2\sigma^2}} \tag{22}$$

unde p( ) reprezintă densitatea de probabilitate a unei funcţii de o variabilă aleatoare.

2.3.2.2 Coduri de canal

Scopul codării de canal este de a face mesajul transmis mai puţin susceptibil la zgomotul introdus de canal. Acest lucru este realizat prin adaugarea de informaţie redundantă structurată sevenţei transmise, deci crescând numărul de biţi transmişi. Există două tipuri primare de coduri folosite pentru a introduce această redundanţă: coduri bloc şi coduri convoluţionale. Un cod bloc transformă un bloc de informaţie de lungime k într-un bloc cod de dimensiune n, unde n>k. Un codor convoluţional este în schimb un codor ce introduce în mod continuu redundanţă unui şir de biţi de informaţie primiţi în mod continuu. Pentru ambele tipuri de coduri, rata de cod R este dată de numărul de simboluri de informaţie transmise pentru un simbol iniţial. Pentru coduri bloc aceasta este R=k/n . În acest proiect sunt folosite coduri binare deci fiecare simbol va putea lua numai valorile 0 sau 1. Datorită redundanţei adăugate de către codorul de canal, receptorul poate detecta şi corecta erorile introduse de canal.

Codurile de canal care sunt realizate în acest scop se numesc coduri corectoare de erori. Codurile LDPC fac parte din această categorie. Performanţa unui cod corector de erori poate fi masurată, de exemplu, în raportul semnal zgomot(SNR) necesar pentru a obţine comunicaţia cu o anumită eroare de rată maximă. De obicei SNR-ul necesar pentru a obţine o anumită rată de eroare este cu mult mai mic când sunt folosite coduri corectoare decât atunci când informaţia este transmisă fără codare. Totuşi există şi două dezavantaje date de folosirea codurilor corectoare de erori: măreşte numărul de accesări ale canalului, deci lărgimea de bandă, şi conduce la o complexitate crescută a implementării transmiţătoarelor şi receptoarelor. În secţiunea următoare sunt prezentate câteva limite şi restricţii fundamentale care sunt folositoare pentru analizarea performanţelor unui cod în perspectiva folosirii lui în cadrul unui sistem.

2.3.2.3 Limite de performanţă

Ca parte a teoriei matematice a comunicaţiei, Shannon a definit conceptul de capacitate de canal. Capacitatea unui canal reprezintă o măsura a cantităţii de informaţie ce poate fi transportată între intrarea X şi ieşirea Y a unui canal. Definiţia capacităţii este legată de definiţia matematică a informaţiei; informaţia medie mutuală între variabilele aleatoare continue X şi Y , în biţi, definite astfel:

$$I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} px,y(x,y)\log_2 \frac{p_{x,y}(x,y)}{p_x(x)p_y(y)}dxdy \quad (23)$$

unde $p_{x,y}(x,y)$, $p_x(x)$ şi $p_y(y)$ sunt densitatile de probabilitate pt *X* si *Y*.

Capacitatea canalului C este definită ca maximul valorii *I(X,Y)* pentru distribuţia de intrare $p_x(x)$ măsurată în biţi/canal:

$$C=maxI(X;Y) \quad (24)$$

Teorema codării(Shanon) de canal asigură o relaţie între capacitatea canalului C şi rata de transmisie a informaţiei R: Dacă avem o sursă cu un debit de informaţie R biţi/s şi un canal de capacitate C biţi/s şi dacă R < C, există un cod cu cuvinte de lungime n astfel încât probabilitatea unei erori de decodare să fie:

$$P_E \leq 2^{-nE(R)} \quad (25)$$

unde n este lungimea cuvântului de cod şi E(R) o funcţie nenegativă numită exponentul erorii.

Deci, indiferent de nivelul perturbaţiilor din canal se poate face transmisiunea cu o probabilitate a erorii oricât de mică.

Performanţele codorului LDPC sunt foarte bune tinzând spre limita teoremei lui Shannon. Se defineşte eficienţa spectrală: $ES=r_b$ bit/sec/hz, unde r_b este rata de transmisie şi B banda .

Eficienţa spectrală maximă

$$ES_{max}=\log_2 MR \quad (26)$$

Raportul semnal/zgomot

$$\frac{S}{N} = \log_2 MR\frac{Eb}{N0} \quad (27)$$

unde M este modulaţia, R rata codorului şi $\frac{Eb}{N0}$ reprezintă raportul dintre energia de bit şi densitatea spectrală de putere a zgomotului. Teorema lui Shannon spune că se poate realiza o probabilitate de eroare de bit oricât de mică prin codare-decodare dacă rb < C, unde C este capacitatea canalului

$$C=B\log_2(1+\frac{S}{N}) \quad (28)$$

Daca $r_b=C$

(29)

Rezultă că eficienţa spectrală maximă este:

$$ES_{\max} = \log_2(1 + ES_{\max}\frac{E_b}{N_0}) \Rightarrow \frac{E_b}{N_0} = \frac{2^{ES_{\max}} - 1}{ES_{\max}} \tag{30}$$

şi reprezintă raportul minim dintre energia de bit şi densitatea spectrală de putere a zgomotului necesar pentru transmisie. Pentru canalul discret AWGN cu intrări şi ieşiri continue capacitatea canalului este dată de relaţia :

$$C = \frac{1}{2}\log_2(1 + \frac{\sigma^2{}_X}{\sigma^2}) \text{ biţi/canal} \tag{31}$$

unde puterea la intrarea canalului este limitată de $E[X^2] \leq \sigma^2{}_x$ si σ^2 este varianţa zgomotului. În practică forma de undă folosită este limitată de bandă. Deci rata de transmisie nu poate fi crescută oricât prin accesarea oricât de des a canalului. O cerinţă fundamentală pentru comunicaţie este dată de:

$$\frac{E_b}{N_0} > \frac{1}{\log_2 e} = \ln 2 \tag{32}$$

deci nu este posibil să se conceapă un sistem cu o rată arbitrară scăzută de erori dacă $E_b/N_0 < 0.7$ dB. Pe de altă parte, atât timp cât condiţia este îndeplinită, teoretic se pot atinge probabilităţi de eroare scăzute folosind coduri suficient de mari. Această restricţie impune o limită inferioară în ceea ce priveşte raportul energie per bit pe densitatea spectrală a zgomotului (E_b/N_0), limită ce trebuie respectată pentru a putea obţine o comunicaţie fără erori. Oricum, presupunerile folosite pentru a obţine aceasta limită sunt prea optimiste pentru cele mai multe situaţii reale. Cea mai importantă diferenţă între descriererea teoretică şi realizarea practică este dată de faptul că banda canalului este, în realitate, limitată. Din această cauză raportul E_b/N_0 necesar pentru o comunicaţie bună, chiar folosind coduri foarte mari, este mai mare decât 0.7 dB.

2.3.3. Codurile bloc liniare

Codurile de control al parităţii de densitate mică fac parte din categoria codurilor bloc liniare. Pentru o mai bună înţelegere vom defini proprietăţile codurilor cu diagrame liniare:

Sursa informaţiei Împărţim o secvenţă cu sursa primară într-un şir de vectori de lungime k înainte de codificare. Un astfel de vecor se notează cu u.

Codarea se face pe baza lungimii unei diagrame aplicând u pe un şir de vectori cifrat cu x. Un cod (n,k) are lungimea de cod n.

Cod sistematic Un cod în care informaţia vector apare ca o parte a cuvântului de cod sistematic. Segmentul de informaţie din cuvântul de cod este numit $x_p = u$ şi lungimea n-k biţi de paritate adăugaţi x_p.

Proprietatea de liniaritate Codul C este un cod liniar binar dacă şi numai dacă C formează un subspaţiu vectorial peste+. Deci, suma oricăror două cuvinte de cod trebuie să fie în sine un cuvânt de cod.

Dimensiunea codului este dimensiunea spaţiului vectorial care îi corespunde. Un cod binar (n,k) are dimensiunea k, şi astfel are un total de 2^k cuvinte de cod, fiecare de lungime n.

Cuvânt de cod diferit de zero O consecinţă directă a proprietăţii de liniaritate este aceea că numele de cod total=zero , x=0, este un membru al oricărui cod liniar.

Coeficientul codului Coeficientul codului este r=k/n. Coeficientul codului indică proporţia de informaţie transferată pe canal utilizat.

Matricea Generatoare Proprietatea de liniaritate implică existenţa unei baze pentru cod. Să construim o matrice generatoare pentru cod, denumită G, folosind baza independentă de cuvântul de cod k pe post de şir de vectori G. Matricea generatoare are dimensiunea k x n şi poate fi folosită pentru a codifica vectorul de informaţie. Să considerăm forma sistematică $G_{sys}=[P|I_k]$ în aşa fel încât biţii de paritate ai cuvântului de cod să preceadă biţii de informaţie. Aici $[P|I_k]$ denotă legatura matricei unitare k x k cu P. Aşadar, codificarea este realizată respectând $x=[x_p|x_u]=uG$. Matricea generatoare descrie structura codului, totuşi nu este definită numai pentru cod.

Matricea Controlului – Parităţii Un alt mod de a descrie structura codului, este furnizat de matricea de control al parităţii, denumită H. În general, H are dimensiunea m x n, unde m=n-k. Toate cuvintele de cod trebuie să respecte condiţia $Hx^T=0$. Matricea de control al parităţii nu este definită numai pentru cod şi este legată de G prin $HG^T=0$. Forma sistematica a lui H poate fi obţinută din G, şi viceversa, folosind $H_{sys}=[I_m|P^T]$.

Distribuţia greutăţii Definim vectorul de greutate x, W(x), ca numărul elementelor diferite de zero pe care le conţine. Distribuţia greutăţii unui cod este o listă cu numărul de cuvinte la fiecare greutate, pentru toate greutăţile cuvântului cod.

Distanţa Hamming dintre două cuvinte cod este numărul poziţiilor în care ele diferă.

Distanţa minimă se notează cu d_{min} şi este cea mai mică distanţa Hamming dintre două cuvinte de cod din mulţimea tuturor cuvintelor. Având în vedere că un cuvânt de cod nul (cu toti biţii zero) este un membru al fiecarui cod liniar, distanţa minimă a unui cod liniar este egală cu greutatea cuvântului de cod care are cea mai mică greutate. În general, un cod cu greutatea minimă d_{min} poate corecta $[(d_{min}-1)/2]$ greşeli.

În timp ce şirurile matricei generatoare sunt cuvinte de cod, şirul cu cea mai mică greutate reprezintă o legatură superioară simplă sau o distanţă minimă. Teorema următoare se leagă de structura codului prin matricea de control al parităţii[14].

Teorema (Massey). Distanţa minimă a unui cod cu diagrama liniară binară este egală cu numărul minim de coloane diferite de zero în matricea sa de control al parităţii care însumează zero.

2.3.4. Codurile de Control al Parităţii cu Densitate Mică

Un cod de control al parităţii cu densitate mică(LDPC) este un cod cu o matrice liniară care are o matrice de control al parităţii slab populată. În acest capitol vom prezenta evoluţia codurilor LDPC, începând de la codurile prezentate prima dată de Gallager până la posibilitatea recentă de abordare a structurii.

Considerând un cod LDPC regulat cu matricea de paritate H dată :

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 2.10 Matrice de control a parităţii

Atribuim fiecărui bit al cuvântului de cod o variabilă, v_s:s ϵ {1..m}, unde fiecare variabilă corespunde de asemenea unei coloane H. Aşadar, fiecare şir din H reprezintă o restricţie a controlului de paritate cu codul, c_r:r ϵ {1..m}.

Participarea variabilei v_s la controlul restricţiei este sugerată de un element diferit de zero de pe poziţia (r,s) în H. Nodurile variabile si cele de control pot de asemenea să aibă valori asociate cu ele. Simbolurile vs şi cr sunt folosite atât pentru a denumi aceste noduri cât şi pentru a reprezenta valorile. Denumim numărul de elemente diferite de zero din şir, sau coloană, ca fiind greutate.

Fiecare linie din H reprezintă o constrângere a controlului de paritate. Deci, urmează mulţimea completă de restricţii, presupunând aritmetica binară. Dacă valorile atribuite mulţimii de variabile reprezintă un cuvânt de cod valid, atunci c_r=0 pentru toate r ϵ {1..m}.

$$v1+v3+v5+v7=c1 \quad (33)$$

$$v1+v4+v6+v8=c2 \quad (34)$$

$$v2+v3+v6+v7=c3 \quad (35)$$

$$v2+v4+v5+v8=c4 \quad (36)$$

Gallager defineşte o structură regulată pentru codurile de control al parităţii cu densitate mică după cum urmează:

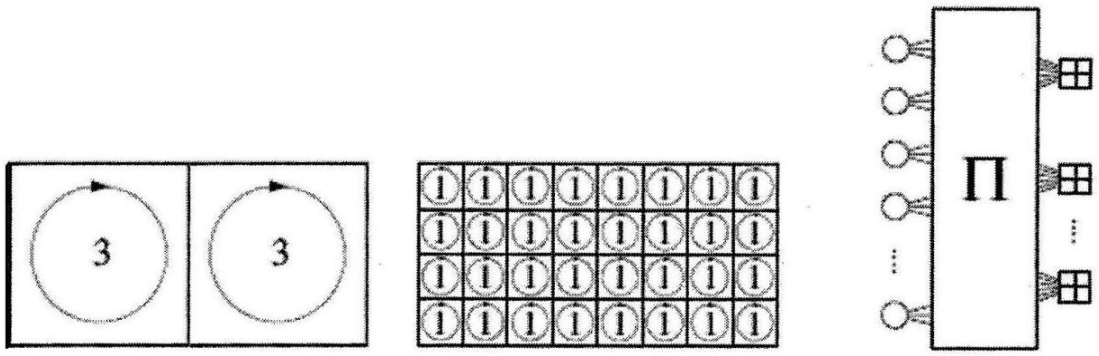
Definiţia 1. Codul regulat LDPC

Un cod regulat LDPC cu parametrii (n,j,i) este un cod bloc de lungime n care are o matrice de control al parităţii cu greutăţi pentru rând şi coloană de exact j pe coloană şi i pe linie.

Prin contrast, codurile neregulate LDPC permit variabilelor să participe în numere diferite de controale şi fac posibilă aplicarea restricţiilor controlului la numere diferite de variabile. Un cod neregulat LDPC poate fi descris de distribuţiile greutăţii vectorului de linie şi de coloană ale matricei sale de control al parităţii.

După cum am spus şi mai devreme codul dat ca exemplu în fig. 2.10 are o structură regulată(8,2,4). Este foarte utilizată denumirea de coduri regulate LDPC fără a se specifica lungimea în serie, ex(2,4)-regulat. După cum vom vedea în secţiunea de codificare, eficienţa acestor coduri este permisă de proprietatea matricei de control al parităţii de a fi slab populată.

Există multe variante de a construi o matrice de control al parităţii p(j,i). O variantă des folosită este de a aduna sau a suprapune matricele permutărilor aleatorii. În fig. 2.11 folosim notatia din[15], unde un număr încercuit reprezintă acel număr al matricei permutărilor aleatorii suprapuse (nu îmbinate). Figura 2.11(a) oferă un exemplu de construcţie regulată (3,6). Figura 2.11(b) arată cum pot fi adunate matricile permutărilor aleatorii în aşa fel încât să rezulte un cod regulat(4.8). Figura 2.11(c) ne arată cum putem de asemenea vizualiza construcţia suprapusă sub forma unei permutări aleatorii de legături între muchii:



(a) (b) (c)

Fig. 2.11 Moduri de construcţie ale matricilor circulare

Un total de coduri rămase n cu gradul j reprezintă variabilele şi nodurile drepte m de gradul j reprezintă restricţiile controlului.

Dacă coloanele lui H sunt independent liniare atunci coeficientul codului este r =(i-j)/i, în caz contrar coeficientul este r=(n-i-I)/n, unde I este dimensiunea spaţiului H al linei peste câmpul binar care este folosit aici.

Există rezultate importante în cazul codurilor LDPC regulate generate aleator care au greutatea coloanei j>=3[2,10]. În primul rând distanţa minimă a unui astfel de cod creşte liniar cu lungimea diagramei. În al doilea rând, dacă este decodificat de un decodor optim şi dacă acel coeficient diferit de zero al codului este sub vreun coeficient maxim, un astfel de cod va dezvolta o probabilitate de eroare arbitrară mică în limita în care lungimea diagramei tinde la infinit.

2.3. 5. Reprezentarea grafică a coeficientului codurilor LDPC

Folosim termenul cod compus pentru a clasifica orice cod care poate fi divizat în subcoduri constituente, de exmplu, un cod LDPC[16]. Reînvierea interesului pentru structurile codului compus şi algoritmii iterativi de decodare au dus la noi reprezentări pentru coduri în grafuri. Acestea ne oferă o unealtă folositoare pentru reprezentarea codului şi iau parte la analiza de decodificare a algoritmilor.

Să luăm o funcţie globală, de exemplu, o funcţie a tururor variabilelor n, g (x1,...,xn). Presupunem că această funcţie globală se descompune într-un produs de funcţii locale, $f_r(X_r)$. Aici rϵR este indexul funcţiei locale şi X_r reprezintă submulţimea variabilelor care participă la f_r .

Rezultatul va fi:

$$g(x_1,...,x_n)=\prod_{r\in R} f_r(X_r) \tag{37}$$

Definiţia 2. Graful coeficientului.

Un graf al coeficientului este un graf bipartiţionat care reprezintă factorizarea descrisă în (37). Graful constă în variabile şi noduri de decodificare. Există un nod variabil pentru fiecare variabilă x_s, astfel încat sє{1...n}. Există un nod de decodificare pentru fiecare funcţie locală fr. Legătura se află în cadrul diagramei între nodul variabil x_s şi nodul funcţional f_r dacă şi numai dacă $x_s \epsilon X_r$.

Un exemplu simplu ar fi să considerăm o funcţie globală de valoare reală cu trei variabile, g(x1,x2,x3), care poate fi reprezentată ca produsul a două funcţii locale f1 şi f2 după cum urmează:

$$g(x1,x2,x3)=f1(x1,x2,x3)f2(x1,x2,x3) \tag{38}$$

Reprezentarea diagramei coeficientului pentru (38) este redată în figura (2.12). Nodurile variabile şi nodurile funcţionale sunt reprezentate prin cercuri respectiv cutii. Nodul variabil x_s pentru sє{1,2,3}, este conectat printr-o muchie la nodul funcţional f_r , pentru rє{1,2,3}, dacă şi numai dacă x_s este o variabilă independentă a f_r.

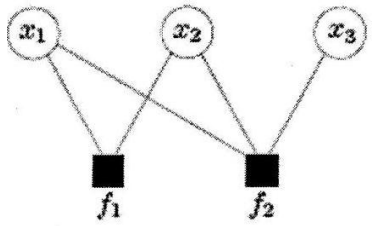


Fig. 2.12 Structura diagramei bipartiţionate a coeficientului

Structura diagramei bipartiţionate a coeficientului reprezintă foarte bine codurile LDPC. Nodurile funcţionale reprezintă restricţiile controlului de paritate ale codului şi se mai numesc prin urmare şi noduri de control. Structura regulată descrisă în fig.(2.10) este reprezentată prin următoarea diagramă:

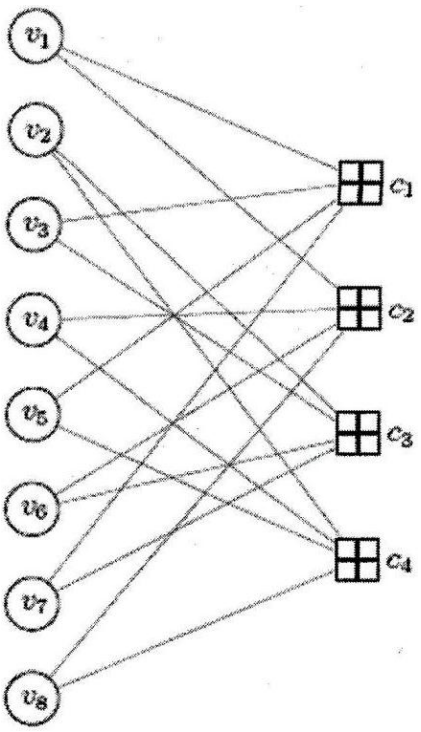


Fig. 2.13 Diagrama ascociată matricei de verificare a parităţii

Matricea H indică muchiile (bidirecţionale) ale diagramei în funcţie de poziţia elementelor sale diferite de zero. În acest caz, nodurile variabile n=8, fiecare cu gradul j=2, reprezintă simboluri ale cuvintelor de cod. Nodurile de control m=4, fiecare cu gradul i=4, reprezintă controlul de paritate.

Să luăm un cod C de lungimea n cu matricea de control de paritate H. Să ne amintim că un cuvânt de cod valid x respectă Hx = 0, şi că fiecare linie din H reprezintă o restricţie separată a controlului de paritate. Atribuim o funcţie locală de indicare fiecărei restricţii de control. Funcţia locală asociată cu controlul c_r este îndeplinită când participă la control toate variabilele, exemplu setul Xr, suma zero pe F2, de exemplu $|\sum_{x_S \in X_r} x_S = 0|$

Aici folosim convenţia lui Iverson[17] pentru a arăta gradul de adevăr al unei declaraţii S. Dacă S este adevărat atunci [S] = 1, în caz contrar [S] = 0. Deci, când funcţia locală este îndeplinită aceasta îşi însuşeşte valoarea 1, în caz contrar are valoarea 0. În concluzie, rezultatul acestor funcţii locale de indicare se numeşte funcţia globală de indicare. Vectorul x este un cuvânt de cod valid, care corespunde pe diagrama coeficientului unei configuraţii valide, dacă şi numai dacă îndeplineşte funcţia globală de indicare a codului. Aşadar, funcţia globală se mai numeşte funcţia ce aparţine unei mulţimi de coduri şi este notataă prin [(xi,... ,xn)∈ C] în cazul unui cod C de lungimea n. Să luăm exemplul simplu de cod dat în (1) cu grupul de restricţii listat în (33-36). Folosind aritmetica în loc de F2, funcţia globală de indicare pentru acest cod este dată de:

$$[(x_1...x_8)\epsilon C]=[x_1+x_3+x_5+x_7=0][x_1+x_4+x_6+x_8=0]$$

$$[x_2+x_3+x_6+x_7=0][x_2+x_4+x_5+x_8=0] \quad (19)$$

Tanner a introdus pentru prima dată ceea ce este azi cunoscut sub denumirea de reprezentarea prin diagrama de coeficient a codurilor lui Gallager [10]. El a prevăzut de asemenea şi potenţialul ce se dezvolta în paralel, acela de a rupe codurile lungi în componente constituente mai mici pentru a fi decodificate. Wiberg, Loeliger şi Kotter [18,19] au adăugat la diagrama lui Tanner, reprezentarea variabilei de stare, făcând posibilă astfel reprezentarea codurilor turbo şi trellis. Aplicaţiile diagramei de coeficient se extind mult în afara domeniului codificărilor pentru controlul greşelilor, şi pot fi folosite pentru a descrie multi alţi algoritmi, cum ar fi filtrul Kalman şi anumiţi algoritmi de transformare Fourier rapidă (TFR)[20].

Capitolul IV. Codarea LDPC

Codurile de densitate mică pentru controlul parităţii reprezintă o clasă a codurilor bloc liniare. Astfel ele pot fi codificate folosind matricea generatoare însă în absenţa unei structuri în interiorul codului, pe măsura ce dimensiunea blocului de date creşte această metoda impune decodorului stocare mare şi cerinţe legate de procesare. Deşi H este construit ca sa fie slab populat, matricea generatoare a codului, G, este încărcată în general. De fapt,daca G ar fi rar atunci nu ne-am fi aşteptat ca acel cod să fie o distanţă minimă bun. Să ne amintim că şirul cu cea mai mică greutate din G e limitat de d_{min} . Aşadar codoarele pot deveni considerabil mai lente şi mai mari când lungimea blocului n este crescută, având în vedere că înmulţirea matricei-vector are complexitatea $O(n^2)$. Acest lucru a dirijat cercetarea spre găsirea unor codoare eficiente din punct de vedere al calculului şi a codurilor structurate care necesită implementarea unui codificator de complexitate redusă.

În acest capitol sunt recapitulate mai multe propuneri de construire a unei arhitecturi centrate pe codificator pentru codificarea liniară în timp a codurilor LDPC. Mai întâi se face o recapitulare a metodelor existente care pot fi folosite la construirea codurilor structurate, care sunt proiectate special pentru a permite implementarea simplă a codificatorului. Apoi este descrisă o tehnică ce poate fi folosită pentru a transforma codurile, astfel încât un codificator cu o complexitate liniară de timp aproximativă să poată fi construit. Următorul argument al acestei abordări este să se reducă mărimea per total a sistemului de comunicaţii prin îndeplinirea ambelor funcţii de către un singur circuit având o bază de schimbare a timpului.

4.1Coduri Structurate

Structura codurilor LDPC cu matrice de control al parităţii care au o structură aproape triunghiulară a fost propusă de MacKay ş.a. [21]. Această metodă permite ca cea mai mare parte din biţii de paritate să fie calculaţi în timp liniar folosind operaţii puţine, de exemplu funcţiile care nu implică decât un număr mic de variabile. Experimentele lor arată că astfel de coduri au o performanţă care e comparabilă cu cea a codurilor regulate LDPC. O matrice triunghiulară de control al parităţii este exemplificată în figura 4.1.

Se presupune că cuvintele de cod sunt aranjate în şiruri de vectori $x = [x_p \mid x_u]$, unde x_u sunt biţii de informaţie şi x_p sunt biţii de paritate. În aproape acelaşi fel este partiţionată şi matricea de control al parităţii, $H = [H_p \mid H_u]$. Vectorul intermediar b este definit $b = H_u x_u^T$, care poate fi evaluat în timp liniar prin calculul matricei-vector. Când H_p are formă triunghiulară putem calcula x_p în timp liniar folosind H_p şi b, prin substituţie regresivă. În cazul exemplului precizat mai sus acest lucru poate fi realizat în următorii trei paşi:

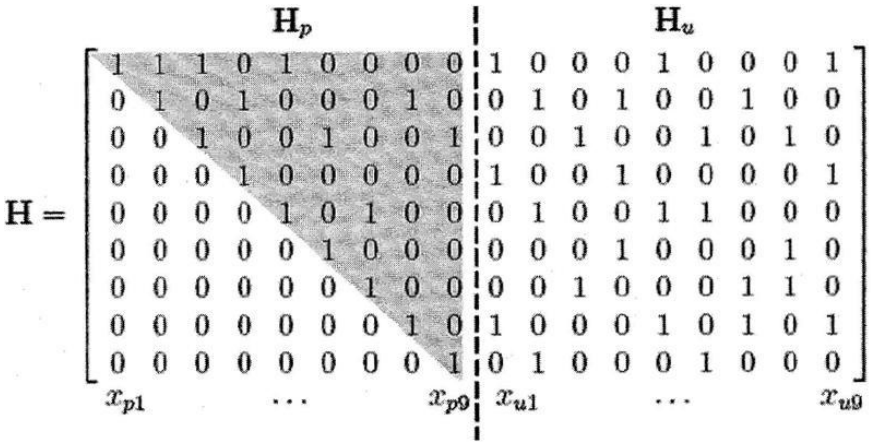


Fig. 4.1. O matrice triunghiulară superioară a controlului de paritate

1. $x_{p4} = b_4$, $x_{p6} = b_6$, $x_{p7} = b_7$, $x_{p8} = b_8$, $x_{p9} = b_9$
2. $x_{p2} = x_{p4} + x_{p8} + b_2$, $x_{p3} = x_{p6} + x_{p9} + b_3$, $x_{p5} = x_{p7} + b_5$
3. $x_{p1} = x_{p2} + x_{p3} + x_{p5} + b_1$

Un caz special de triunghiularitate este întâlnit atunci când Hp are structura dual diagonală expusă în Figura 4.2. Structura a fost prima dată propusă de Ping ş.a.[22], şi a reapărut recent în literatură[23,24]. Aici substituţia regresivă urmează o structură simplă şi poate fi efectuată folosind un acumulator.

$$\mathbf{H}_p = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 4.2 O structură în formă de s pentru H

Se observă că dacă matricea de control al parităţii are vreo coloană de greutatea j>1, şi este strict triunghiulară, atunci structura codului este neapărat neregulată.

4.2Structuri de Cod Ciclic

O altă variantă de a asigura codificarea liniară de timp foloseşte structurile ciclice şi cvasi-ciclice[25,26]. Un cod ciclic are proprietatea că orice deplasare ciclică a unui cuvânt de cod este în sine un cuvânt de cod. Un cod cvasi-ciclic are proprietatea că, pentru mărimea fixă de deplasare, o deplasare ciclică a oricărui cuvânt de cod de acea mărime este în sine un cuvânt de cod. Putem construi matricea de control al parităţii pentru un cod LDPC cvasi-ciclic prin unirea pe orizontală a matricelor rare circulare, fiecare având dimensiunea m x m[27].

Definiţie. Matricea Circulară

O matrice circulară este o matrice pătrată în care fiecare şir este o deplasare ciclică a şirului anterior. Un exemplu de matrice de control al parităţii pentru un cod cvasi-ciclic având n=18 şi k=12, şi deci r=2/3 , este dat în figura 5.3. Această matrice poate fi rearanjată cu uşurinţă în forma cvasi-ciclică prezentată în [28] prin permutarea coloanelor sale în ordinea 1, m + 1, 2m + 1, 2, m + 2, 2m + 2, ...m, 2m, 3m. Se observă că permutarea liniilor şi coloanelor H nu schimbă structura codului ci doar redenumeşte controalele sau respectiv variabilele.

$$\mathbf{H} = \left[\begin{array}{cccccc|cccccc|cccccc} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}\right]$$

$$\mathbf{H}_p \qquad \mathbf{H}_{u1} \qquad \mathbf{H}_{u2}$$

Fig. 4.3 O matrice cvasi-ciclică de control al parităţii

Împărţim H în trei componente circulare m x m. Aici există un total de componente ale matricei n_0=3, iar k_0=2 din acestea corespunde biţilor de informare ai cuvântului de cod. Mai exact, H_p corespunde biţilor de paritate, în timp ce H_{u1} şi H_{u2} corespunde biţilor de informare. Dacă H_p nu este singular atunci putem începe prin a multiplica H cu H_{p-1} pentru a obţine forma sistematică $H_{sys} = [I_m|H_{u(sys)}] = [I_m \mid H_{p-1}H_{u1} \mid H_{p-1}H_{u2}]$. Apoi permutăm coloanele aparţinând $H_{u(sys)}$ în ordinea 1, m + 1, , 2, m + 2 ...m, 2m pentru a redefini biţii de informare corespunzător. Acest loc apare într-o formă sistematică cvasi-ciclică dupa cum e arătat în figura 4.4.

Se observă că fiecare şir aparţinând $H_{u(sys)}$ este o deplasare ciclică corectă, prin locurile k_0. Acest lucru permite codurilor cvasi-ciclice să fie codificate folosind aceeaşi metodă ca aceea folosită de codurile ciclice[28]. Pentru a genera biţii de paritate pentru codul de mai sus folosim arhitectura registrului de deplasare regresivă a stadiului k din figura 5.5. În această diagrama vectorul sursa, u, este luat din stânga şi plasat în dreapta,

$$\mathbf{H}_{sys} = \left[\begin{array}{cccccc|cccccccccccc} & & & \mathbf{I}_m & & & & & & & & \mathbf{H}_{u(sys)} & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ \hline x_{p1} & & \cdots & & & x_{p6} & x_{u1} & & & & & \cdots & & & & & & x_{u12} \end{array}\right]$$

Fig. 4.4 Forma sistematică cvasi-ciclica a H

începând cu bit-ul u_I. Fiecare cutie reprezintă o celulaă de memorie care păstrează valoarea încărcată de obicei până când este aplicată o deplasare, stadiu în care ia valoarea celulei precedente. Schimbătorul s este iniţial conectat la o sursă de informaţie în serie, pentru a încărca vectorul sursă în registru. Această operaţie are loc în k deplasări. Primul bit de paritate este apoi calculat printr-o operaţie XOR pe biţii de informaţie selectati, aleşi de ramurile paralel ale codorului cu registru de deplasare. Poziţiile ramurilor sunt setate pe poziţia (inversată) de termeni zero în primul şir de $H_{u(sys)}$. Schimbătorul este apoi închis în poziţia iniţiată şi datele sunt deplasate ciclic de k_0 ori. Apoi următorul bit de paritate este calculat. Acest proces este repetat până când toti biţii de paritate m vor fi generaţi.

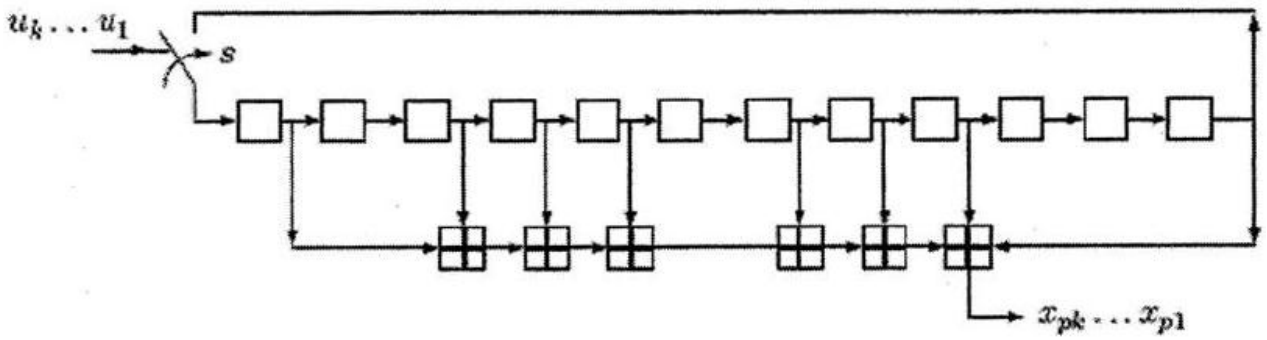


Fig. 4.5 Codor cu registru de deplasare pentru un cod cvasi-ciclic

Se poate extinde matricea de control al parităţii pentru a construi coduri de un randament mai mare prin adunarea orizontală a altor matrice circulare. În cazul particular al n_0 matrici circulare adunate lungimea diagramei este $n = mn_0$. Deci, alegerea unei rate de cod este restrânsă de relaţia $r = (n_0 - 1)/n_0$.

4.3Alte Structuri de Cod

Sipser şi Spielman au propus o clasă de coduri de expansiune decodabilă şi codificate în timp liniar in [29]. Abordarea lor implică folosirea diagramelor în cascadă pentru a construi recursiv coduri bazate pe subcoduri simple la fiecare stadiu al diagramei. Codurile corectoare de erori sunt construite combinând recursiv coduri mai slabe de reducere a erorii. Au dovedit că în cazul în care codurile de reducere a erorii sunt codificabile/decodificabile în timp liniar, această proprietate va fi păstrată până la obţinerea codului final corector de erori [30]. Luby ş.a. au construit coduri bazate pe aceste structuri care au o performanţă foarte bună [31].

Echard şi Chang au prezentat o variantă de construire a codurilor LDPC srtucturate cvasi-regulat, numită codurile de rotaţie [32,33]. Matricea de control al parităţii pentru aceste coduri este construită din componente ale matricelor de permutare şi rotaţiile lor. Ele ar putea fi codificate în timp liniar folosind un circuit codificator bistabil.

Zhang şi Parhi au prezentat un desen codec pentru codurile LDPC regulate [34]. Algoritmul lor de codificare este similar cu acela prezentat în secţiunea următoare. Totuşi, ei mai degrabă construiesc codul în aşa fel încât să optimizeze viteza codificatorului, decât să transforme un cod existent.

4.4 Transformarea Matricii H

Richardson şi Urbanke au arătat că H poate fi manipulat la cea mare parte a codurilor LDPC, astfel încat coeficientul termenului pătratic în complexitatea codificării este foarte mic [35]. Aceasta este o soluţie generală la problema codificării eficiente în timp, deoarece nu se bazează pe o nouă structură de cod dar mai degrabă restructurează orice matrice de control al parităţii existente (non-singură şi rară). Motivat de avantajele calculului triunghiular, H este mai întâi rearanjat în forma de triunghi aproximată arătată în figura 5.6. Acest lucru este realizat prin exploatarea spaţiului liber al H, folosind doar reordonarea coloanelor şi liniilor, şi deci densitatea matricei rămâne neschimbată.

Distanţa, g, este definită ca diferenţa dintre numărul de linii din forma aproximativă triunghiular-inferioară şi numărul de randuri în H. Codificarea complexităţii este proporţională cu $n + g^2$. Prin urmare problema de a reduce complexitatea codificărilor devene astfel una cu g redus. Pentru un cod LDPC regulat numărul clar de operaţii cerute nu este mai mare de $0.0172n^2 + O(n)$. Deci până şi cea mai mică durată constantă admite aproximativ complexitatea codificării liniare chiar şi în cazul unor durate lungi. Un alt rezultat cheie este că toate codurile LDPC probabil bune [36] au distanţe foarte mici, tipic pe o arie de la 1 la 3, şi în consecinţă au complexitate de codificare liniară.

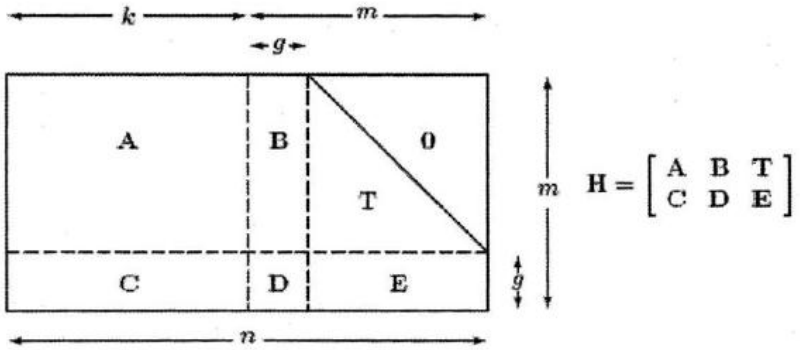


Fig. 4.6 Matricea de control al parităţii rearanjată într-o formă aproximativ triunghiular-inferioară

Matricea e împărţită în şase componente după cum se vede. Matricea T are o forma triunghiular-inferioară şi toate elementele diagonale ale T se unesc devenind unul. Să considerăm secţia de paritate a cuvântului de cod sistematic ca fiind împărţită în două segmente, de exemplu, $x = [u|x_{p1}|x_{p2}]$. Biţii de paritate sunt generaţi după cum urmează.

4.5 Calcul Anterior

Se calculează înainte matricea de densitatea g x g Ø= $-ET^{-1}B + D$ şi calculaţi inversul ei. De observat că H trebuie să fie în faţa şirului pentru ca Ø să fie reversibil.

Se calculează primul vector de paritate xp1:

1. $z_1 = Au^T$ (înmulţire împrăştiată în timp liniar)
2. $z_2=T^{-1}z_1$. Având în vedere că T este triunghiular-inferior şi împrăştiat această operaţie poate fi efectuată prin înlocuire regresivă în timp liniar.
3. $z_3=-Ez_2$ (multiplicare împrăştiată în timp liniar).
4. $z_4=z_3+Cu^T$ (multiplicare împrăştiată în timp liniar şi adunare).
5. $x_{p1}=Ø^{-1}z_4$ (multiplicare prin matricea de densitatea g x g).

Se calculează al doilea vector de paritate x_{p2}:

1. $z_5=Bx_{p1}^T$(multiplicare împrăştiată în timp liniar).
2. $z_6=z_1 + z_5$ (adunare în timp liniar).
3. $X_{p2}=-T^{-1}z_6$ (înlocuire regresivă în timp liniar).

Capitolul V. Decodarea LDPC

5.1 Decodarea iterativă

Cuvântul de cod transmis, x, este expus zgomotului provenit de la canal. Decodorul oferă un estimat al informaţiei transmise, pe baza restricţiilor codului şi a vectorului y recepţionat. În acest subcapitol sunt prezentate câteva abordări ale decodificărilor care folosesc trecerea iterativă a mesajelor pe diagrama de coeficient a codului. Valorile mesajului sunt calculate la fiecare nod pe baza restricţiilor de cod locale. H este calculat uşor datorită numărului scăzut de componente.

Algoritmii iterativi vin in două versiuni: algoritmul sumă-minimă şi algoritmul sumă-produs. Aceşti algoritmi sunt mai degrabă nişte generalizări ale algoritmului Viterbi sau ale altor algoritmi bazaţi pe trellis. Un alt caz special îl reprezintă algoritmul lui Gallager de decodare a codurilor cu densitate mica si control al parităţii. O formulare relativ generală a algoritmilor a fost dată de Tanner.

Structura generală a algoritmilor si contextul în care se aplică este ilustrat in Fig. 6.1. După cum se poate observa, algoritmii nu iau decizii, în schimb calculează un set de *funcţii*

finale de cost pe baza cărora se va lua o decizie finală. Ieşirea canalului intră in algoritmi ca un set de *funcţii locale de cost* iar scopul algoritmilor este de a concentra, pentru fiecare variabilă, toată informaţia din ieşirea canalului care este relevantă pentru acea variabilă.

Oficial, există o singură funcţie locală de cost pentru fiecare variabilă *s* ϵ *N*, exprimată de $\gamma_s : A_s \to \mathscr{R}$(unde R reprezintă mulţimea numerelor reale), si una pentru fiecare set de controale *E* ϵ *Q*, exprimată de $\gamma_E : W_E \to \mathscr{R}$. În mod similar, avem o funcţie de cost finală pentru fiecare variabilă *s* ϵ *N*, exprimată de $\mu_s : A_s \to \mathscr{R}$, şi una pentru fiecare set de controale, dată de $\mu_E : W_E \to \mathscr{R}$.

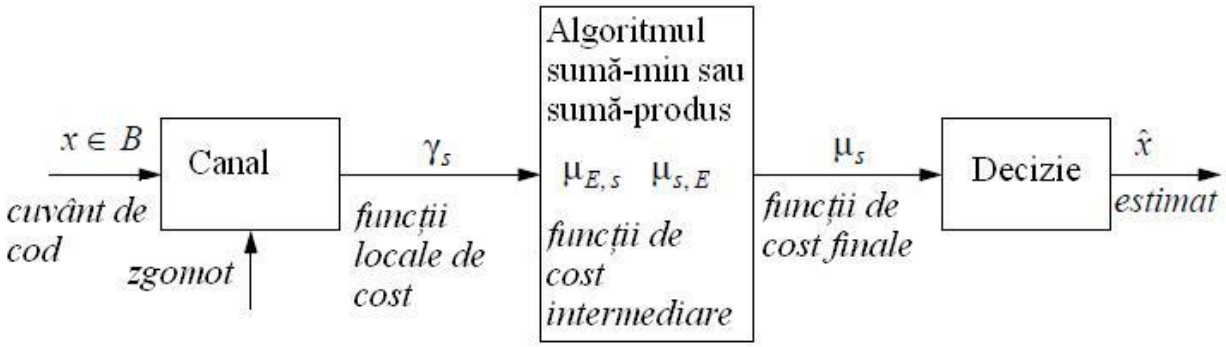
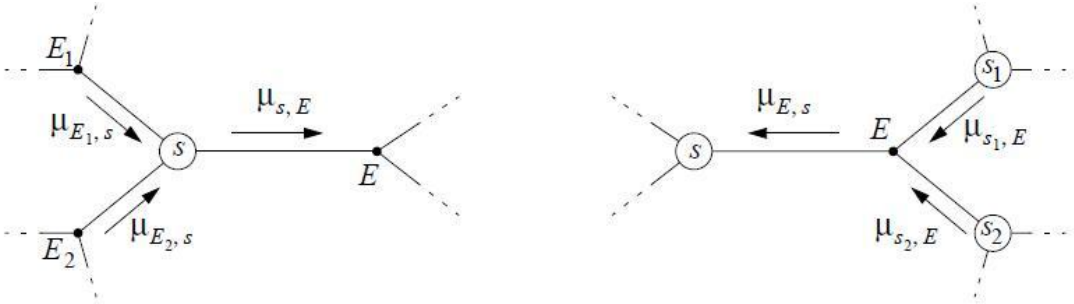


Fig. 5.1. Aplicarea tipică a algoritmilor de decodare sumă-minimă şi sumă-produs. Ieşirea canalului ia forma funcţiilor locale de cost γ_s *care sunt folosite de algoritmi pentru a calcula funcţiile finale de cost,* μ_s*, pe baza cărora vor fi luate deciziile finale. În timpul procesului de calculare, algoritmii menţin un set de funcţii de cost intermediare.*

În timpul calculelor, algoritmii menţin un set de funcţii de cost intermediare: pentru fiecare pereche (s, E) de variabile si controale adiacente(s ϵ E), există o funcţie de cost control-spre-variabilă $\mu_{E,s} : A_s \to \mathscr{R}$ şi o funcţie de cost variabilă-spre-control $\mu_{s,E} : A_s \to \mathscr{R}$. Cel mai bine este ca aceste funcţii să aibă o direcţie in graful Tanner. De exemplu, vom numi $\mu_{E,s}$ "contribuţia" dinspre controlul E spre variabila s.

5.2 Algoritmul sumă-minimă

Algoritmul sumă-minimă este o generalizare directă a algoritmului Viterbi. Scopul acestui algoritm este de a găsi o configuraţie validă $x \in B$ astfel încât suma costurilor locale(pentru toate controalele si variabilele) este minimul posibil. Când se foloseşte algoritmul sumă-minimă pentru decodarea intr-un canal fără memorie şi un un vector primit y, costurile locale ale controalelor $\gamma_E(x_E)$ sunt omise de obicei(setate la zero) iar costurile locale ale variabilelor $\gamma_s(x_s)$ sunt de obicei $-\log p(y_s|x_s)$. Într-un canal binar simetric, costul local al variabilei $\gamma_s(x_s)$ va fi reprezentat de distanţa Hamming dintre x_s si simbolul recepţionat y_s.



$$\mu_{s,E}(a) := \gamma_s(a) + \mu_{E_1,s}(a) + \mu_{E_2,s}(a) \qquad \mu_{E,s}(a) := \min_{\substack{x_E \in B_E \\ x_s = a}} [\gamma_E(x_E) + \mu_{s_1,E}(x_{s_1}) + \mu_{s_2,E}(x_{s_2})]$$

Fig. 5.2 Regulile de actualizare pentru algoritmul sumă-minimă

Algoritmul constă în următorii paşi:

- *Iniţializare.* Funcţiile de cost locale γ_s şi γ_E sunt iniţializate corespunzător(folosind informaţiile de canal). Funcţiile intermediare $\mu_{E,s}$ şi $\mu_{s,E}$ sunt setate la zero.
- *Iteraţii.* Funcţiile intermediare de cost $\mu_{E,s}$ şi $\mu_{s,E}$ sunt actualizate alternativ de un numar potrivit de ori, în funcţie de regulile prezentate in figura de mai sus. Costul variabilă-spre-control $\mu_{s,E}(a)$ este calculat ca suma ccosturilor lecale ale variabile si toate contribuţiile care intră in s, cu excepţia celei dinspre E:

$$\mu_{s,E}(a) := \gamma_s(a) + \sum_{\substack{E' \in Q: \\ s \in E', E' \neq E}} \mu_{E',s}(a) \tag{39}$$

Costul control-spre-variabilă $\mu_{E,s}(a)$ este obţinut examinând toate configuraţiile valide ale lui E care indeplinesc a pentru variabila s, pentru fiecare adunând costurile locale ale controalelor si toate contribuţiile care intră in E mai putin cea dinspre s. Minimul acestor sume va lua valoarea $\mu_{E,s}(a)$:

$$\mu_{E,s}(a) := \min_{x_E \in B_E : x_s = a} \left[\gamma_E(x_E) + \sum_{s' \in E : s' \neq s} \mu_{s',E}(x_{s'}) \right] \tag{40}$$

- *Finalizare.* Funcţiile finale de cost μ_s şi μ_E sunt calculate. Costul final al variabilelor $\mu_s(a)$ este calculat ca suma dintre costurile locale ale variabile şi toate contribuţiile care intră în variabila *s*:

$$\mu_s(a) := \gamma_s(a) + \sum_{E' \in Q : s \in E'} \mu_{E',s}(a), \tag{41}$$

iar costul final al variabilelor $\mu_E(a)$ pentru o configuraţie $a \in B_E$ este calculat ca suma dintre costurile locale ale controalelor si toate contribuţiile care intră în *E*:

$$\mu_E(a) := \gamma_E(a) + \sum_{s' \in E} \mu_{s',E}(a_{s'}). \tag{42}$$

După cum am menţionat, scopul acestui algoritm este de a găsi o configuraţie cu cel mai mic cost posibil. Pentru a formula mai precis, definim *costul global* a unei configuraţii valide $x \epsilon B$ astfel:

$$G(x) \triangleq \sum_{E \in Q} \gamma_E(x_E) + \sum_{s \in N} \gamma_s(x_s). \tag{43}$$

Teoremă. Dacă structura codului este finită si fără cicluri, funcţiile costurilor converg dupa un număr finit de iteraţii, iar funcţiile finale de cost devin:

$$\mu_s(a) = \min_{x \in B : x_s = a} G(x) \tag{44}$$

şi

$$\mu_E(a) = \min_{x \in B : x_E = a} G(x). \tag{45}$$

Pentru grafuri Tanner care conţin cicluri, nu există un rezultat general pentru costurile finale, sau pentru performanţa decodării.

5.3 Algoritmul sumă-produs

Algoritmii de decodare cu mai multe nivele de decizie se folosesc de informaţia recepţionată. Luând în considerare canalul folosit, un vector de valori a priori este generat de y şi îndeplineşte funcţia de transmisie pentru decodor. Un vector de valori a posteriori este returnat atunci când se încheie procesul de decodificare. x se produce în urma impunerii unei admiteri dure asupra acestui vector.

Algoritmul sumă-produs este o generalizare directă a algoritmului înainte-înapoi al lui Bahl[2] pentru calcularea probabilităţiilor a posteriori într-un trellis. Două cazuri speciale ale algoritmului sumă-produs sunt decodorul turbo clasic al lui Berrou[1] şi algoritmii de decodare ai lui Gallager pentru coduri cu densitate mică şi control al parităţii. Cazul general a fost descris de către Tanner, care nu a luat în considerare, totuşi, probabilităţiile a priori.

În algoritmul sumă-produs, funcţiile locale de cost γ_s şi γ_E iau o interpretare multiplicativă: definim un „costul" global pentru orice configuraţie $x \epsilon W$ ca produsul

$$G(x) \triangleq \prod_{E \in Q} \gamma_E(x_E) \prod_{s \in N} \gamma_s(x_s). \tag{46}$$

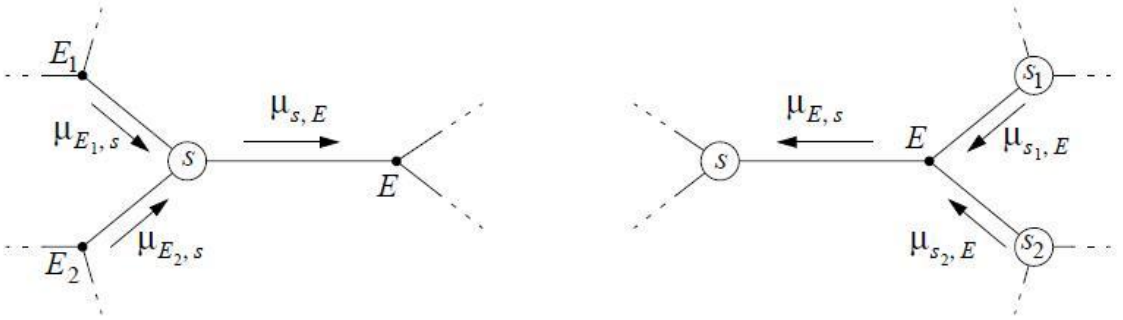
Termenul de "cost" este oarecum derutant în algoritmul sumă-produs, deoarece în general va fi subiectul unor maximizări(spre deosebire de minimizăriile din algoritmul sumă-min); am ales acest termen pentru a face transparentă strânsa relaţie dintre cei doi algoritmi. Algoritmul nu maximizează G direct; el doar calculează anumite „proiecţii" ale lui G, care vor fi la rândul lor candidaţi naturali pentru maximizare.

Când discutăm despre algoritmul sumă-produs, este normal să punem condiţia ca costurile controalelor $\gamma_E(x_E)$ să fie zero pentru configuraţiile locale invalide şi pozitive in rest. Astfel, se cere

$$\gamma_E(x_E) \geq 0 \text{ cu egalitate dacă şi numai dacă } x_E \notin B_E. \tag{47}$$

Dacă se impune de asemenea ca şi costurile locale ale variabilelor $\gamma_s(x_s)$ să fie strict pozitive pentru orice x_s, se poate observa că costul global(8) al unei configuraţii x este strict pozitiv dacă x este valid, si zero altfel. În particular, o configuraţie care maximizează G este întotdeauna validă(cu condiţia că B nu este vidă).

În situaţia în care avem un canal fara memorie şi un vector primit y, costurile locale ale variabilelor $\gamma_s(x_s)$ sunt setate ca probabilităţiile canalului $p(y_s|x_s)$, iar costurile locale ale controalelor sunt alese in funcţie de distribuţia a priori pentru configuraţia transmisă x, care trebuie să fie de forma $p(x) = \prod_{E \in Q} \gamma_E(x_E)$.



$$\mu_{s,E}(a) := \gamma_s(a)\,\mu_{E_1,s}(a)\,\mu_{E_2,s}(a) \qquad \mu_{E,s}(a) := \sum_{\substack{x_E \in W_E \\ x_s = a}} \gamma_E(x_E)\,\mu_{s_1,E}(x_{s_1})\,\mu_{s_2,E}(x_{s_2})$$

Fig.5.3. Regulile de actualizare pentru algoritmul sumă-produs.

Algoritmul sumă-produs constă în următorii paşi:

- *Iniţializare.* Funcţiile locale de cost γ_s şi γ_E sunt iniţializate corespunzător(folosind informaţiile din canal şi/sau o distribuţie a priori cunoscută). Funcţiile intermediare de cost $\mu_{E,s}$ şi $\mu_{s,E}$ sunt setate unu.
- *Iteraţii.* Funcţiile intermediare de cost $\mu_{E,s}$ şi $\mu_{s,E}$ sunt actualizate de un număr corespunzător de ori(Fig. 6.3). Costul variabilă-spre-control $\mu_{s,E}$(a) este calculat ca produsul dintre costul local al variabilei si toate contribuţiile care intră in *s* în afară de cea dinspre *E*:

$$\mu_{s,E}(a) := \gamma_s(a) \prod_{\substack{E' \in Q: \\ s \in E',\, E' \neq E}} \mu_{E',s}(a) \tag{48}$$

Costul control-spre-variabilă $\mu_{E,s}(a)$ este obtinând prin însumarea tuturor configuraţiilor locale ale lui *E* care îndeplinesc *a* pentru variabila *s*, fiecare termen fiind produsul dintre costurile locale ale variabilelor si toate contribuţiile care intră în *E* cu excepţia celei dinspre *s*:

$$\mu_{E,s}(a) := \sum_{x_E \in W_E : x_s = a} \gamma_E(x_E) \prod_{s' \in E : s' \neq s} \mu_{s',E}(x_{s'}). \tag{49}$$

Se observă că suma din (49) este doar peste B_E(configuraţiile locale valide), deoarece $\gamma_E(x_E)$ este presupus ca fiind zero pentru $x_E \notin B_E$.

- *Finalizare.* Funcţiile finale de cost μ_s şi μ_E sunt calculate. Costul final al variabilelor $\mu_s(a)$ este calculat ca fiind produsul dintre costul local al variabilei si toate contribuţiile care intră in *s*:

$$\mu_s(a) := \gamma_s(a) \prod_{E' \in Q : s \in E'} \mu_{E',s}(a) \tag{50}$$

Iar costul final al controalelor $\mu_E(a)$ va fi calculat ca produsul dintre costul local al controlului şi toate contribuţiile care intră în *E:*

$$\mu_E(a) := \gamma_E(a) \prod_{s' \in E} \mu_{s',E}(a_{s'}). \tag{51}$$

Teorema 2. Dacă structura codului este finită si fără cicluri, funcţiile de cost converg după un număr finit de iteraţii şi funcţiile finale de cost devin

$$\mu_s(a) = \sum_{x \in B : x_s = a} G(x) \tag{52}$$

şi

$$\mu_E(a) = \sum_{x \in B : x_E = a} G(x). \tag{53}$$

Ca şi în cazul algoritmului sumă-minimă, nu există nici o garanţie a performaţei de decodare dacă graful Tanner conţine cicluri.

Într-o aplicaţie de decodare(estimare), un estimat pentru valoarea variabilei x_s este obţinut prin alegerea valorii x_s care maximizează costul final $\mu_s(x_s)$. Pentru o structură ce nu conţine cicluri, această operaţiune minimizează probabilitatea de eroare de simbol.

Diagrama de coeficient reprezintă factorizarea unei funcţii globale care confirmă restricţia privind încadrarea într-o mulţime de coduri. Fiecare nod de control confirmă o restricţie locală adjunctă (XOR), susţinând ideea că paritatea per total a mesajelor care intră în nod ar trebui să fie zero. Fiecare variabilă confirmă o restricţie locală a egalităţii care se aplică mesajelor ce intră în nod, susţinând ideea că ele toate ar trebui să indice aceeaşi valoare pentru variabilă. Aceste restricţii dictează calculele care se fac în noduri, şi deci dictează şi mesajele care rezultă la ieşirea din nod. Având în vedere că mesajele reprezintă valori de probabilitate dură, folosim porţile logice [20,37] pentru a întări restricţiile. În plus, nodurile calculează un mesaj de output pentru fiecare muchie bidirecţională. La început, luăm un nod cu trei muchii, care descrie calculul ce generează un mesaj de ieşire pentru o singură muchie şi apoi aplicăm acest lucru unui nod bidirecţional. Calculam mesajul de ieşire, de exemplu, funcţia cu probabilitate binară, $p_z(z)=(p_z(0), p_z(1))$, foloseşte mesajele de intrare $p_x(x)$ şi $p_y(y)$, care se presupune că sunt independente.

Definiţia 1.(Poarta XOR). Poarta egal calculează masa de ieşire de probabilitate $p_z(z)$, folosind intrările $p_x(x)$ şi $p_y(y)$ după cum urmează:

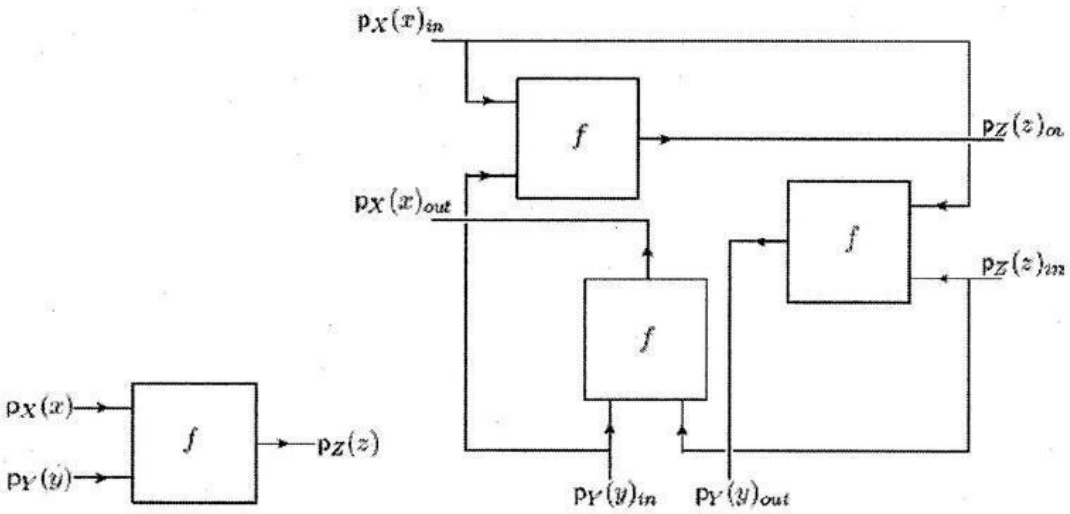
$$\begin{bmatrix} p_z(0) \\ p_z(1) \end{bmatrix} = \begin{bmatrix} p_X(0)p_Y(0) + p_X(1)p_Y(1) \\ p_X(0)p_Y(1) + p_X(1)p_Y(0) \end{bmatrix} \tag{54}$$

Definiţia 2(Poarta egal). Poarta egal calculează masa de ieşire de probabilitate $p_z(z)$ ţinând cont de următoarea expresie, unde factorul de normalizare Y este ales să asigure $p_z(0) + p_z(1)=1$.

$$\begin{bmatrix} p_z(0) \\ p_z(1) \end{bmatrix} = \upsilon \begin{bmatrix} p_X(0)p_Y(0) \\ p_X(1)p_Y(1) \end{bmatrix} \tag{55}$$

Figura 4 ne arată cum putem folosi aceste câmpuri pentru a construi câmpuri bidirecţionale, pentru care avem următoarea definiţie.

Definiţia 2.5 (Poarta Bidirecţională Logică (BSG). Poarta bidirecţională logică este un dispozitiv cu 3 porturi unde fiecare port are o ieşire şi o intrare. Ieşirea fiecărui port este calculată folosind cele două intrări cu muchie extrinsecă şi întrebuinţând aceeaşi funcţie, de exemplu, $p_x(X)_{out}=f(p_y(y)_{in}, p_z(z)_{in})$, $p_y(y)_{out}=f(p_x(x)_{in}, p_z(z)_{in})$ şi $p_z(z)_{out}=f(p_x(x)_{in}, p_y(y)_{in})$. Deci un BSG poate fi construit pornind de la trei porţi, prin alinierea ieşirii unuia la fiecare dintre muchiile de ieşire a fiecărui BSG. Datorită realizărilor lui Forney în domeniul grafurilor normale, putem să spargem structura unui nod cu mai multe muchii în elemente constituente. Variabilele şi nodurile de control de mărime arbitrară pot fi construite prin legarea în cascadă a porţilor bidirecţionale şi respectiv a porţilor XOR[20,37].



(a) poartă cu o ieşire (b) poartă cu trei porturi bi-direcţionale

Fig. 5.4 Porti logice

Odată ce nodurile variabile şi cele de control au fost construite, se leagă folosind muchii bidirecţionale, în conformitate cu structura diagramei de coeficient. Definim E(vs) ca fiind o mulţime a tuturor nodurilor de control alăturate nodului variabil vs, specificat de coloana s a lui H. Tot astfel, mulţimea tuturor nodurilor variabile alăturate nodurilor de control c_r, $E(c_r)$, este specificată de şirul r al matricei H. Mesajele de control al variabilelor sunt notate cu $\mu_{s,E}$ iar mesajele variabilelor de control cu $\mu_{E,s.}$ Prin E(vs)\c definim toate controalele E(vs) excluzând controlul c, şi tot astfel definim şi $E(c_r)$\s. O muchie de intrare cu o singură direcţie, folosită la transportarea mesajelor a priori, p_s, este adaugată la fiecare nod variabil în diagrama de coeficient. În acelaşi fel, o muchie de ieşire cu o singură direcţie este ataşată pentru a transporta mesajul a posterior, q_s.

Valoarea a posterioară, q_s, este generată pentru fiecare simbol al cuvântului de cod, la sfârşitul fiecărei iteraţii. În concluzie la aceste valori este executată o admitere dură ca sa fie posibilă generarea estimatului cuvântului de cod X. Dacă estimaţia satisface toate restricţiile codului, atunci procesul se încheie, în caz contrar el se repetă până când va expira un număr maxim de iteraţii.

Fiind date valorile recepţionate y şi diagrama de coeficient pentru H, acum rezumăm algoritmul sumă-produs cu două nivele de decizie. Sunt furnizate atât descrierea domeniului de probabilitate cât şi descrierea domeniului de înregistrare a probabilităţii, presupunând un canal AWGN şi o transmisie aplicată BPSK, M(0)=+1, M(1)=-1. Presupunem de asemeni că deţinem cunoştinţe despre variaţia canalului de zgomot, σ^2. Variabila de control a domeniului de probabilitate şi regulile revizuite ale controluiui variabilei revizuită din descrierile porţilor XOR şi egal de mai sus. Regulile domeniului de înregistrare a probabilităţii sunt uşor derivate, folosind definiţia LLR.

Când executăm algoritmi cu două nivele de decizie este important să luăm în calcul potenţiale chestiuni numerice. La Pasul 3 al algoritmului de domeniu de înregistrare a probabilităţii, trebuie să rezolvăm problema discontinuităţii $\tanh^{-1}(a)$, când a = 0. Pentru a face asta definim parametrul limitor η

.

5.4 Decodarea cu două nivele de decizie (Hard Decision Decoding)

Putem aplica decodarea cu două nivele de decizie în cazul AWGN, ştergerii binare şi canalelor binare simetrice[38]. Dacă informaţia este purtată cu vectorul de recepţionare atunci este descărcată. Aceşti algoritmi nu au acelaşi nivel de performanţă ca algoritmii de decizie cu două nivele. Totuşi, simplitatea lor permite executarea lor rapidă şi a dus de asemenea la câteva rezultate analitice interesante[5, 39, 40, 41].

În cadrul acestei discuţii considerăm y pentru a reprezenta vectorul de recepţionare, care provine din M(ys<0)=+1, M(ys>=0)=-1. La început Gallager a propus următorii doi algoritmi de admitere dură.

Algoritmul A de Decodare Gallager

Mesajul de ieşire din nodul variabil v_s de-a lungul muchiei *e* este egal cu valoarea recepţionată y_s deoarece acel nod, altul decât cel de-a lungul muchiei *e* este diferit de v_s, dacă toate mesajele intră în y_s. În acest caz opusul lui y_s este trimis, de exemplu, dacă y_s = -1 atunci o valoare de mesaj de +1 este trimisă, şi invers.

Mesajul cu ieşire din controlul c_r către o variabilă conectată de-a lungul muchiei *e* este rezultatul tuturor mesajelor primite pe muchii conectate la c_r, excluzând cea de la *e*.

Algoritmul 1: Decodorul Sumă-Produs (Domeniul de Probabilitate)

1. Folosind fiecare simbol recepţionat y_s se calculează $u=2y_s/\sigma^2$.
   Se calculează $p_s=e^u/(1+e^u)$ si $p_s=e^{-u}/(1+e^{-u})$
2. Variabila spre control:

   Presupunem $\delta_{vs}= \mu_{s,E}(0) - \mu_{s,E}(1)$ *şi calculăm*

$$\delta_{Cr\to v} = \prod_{vs\in\Gamma(cr)\setminus v} \delta_{vs}$$

   De la fiecare control cr la fiecare variabilă s ϵ E(vs) se transmite:

$$\mu_{cr\to u}(0) = 1/2(1+\delta_{cr\to u}) \text{ si } \mu_{cr\to u}(1) = 1/2(1-\delta_{cr\to u})$$

3. Control spre variabilă:
   Din fiecare variabilă *s* pentru fiecare *c ϵ E(vs)* se transmite

$$\mu_{v_S\to c}(0) = \gamma p_S(0) \prod_{c_r\in\Gamma(v_S)\setminus c} \mu_{c_r\to v_s}(0)$$

şi

$$\mu_{v_S\to c}(1) = \gamma p_S(1) \prod_{c_r\in\Gamma(v_S)\setminus c} \mu_{c_r\to v_s}(1)$$

   Factorul de normalizare γ este ales astfel încât $\mu_{s,E}(0) + \mu_{s,E}(1) = 1$.
4. Calcule a posteriori

   Pentru fiecare simbol se calculează :

$$q_s(0) = \gamma p_s(0) \prod_{c_r \in \Gamma(v_s)} \mu_{c_r \to v_s}(0)$$

şi

$$q_s(1) = \gamma p_s(1) \prod_{c_r \in \Gamma(v_s)} \mu_{c_r \to v_s}(1)$$

Factorul de normalizare γ este ales astfel încât $q_s(0) + q_s(1) = 1$.

5. Stop/Continuă:
   Se calculează din x din:
   $$\hat{x}_s = q_s(1) > 1 \setminus 2$$
   Dacă H xˆ =0, apoi operaţiunea este încheiată cu succes, astfel dacă se atinge limita iterativă operaţiunea trebuie încheiată şi declarată un eşec, în caz contrar se face întoarcerea la Pasul 2.

Algoritmul 2: Decodorul Sumă-Produs(Domeniul de Înregistrare a Probabilităţii)

1. Iniţializarea:

   Se initializeaza mesajele $\mu_{s,E} = \mu_{E,s} = 0$

2. Setarea priorităţii:

   Folosind fiecare simbol y_s se stabileşte $p_s = 2y_s/\sigma^2$

3. Control spre variabilă:

   De la fiecare control c_r la fiecare variabilă $s \in E(c_r)$ se trimite

   $$\mu_{c_r \to v} = 2\tanh^{-1}\left(\prod_{v_s \in \Gamma(c_r) \setminus v} \tanh(1 \setminus 2\mu_{v_s \to c_r})\right)$$

4. Variabilă spre control:
   De la fiecare variabilă s la fiecare $c \in E(vs)$ se trimite
   $$\mu_{v_s \to c} = p_s + \sum_{c_r \in \Gamma(v_s) \setminus c} \mu_{c_r} \to v_s$$
5. Calcule a Posteriori:

   Pentru fiecare simbol se calculează

   $$q_s = p_s + \sum_{c_r \in \Gamma(v_s)} \mu_{c_r \to v_s}$$

6. Stop/Continuă:

   Dacă toate controalele respectă

$$\prod_{v_s \in \Gamma(c_r)} q_s > 0$$

   atunci operaţiunea se încheie fiind declarată un succes, altfel dacă este atinsă limita de iteraţie atunci operaţiunea se încheie fiind declarată un eşec, în caz contrar trebuie să ne întoarcem la Pasul 3.

Algoritmul B de Decodare Gallager

Acest algoritm este o versiune modificată a algoritmului A. Când se calculează mesajul variabil, opusul valorii recepţionate va fi trimis dacă cel puţin un număr b de mesaje în curs de expediere sunt în dezacord. Această valoare a mesajelor b este în mod tipic o funcţie a gradelor muchiei şi a numărului de iteraţie.

Decodare cu eliminare

Să luăm în considerare canalul binar de ştergere, cu aplicaţia ieşirii M(0) = +1, M(1) = -1, M(stergere) = 0. Definim operatorul sgn(a) = +-1 pentru a <> 0 şi sgn(0) = 0. Decodorul cu eliminare a mesajelor în curs acţionează, folosind aritmetica reală, după cum urmează. Putem considera acest algoritm ca fiind o versiune de admitere cu două nivele de decizie a decodorului sumă-produs.

<u>Algoritmul 3: Decodorul cu eliminare</u>

1. lniţializarea:
   Se fixează valoarea variabilei s în funcţie de valoarea corespunzătoare simbolului primit $y_s \in \{0,-1,+1\}$. Se iniţializează toate mesajele cu 0.
2. Variabilă spre control:
   De la fiecare variabilă s la fiecare $c_r \in E(v_s)$ se trimite

$$\mu_{Vs \to c} = \operatorname{sgn}\left(v_s + \sum_{C_r \in \Gamma(v_s) \setminus c} \mu_{c_r} \to v_s \right)$$

3. Control spre variabilă:
   De la fiecare punct de control c_r la fiecare variabilă $s \in E(c_r)$ se trimite

$$\mu_{Cr \to v} = \prod_{V_s \in \Gamma(Cr) \setminus V} \mu_{Vs \to Cr}$$

   Pentru toate variabilele s, dacă cel puţin o $\mu_{s,E} \neq 0$, atunci se dă valoarea variabilei s.
4. Stop/Continuare:
   Dacă $s \neq 0$ pentru toate variabilele s atunci operaţiunea e încheiată şi declarată un succes. Dacă aceasta nu e prima iteraţie şi mulţimea de valori pe care o au în prezent variabilele s

este aceeaşi cu mulţimea de valori a ultimei iteraţii, atunci operaţia se încheie şi e declarată un succes. În caz contrar trebuie să ne întoarcem la Pasul 2.

Capitolul VI. Aplicaţii ale codurilor LDPC

Codurile cu densitate mică şi control a parităţii posedă o serie de caracteristici care le fac deosebit de atractive pentru cercetători, modul lor de construcţie permiţând o implementare fizică facilă, şi atingerea unor performanţe mult îmbunătăţite faţă de alte coduri clasice, la costuri similar, într-o gamă variată de domenii.

În ultima decadă codurile LDPC au primit multă atenţie datorită performanţelor superioare de corecţie de erori, şi au fost adoptate de mai multe standarde recente de comunicaţie, cum ar fi 10 Gigabit Ethernet(10GBASE-T), broadcast video digital(DVB-S2(satelit), DVB-T2(terestru), DVB-C2(cablu)), Wimax(802.16e), Wi-Fi(802.11n) şi 60 GHz WPAN(802.15.3c)[1].

6.1 Codurile LDPC in comunicaţii optice

Comunicaţiile optice reprezintă un domeniu care începe să folosească coduri cu densitate mică şi control al parităţii pentru rezolvarea unor probleme importante.

Performanţele sistemelor de comunicaţii prin fibră optică ce operează la viteze mari sunt degradate semnificativ datorită mai multor imperfecţiuni de transmisie, cum ar fi neliniarităţi ale fibrei la nivel intra sau inter-canal, efectul Gordon-Mollenauer sau dispersia în polarizare(PMD). PMD este un defect dificil de compensat datorită caracteristicilor sale stocastice şi variabile în timp. În ultima vreme, mai multe compensatoare electrice de PMD au fost propuse, care folosesc egalizatoare Viterbi sau turbo, dar s-au obţinut rezultate foarte bune şi folosind multiplexarea ortogonală prin divizie in frecvenţă(OFDM) a unor semnale codate LDPC[42]. OFDM este un caz special de transmisie multi-purtătoare în care un singur sir de informaţii este transmis pe mai multe sub-canale de rată mai mică, şi este deja folosit pentru o multitudine de aplicaţii de ultimă generaţie, printre care transmisia HDTV, transmisia audio digitală, IEEE 802.11, comunicaţii optice prin spaţiu liber, legături de fibră multi-mode sau 100Gb/s Ethernet.

Datorită ortogonalităţii dintre sub-purtătoarele OFDM este permisă suprapunerea parţială a sloturilor de frecvenţe vecine, astfel îmbunătăţindu-se eficienţa spectrală comparativ cu alte sisteme convenţionale multi-purtătoare. De asemenea, folosind un număr suficient de mare de de sub-purtătoare, interferenţa inter-simbol(ISI) datorită PMD poate fi redusă.

Această schemă de compensare a PMD este combinată cu o corecţie de erori avansată care se bazează pe coduri LDPC structurate.

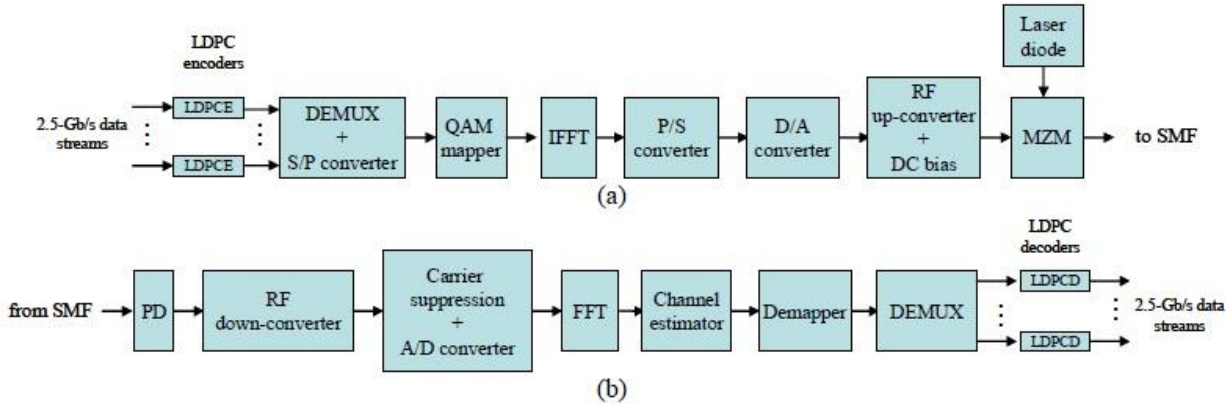


Fig.6.1 Configuraţia transmiţătorului(a), configuraţia receptorului(b)

Pentru implementarea acestui sistem sunt propuse două familii de coduri LDPC de rată mare, bazate pe două obiecte combinatoriale diferite: (i) sisteme de diferenţe, si (ii) un produs de designuri combinatoriale.

Un set $S_1....S_t$ de dimensiune k formează un sistem de diferenţe (v, k, λ) peste un grup abelian aditiv V de ordinul v, dacă diferenţele faţă de S_i ale fiecărui element nonzero din V sunt maxim λ.

Deplasând ciclic fiecare set S_i de m ori(mod v), b=tm seturi diferite pot fi create. Considerând elementele a b seturi ca poziţii de 1 în coloanele corespondente ale unei matrici de incidenţă, se poate verifica faptul că o astfel de matrice de incidenţă are proprietăţiile unei matrici de control a parităţii H a codului LDPC corespondent. Alegând indicele λ=1, se poate demonstra ca matricea H rezultată are o circumferinţă de minim 6.

Pentru aceste sisteme sunt folosite următorii parametrii: v=20t+1, k=5, λ=1. Numărul de blocuri din aceste sisteme de diferenţe este b=t(20t+1). Codul LDPC corespondent are lungimea N= t(20t+1), numărul de biţi de paritate N-K= 20t+1, rata de cod este limitată inferior de R≥1-1/t, iar circuferinţa este minim 6.

Pentru rate de cod peste 0.9, parametrul t trebuie să fie mai mare sau egal cu 10. Deoarece standardul ITU-T G.975 pentru comunicaţii prin fibră optică limitează overhead-ul pentru corecţie a erorilor la 7%, designul codurilor LDPC cu rată peste 0.9 este foarte important. În acelasi timp, întârzierile în decodare nu pot fi acceptate din cauza vitezelor extrem de mari, fiind necesare coduri LDPC de lungime medie. De exemplu, pentru t=14 se obţine un cod LDPC de rată R=1-1/14=0.93, cu 7.7% overhead. Lungimea cuvântului de cod va fi 3934 iar numărul de biţi de paritate N-K=281.

Aceste coduri au dus la rezultate foarte bune în compensarea PMD pentru grupuri diferenţiale de întârziere(DGD) mai mari de două perioade de bit. Pentru DGD-uri sub două perioade de bit este mai eficient de folosit egalizarea turbo a semnalului codat LDPC. Aceste scheme de compensare au adus importante avantaje OFDM, cum ar fi insensitivitatea la

zgomotul de fază al laserului, o complexitate mai mică a implementării receptorului şi compatibilitatea cu sistemele IM/DD instalate anterior.

6.2 Codurile LDPC in comunicaţiile în spaţiul cosmic

Unul dintre cele mai interesante şi problematice domenii în care codurile LDPC şi-au arătat eficienţa în ultima decadă este cel al comunicaţiilor în spaţiul cosmic. Comparativ cu comunicaţiile standard terestre şi satelitare, comunicaţiile în spaţiul cosmic prezintă o serie de probleme pentru transmisia de informaţii, cum ar fi distanţele foarte lungi care trebuie parcurse de semnal(prin spaţiu cosmic se înţelege de obicei spaţiul aflat la mai mult de 2 milioane de km de Pământ), raport semnal-zgomot foarte mic, întârzieri mari în propagarea semnalelor, rate de corupţie a datelor ridicate sau lărgimi de bandă asimetrice. În studierea spaţiului, pe lângă tehnologia extraordinar de avansată folosită la emiţător, receptor sau antene, un rol foarte important îl au şi codarea la sursa imaginii, codarea de canal, modulaţia sau demodulaţia.[43]

Comunicaţiile în spaţiu au continuat să se îmbunătăţească odată cu dezvoltarea explorărilor spaţiale, în mai mult de 40 de ani. Mariner4, lansat în 1965, comunica folosind banda S(2.3 GHz), şi nu existau nici coduri corectoare de erori, nici compresia datelor, iar rata era de doar 8.33 bps. Mars Global Survezor(MGS), lansat in 1997, folosea banda X(8.4 GHz). Codul de canal folosea un cod convolutional cu constrângere de lungime 7 şi rată 1/2 concatenat cu un cod Reed-Solomon(255, 223), iar rata de transmisie era 128 kbps. Mars Pathfinder, lansat în 1997, folosea codarea JPEG cu un raport de compresie de 6. Misiunile de explorare Spirit şi Opportunity(MER-A şi MER-B) trimise pe Marte in 2004 foloseau un cod de compresie a imaginilor bazat pe wavelet. Raportul de compresie era 12, iar rata de transmisie era 168kbps. Pentru misiunea Mars Reconnaissance(MRO), lansată in 2006, s-au folosit coduri Turbo şi LDPC ca şi coduri de canal, şi un sistem de compresie a imaginiilor rapid şi eficient ca şi cod sursă, iar viteza de transmisie este 12Mbps. Sistemele de comunicaţie folosite la MER şi MRO reprezintă cele mai avansate tehnologii din lume în acel moment.

Comparativ cu codurile Turbo, codurile LDPC au prezentat performanţe similare(chiar superioare în cazul codurilor lungi) la decodare, dar au şi avantajul unei viteze superioare de decodare, implementarea VLSI mult simplificată şi un prag scăzut de erori. Aceste coduri sunt potrivite pentru condiţii de comunicaţii în spaţiul cosmic, în care se întâlnesc întârzieri mari. Cercetările au arătat că un cod LDPC neregulat cu rată 1/2 şi mesaje de lungime 10^7 biţi atinge o performanţă cu doar 0.04dB sub limita Shannon într-un canal AWGN, aceste performanţe fiind superioare chiar şi celor mai bune coduri Turbo.

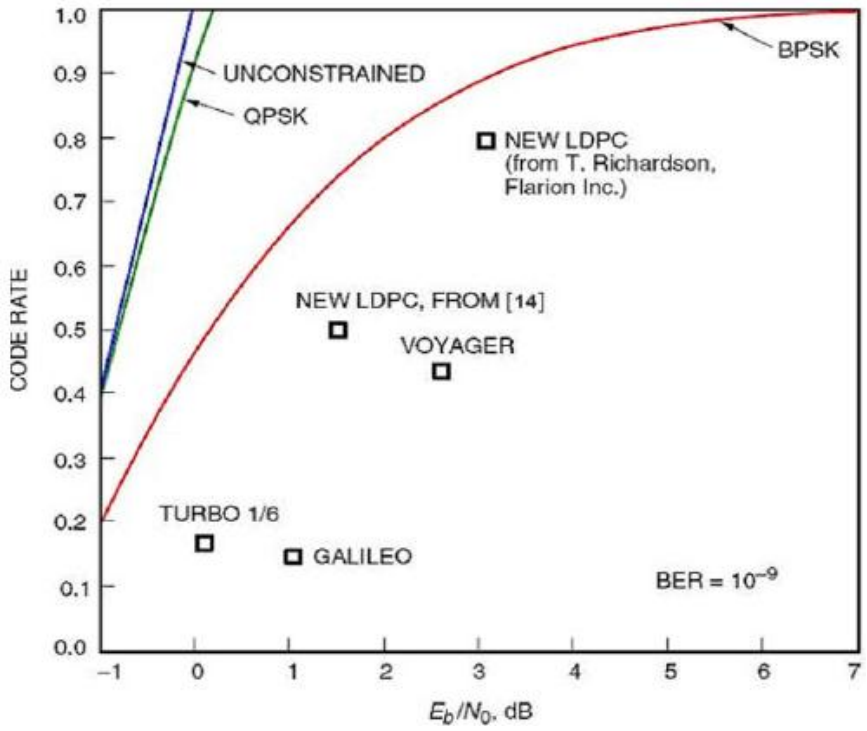


Fig. 6.2 Performanţele codurilor pentru BER=10^{-9}

Codurile cu densitate mică şi control a parităţii folosite sunt coduri de rată 0.5 sau 0.8, după cum se observă în figură. Ratele sub 0.5 sunt potrivite pentru misiuni în spaţiu la care se folosesc rate mici, iar cele între 0.5 şi 0.8 sunt potrivite pentru misiuni în spaţiu cu rate mari, dacă sunt modulate QPSK. Rate de cod peste 0.8 nu sunt recomandate comunicaţiilor spaţiale datorită reducerii severe a eficienţei puterii. Aceste coduri au două mari avantaje: o mai mică complexitate la decodare, fiind deci recomandate pentru pentru viteze foarte mari(>10 Mb/s), şi se comportă mult mai bine la BER foarte mic(10^{-9}), deoarece pragul lor de erori poate fi controlat şi pot fi forţate la BER mai mic. Pentru operaţiuni care necesită valori ale BER mai mici de valoarea tradiţională de 10^{-6} şi rate mari de date, codurile LDPC sunt alegerea potrivită.

6.3 Codurile LDPC în sisteme de comunicaţii RF

În cazul comunicaţiilor RF/microundă, semnalul care se propagă prin atmosferă va experimenta fluctuaţii în amplitudine şi în fază datorita turbulenţelor atmosferice(ceaţă, ploaie). Pentru a se combate măcar o parte din aceste fluctuaţii si a se îmbunătăţii performanţele sistemelor de transmisiune, au fost propuse mai multe tehnici de codare, printre care şi folosirea de coduri convoluţionale, coduri Turbo sau coduri Reed-Solomon dar toate aceste tehnici prezintă o serie de neajunsuri. În cazul codurilor convoluţionale non-binare sau a codurilor

Turbo, complexitatea foarte ridicată la decodare poate fi prohibitivă pentru implementarea unui sistem de transmisie prin care se doreşte atingerea unor rate de cod de ordinul Gbps, iar pentru codurile Reed-Solomon, în condiţii de turbulenţe puternice sau de ceaţă densă, câştigul din codare este prea mic pentru a face faţă cerinţelor şi sunt necesare scheme de codare FEC(Forward Error Control) mai avansate, cum ar fi coduri Turbo sau LDPC.

În ultimii ani, cererea de viteze din ce în ce mai mari de comunicaţii radio a dus la o mai mare nevoie de frecvente de transmisie. În plus, cantitatea de informaţie trimisă între dispozitive a crescut rapid datorită îmbunătăţirilor în calitatea sunetului şi a imaginiilor, fişierelor audio, foto, sau a imaginilor video folosite pentru TV, aparate mobile sau servicii de video-conferinţă online. Aceste îmbunătăţiri au dus la cerinţa ca tehnologiile să faciliteaze transmisia de mari cantităţi de date la viteze mult mai ridicate.

În ideea de a ajuta la rezolvarea acestor probleme, la începutul anului, compania japoneză Sony, în colaborare cu Tokyo Institute of Technology, a anunţat creearea unui sistem de transmisiune în radio frecvenţă format dintr-o serie de LSI-uri(circuite integrate) care permit transfer de date folosind lungimi de undă milimetrice, cu cea mai mare rată de transfer din lume, 6.3 Gb/s.

Implementarea acestei tehnologii va permite utilizatorilor să trimită şi să primească date la viteze mult mai mari fără a avea nevoie de conexiuni prin cablu. De asemenea, cu ajutorul acestei tehnologii, utilizatorii vor putea beneficia de streaming video de cea mai bună calitate dinspre un terminal mobil către un display.

La dezvoltarea acestei tehnologii, un rol important l-au avut o familie de coduri LDPC foarte eficiente, de rata 14/15, dezvoltate de Sony, care scad cantitatea de date redundante necesare corecţiei de erori, acest lucru ducând la o decodare LDPC cu cea mai mare eficienţă energie per-bit din lume: 11.8 pJ/b(74 mW pentru 6.3 Gb/s). Aceste coduri au fost propuse pentru standardul de comunicaţii wireless IEEE 802.15.3c, care foloseşte banda de frecvenţe de 60 GHz, cu lungimi de undă milimetrice.

6.4 Coduri LDPC în sisteme de comunicaţii satelitare

Un alt domeniu de interes major pentru implementarea LDPC îl reprezintă comunicaţiile satelitare, care în prezent cunosc o dezvoltare fără precedent. În momentul de faţă, acest tip de comunicaţii se realizează folosind intervalul de frecvenţe cuprinse între 26,5 – 40 GHz, denumit şi banda Ka. Totuşi, propagarea benzii Ka este mult mai vulnerabilă la condiţiile meteorologice şi la umbrire decât benzile de frecvenţe inferioare, cum ar fi banda Ku (12-18 Ghz) sau banda C(4-8 GHz). În cazul canalelor satelitare în bandă Ka, atenuarea datorată ploii şi umbrirea sunt principalii factori care deteriorează performanţele sistemului, pierderea semnalului radio,

scăderea fiabilităţii şi a performanţelor link-urilor satelitari. Pentru a combate aceste probleme, este necesară o metodă de codare eficientă pentru transmisiuni cu rată mare de date.

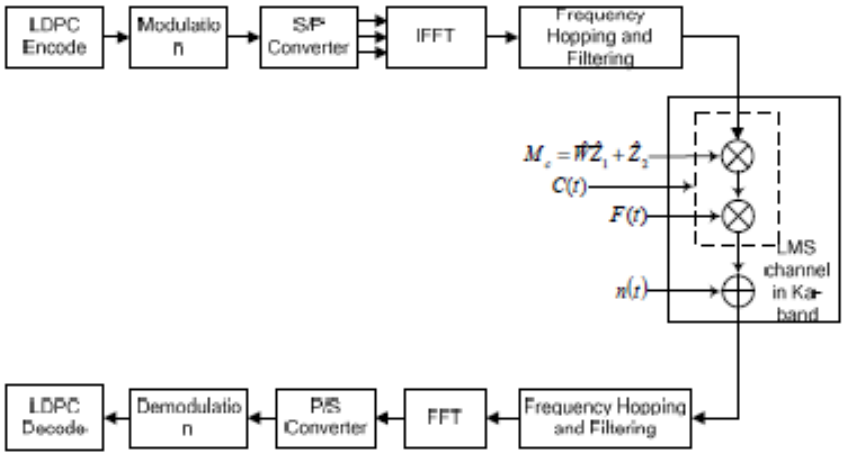


Fig. 6.3 Diagrama bloc a sistemului în bandă Ka codat LDPC

La transmiţător, datele binare sunt codate folosind codorul LDPC si apoi sunt mapate prin modulaţie. Datele complexe sunt trecute apoi printr-un convertor serial-paralel(S/P) şi apoi li se aplică transformata Fourier inversă(IFFT) pentru a fi modulate. Datele sunt împrăştiate în domeniul timp şi frecvenţă, astfel că sistemul poate profita din plin de diversitatea de frecvenţe a canalului.

La receptor, transformata Fourier este aplicată datelor primite, care sunt apoi modificate de către convertorul paralel-serial(P/S) şi demodulate. La final, datele sunt decodificate folosind decodorul LDPC.

În [45] se compară performanţele BER între codurile LDPC şi codurile convoluţionale şi se arată că, pentru modulaţii 16QAM şi BPSK, codurile LDPC oferă un câştig la codare net superior codurilor convoluţionale, în condiţiile unui BER comparativ, ceea ce le face eficiente şi fezabile pentru comunicaţiile satelitare.

6.5 Rezumat

Teoria codării a suferit o schimbare de paradigmă în ultimii ani, odată cu introducerea decodării iterative şi a codurilor în grafice. Codurile de joasă densitate şi verificare a parităţii introduse de Gallager oferă o corectare a erorilor mai bună. De la redescoperirea lor multe modificări au fost propuse pentru a le îmbunătăţi ulterior performanţele.

În orice caz, mai trebuie imbunătăţiri în domeniul analizei decodării iterative şi în proiectarea codurilor bune de lungime scurtă şi medie. Analiza grafică se bazează pe structura unei matrici de verificare a parităţii particulară şi ia în considerare operaţia de decodare iterativă. În acest caz, s-a arătat că metricitatea tradiţională a distanţei minime a codului este mai puţin importantă decât pseudo-greutatea minimă asociată matricii de verificare a parităţii.

Capitolul VII. Structura aplicaţiei de simulare

In simulatorul deja existent,CAMAG 6, am îmbunătăţit şi optimizat modulul de simulare a codurile LDPC(Low Density Parity Check) creată în mediul de dezvoltare Visual Studio 2008 cu limbajul C++.

7.1 Interfaţa cu utilizatorul. Descrierea principalelor comenzi

Aplicaţia realizată oferă o interfaţă simplă şi uşor de utilizat care permite selectarea şi configurarea rapidă a parametrilor simulării. Fereastra principală a programului, prezentată în figura 7.1 permite selectarea tipului de canal magnetic şi a detectorului pentru care se va realiza simularea. Tot aici se poate efectua pas cu pas o simulare prin apăsarea succesivă a butoanelor: **Generare si Codare** – **Detectie si Decodare**.

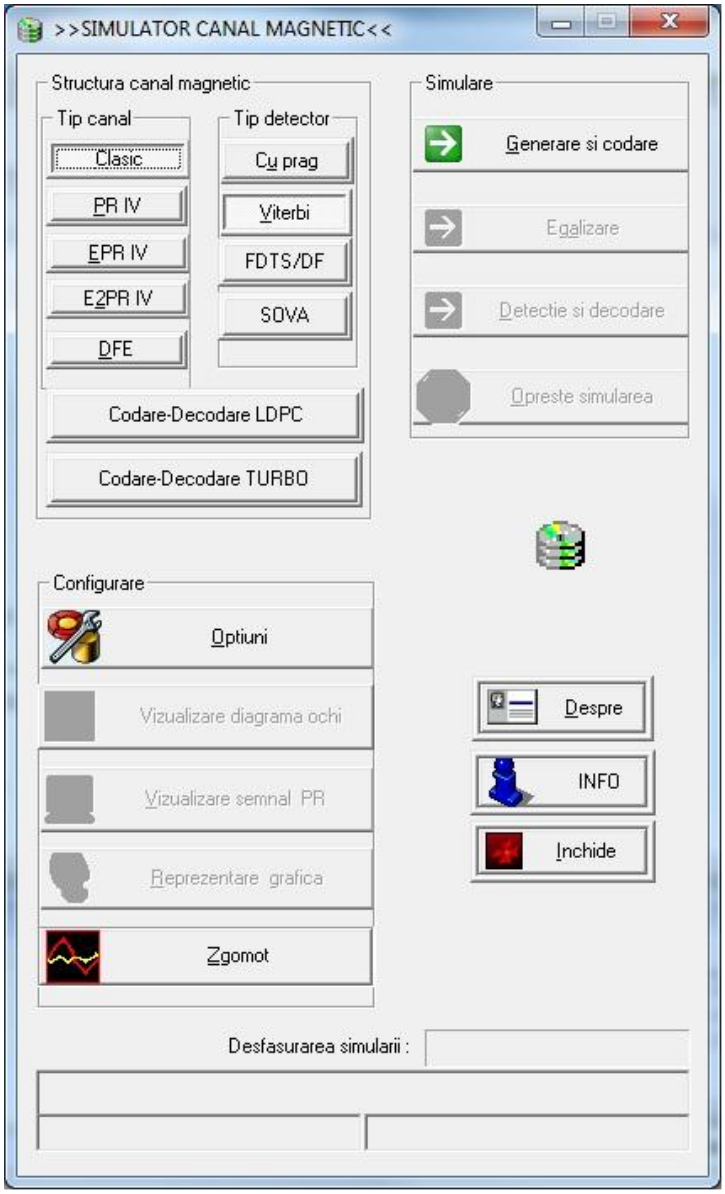


Fig. 7.1 Fereastra principală a simulatorului CAMAG7

Pentru simularea de coduri LDPC ,prin apăsarea butonului şi prin completarea câmpurilor din “Opţiuni”, se lansează programul de simulare.

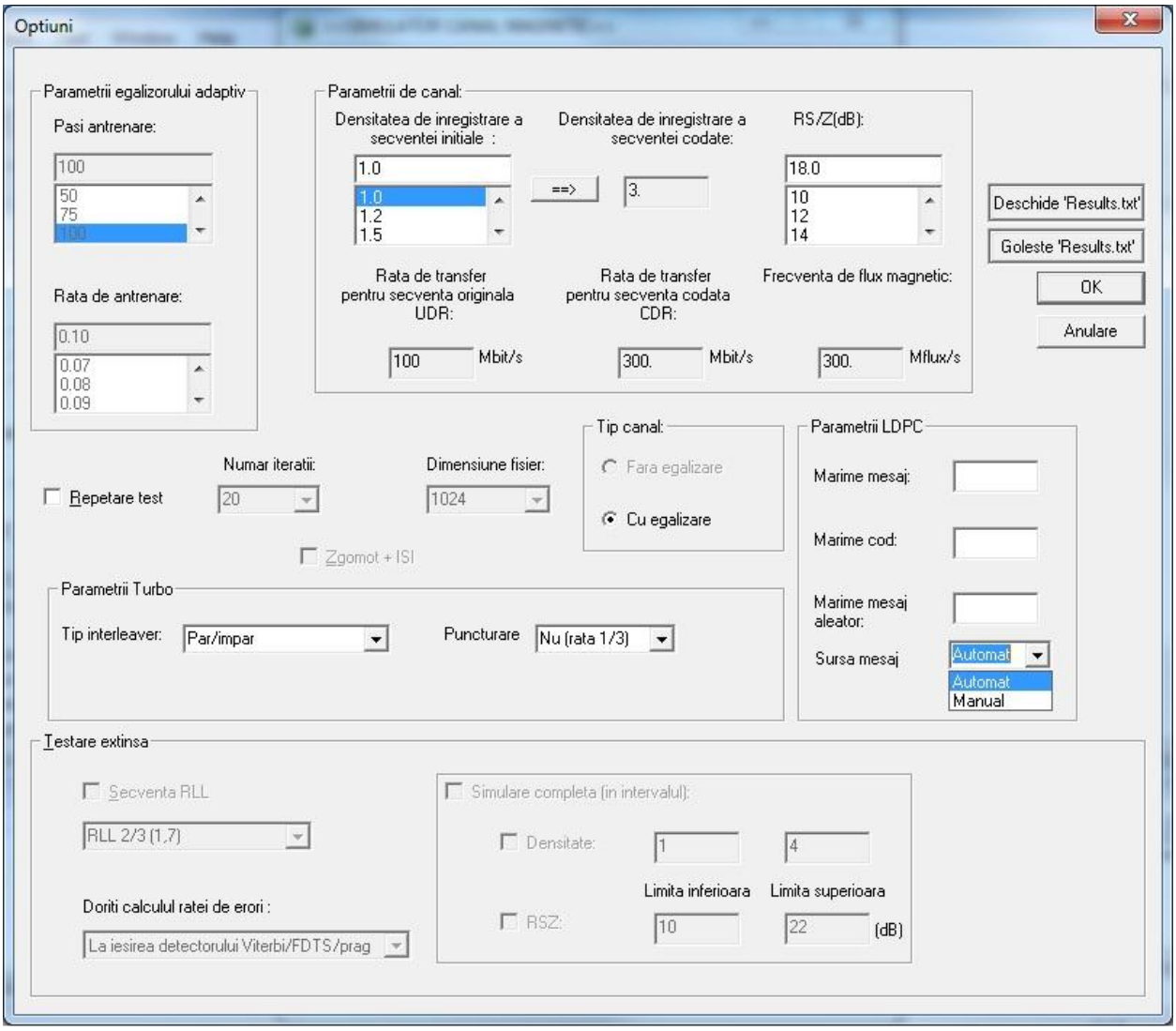


Fig. 7.2 Fereastra Opţiuni

7.2 Explicarea softului de simulare

Iniţial, programul creează matricea H (matrice semi-aleatoare de verificare a parităţii) a cărei coloane conţin acelaşi număr de 1’uri. Coloanele sunt create una după alta, într-o manieră în care nu se permite ca două coloane să aibă mai mult de o suprapunere de 1’uri în toate rândurile lor. Această metodă reduce ciclurile scurte în graful Tanner asociat matricii, dar în schimb face procesul de decodare mult mai eficient. Deasemenea, ajută la conceperea tuturor rândurilor matricii H astfel încât acestea să aibă un număr aproximativ egal de 1’uri. Sunt

necesari câţiva paşi adiţionali pentru a face ca toate rândurile să aibă acelaşi număr de 1’uri, dar au fost evitaţi aceşti paşi deoarece codurile LDPC neregulate au o performanţă mai bună.

După creare, matricea H este adusa la forma sistematica H=[P|I], iar în final, matricea generatoare G este construită sub forma: G=[I | P].

Procesul de decodare este făcut cu algoritmul de transmitere a mesajelor, decodor probabilistic.

La rularea programului, se creează un fişier text (RezultateSimulareLDPC.txt) în care se va păstra ultimul rezultat al simulării. Se cere utilizatorului să introducă dimensiunea mesajului (messageLenght) şi dimensiunea codului (codeLenght).

Programul va aloca spaţiul de memorie necesar pentru crearea matricilor H si G. Pentru matricea H trebuie îndeplinite anumite reguli si se pun următoarele condiţii:

1. să nu existe mai mult de o suprapunere de 1’uri între oricare două rânduri

2. greutatea fiecărei coloane să nu fie mai mare decât cea specificată în program, care este 3

Se vor face 5 încercări pentru generarea matricii H, dar în cazul în care nu se poate genera, se va abandona sarcina şi se va afişa pe ecran numărul de iteraţii în care s-a încercat crearea.

În cazul în care matricea H este generată, se încearcă reducerea acesteia la forma $H=[P_t|I]$. Dacă este imposibilă trecerea la o astfel de formă, se iese din program.

Având H sub aceasta formă, se compune matricea G sub forma [I | P] şi se adaugă zgomotul (AWGN) de dispersie setată manual în cod. La măsurători s-a folosit o valoare de 0.5. Pentru aflarea mesajului care va fi codat, există posibilitatea alegerii între citirea mesajului dintr-un fişier, sau generarea unui mesaj aleator (m) în funcţie de dimensiunea mesajului introdusă de utilizator (k), apoi se codează acest mesaj în funcţie de G, k si n (dimensiunea codului introdusă de utilizator). Mesajului codat C i se adaugă zgomot pentru a se coda pentru canal AWGN. Mesajul codat AWGN, Xg, se trece prin canal şi se scoate Yg. Se verifică dacă cuvântul primit are erori($Y * H^T=0$). Se decodează cuvântul primit în funcţie de H, k, n şi dispersia zgomotului,

se verifică dacă este corect şi se afisează rezultatul, iar după terminarea decodării tuturor blocurilor se afişează numărul de biţi eronaţi rezultaţi după decodare.

Cap VIII. Rezultate experimentale

In acest capitol am efectuat o simulare cu softul de simulare LDPC modificat în programul CAMAG 7. Desfăşurătorul simulării este următorul:

Mărimea mesajului: 15

Mărimea codului: 30

Dimensiunea mesajului creat aleator: 30

Incerc sa creez matricea H pentru coduri LDPC

==========

Se creeaza coloana 0.....Gata!!! A durat 1 iteratii
Se creeaza coloana 1.....Gata!!! A durat 1 iteratii
Se creeaza coloana 2.....Gata!!! A durat 1 iteratii
Se creeaza coloana 3.....Gata!!! A durat 2 iteratii
Se creeaza coloana 4.....Gata!!! A durat 3 iteratii
Se creeaza coloana 5.....Gata!!! A durat 1 iteratii
Se creeaza coloana 6.....Gata!!! A durat 6 iteratii
Se creeaza coloana 7.....Gata!!! A durat 1 iteratii
Se creeaza coloana 8.....Gata!!! A durat 1 iteratii
Se creeaza coloana 9.....Gata!!! A durat 3 iteratii
Se creeaza coloana 10.....Gata!!!A durat 1 iteratii
Se creeaza coloana 11.....Gata!!!A durat 5 iteratii
Se creeaza coloana 12.....Gata!!!A durat 1 iteratii
Se creeaza coloana 13.....Gata!!!A durat 1 iteratii
Se creeaza coloana 14.....Gata!!!A durat 1 iteratii
Se creeaza coloana 15.....Gata!!!A durat 4 iteratii
Se creeaza coloana 16.....Gata!!!A durat 2 iteratii
Se creeaza coloana 17.....Gata!!!A durat 3 iteratii
Se creeaza coloana 18.....Gata!!!A durat 6 iteratii
Se creeaza coloana 19.....Gata!!!A durat 6 iteratii
Se creeaza coloana 20.....Gata!!!A durat 41 iteratii
Se creeaza coloana 21.....Gata!!!A durat 24 iteratii
Se creeaza coloana 22.....Gata!!!A durat 55 iteratii
Se creeaza coloana 23.....Gata!!!A durat 12 iteratii
Se creeaza coloana 24.....Gata!!!A durat 108 iteratii

Se creeaza coloana 25.....Gata!!!A durat 146 iteratii
Se creeaza coloana 26.....Gata!!!A durat 41 iteratii
Se creeaza coloana 27.....Gata!!!A durat 934 iteratii
Se creeaza coloana 28.....Am incercat 100000 de iteratii, si nu am gasit nici un vector potrivit.

Reiau creearea matricii. Mai am 4 incercari!

Se creeaza coloana 1.....Gata!!! A durat 1 iteratii
Se creeaza coloana 2.....Gata!!! A durat 2 iteratii
Se creeaza coloana 3.....Gata!!! A durat 1 iteratii
Se creeaza coloana 4.....Gata!!! A durat 1 iteratii
Se creeaza coloana 5.....Gata!!! A durat 2 iteratii
Se creeaza coloana 6.....Gata!!! A durat 1 iteratii
Se creeaza coloana 7.....Gata!!! A durat 1 iteratii
Se creeaza coloana 8.....Gata!!! A durat 1 iteratii
Se creeaza coloana 9.....Gata!!! A durat 7 iteratii
Se creeaza coloana 10.....Gata!!! A durat 6 iteratii
Se creeaza coloana 11.....Gata!!! A durat 1 iteratii
Se creeaza coloana 12.....Gata!!! A durat 7 iteratii
Se creeaza coloana 13.....Gata!!! A durat 5 iteratii
Se creeaza coloana 14.....Gata!!! A durat 8 iteratii
Se creeaza coloana 15.....Gata!!! A durat 1 iteratii
Se creeaza coloana 16.....Gata!!! A durat 8 iteratii
Se creeaza coloana 17.....Gata!!! A durat 6 iteratii
Se creeaza coloana 18.....Gata!!! A durat 1 iteratii
Se creeaza coloana 19.....Gata!!! A durat 4 iteratii
Se creeaza coloana 20.....Gata!!! A durat 7 iteratii
Se creeaza coloana 21.....Gata!!! A durat 14 iteratii
Se creeaza coloana 22.....Gata!!! A durat 1 iteratii
Se creeaza coloana 23.....Gata!!! A durat 22 iteratii
Se creeaza coloana 24.....Gata!!! A durat 48 iteratii
Se creeaza coloana 25.....Gata!!! A durat 37 iteratii
Se creeaza coloana 26.....Gata!!! A durat 147 iteratii
Se creeaza coloana 27.....Gata!!! A durat 40 iteratii
Se creeaza coloana 28.....Gata!!! A durat 2 iteratii
Se creeaza coloana 29.....Gata!!! A durat 132 iteratii

Matricea H a fost creata cu succes...

Aceasta este matricea H...

0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 Greutatea randului este: 6
1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 Greutatea randului este: 6
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 Greutatea randului este: 6
0 0 0 0 0 0 1 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 Greutatea randului este: 6
0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 Greutatea randului este: 5
0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0 Greutatea randului este: 7
0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 Greutatea randului este: 7
0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 Greutatea randului este: 6
1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 Greutatea randului este: 5
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 0 0 1 0 0 0 1 Greutatea randului este: 7
0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 Greutatea randului este: 6
0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 Greutatea randului este: 5
1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 Greutatea randului este: 6
0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 Greutatea randului este: 6
0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 Greutatea randului este: 6

Matricea H sub forma H = [At|I]...

0 1 0 1 1 0 1 1 1 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Greutatea randului este: 11
0 0 0 1 0 1 0 1 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 Greutatea randului este: 7
1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 Greutatea randului este: 9
1 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 Greutatea randului este: 8
0 1 1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 Greutatea randului este: 7
1 0 0 0 1 1 0 1 0 0 1 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 Greutatea randului este: 9
1 0 1 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 Greutatea randului este: 9
0 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 Greutatea randului este: 7
1 1 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 Greutatea randului este: 7
1 1 0 0 0 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 Greutatea randului este: 9
0 0 0 1 1 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 Greutatea randului este: 8
0 1 0 0 0 1 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 Greutatea randului este: 7
1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 Greutatea randului este: 8
0 0 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 Greutatea randului este: 8
0 0 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 Greutatea randului este: 8

Matricea G...

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1 0 0 1 0 0 Greutatea randului este: 8
0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0 Greutatea randului este: 6
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 Greutatea randului este: 6
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1 0 0 1 1 Greutatea randului este: 8
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 1 0 0 0 0 Greutatea randului este: 6
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 1 1 0 Greutatea randului este: 8
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 1 1 1 1 0 1 Greutatea randului este: 10
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 Greutatea randului este: 10
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 Greutatea randului este: 4
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 1 1 1 Greutatea randului este: 10
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 Greutatea randului este: 8
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 Greutatea randului este: 10
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 0 1 0 1 0 0 1 1 1 Greutatea randului este: 10
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1 0 0 1 1 0 0 Greutatea randului este: 8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0 0 1 1 Greutatea randului este: 10

SIMULARE CU CODURI LDPC PENTRU CANAL AWGN CU SIGMA PATRAT 0.5

==========

Se creeaza un mesaj aleator. Acesta este:
1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1

Luam primi 15 biti si ii codam. Vom coda deci urmatoare secventa din mesaj:
1 1 0 1 0 0 1 0 1 1 1 0 0 1 0

Mesajul codat(din mesajul initial si matricea G) este:
1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0

Mesajul primit de canal dupa ce a fost trecut prin zgomot (AWGN) este:
1 1 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 1 0
3 erori.

Mesajul decodat este:
1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0

Sindrom = Zero
Decodare cu succes. A durat 4 iteratii

Mesajul final (primi 15 biti din mesajul decodat):
1 1 0 1 0 0 1 0 1 1 1 0 0 1 0
****
Mesajul initial a fost:
1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1

Luam urmatori 15 biti si ii codam. Vom coda deci urmatoare secventa din mesaj:
0 0 0 0 0 1 1 1 1 0 0 1 1 0 1

Mesajul codat(din mesajul initial si matricea G) este:
0 0 0 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0

Mesajul primit de canal dupa ce a fost trecut prin zgomot (AWGN) este:
0 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0
3 erori.

Mesajul decodat este:
0 0 0 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0

Sindrom = Zero
Decodare cu succes. A durat 9 iteratii

Mesajul final (primi 15 biti din mesajul decodat):
0 0 0 0 0 1 1 1 1 0 0 1 1 0 1
****
Mesajul initial creat aleator este:

```
        1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1
--------------------------------------------------------------------------------
 Mesajul codat complet este:
        1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 1 0
0 0 0 0 0 1 0
--------------------------------------------------------------------------------
 Mesajul primit complet este:
        1 1 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0
0 0 0 0 0 1 0
--------------------------------------------------------------------------------
 Mesajul decodat complet este:
        1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 1 0
0 0 0 0 0 1 0
--------------------------------------------------------------------------------
 Mesajul final complet este:
        1 1 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1
--------------------------------------------------------------------------------
 Numarul de biti eronati este: 0
--------------------------------------------------------------------------------
```

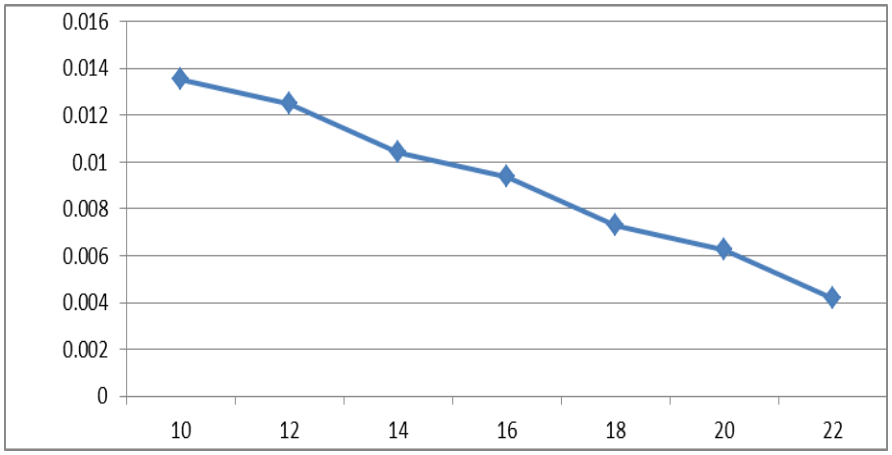


Fig 8.1 BER pentru coduri LDPC

Codurile LDPC generate au următorii parametrii: mărime mesaj 15, mărime cod 30 şi mărime mesaj 990

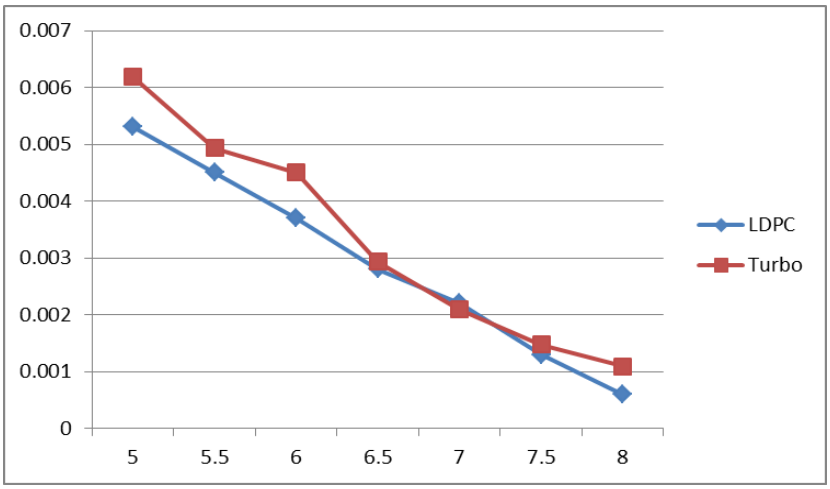


Fig. 8.2 Comparaţie performanţe BER

Coduri TURBO cu parametrii: interleaver de Tip par/impar, număr iteraţii 10, dimensiune fişier 10000

Coduri LDPC cu următorii parametrii: mărime mesaj 15, mărime cod 30 şi mărime mesaj 9990

Capitolul IX. Concluzii

Codurile LDPC au fost introduce iniţial de Gallager în 1962, dar, abia în anul 1996, au fost redescoperite de MacKay şi de R.Neal, care au arătat căştigurile de performanţă a acestor coduri.

La codurile LDPC este mai puţin importantă tradiţionala distanţă minimă a codului decât **greutatea minimă** asociată matricii de verificare a parităţii.

Din punct de vedere al **vitezei,** se estimează că decodificatorii analogici vor funcţiona la viteze cu aproximativ două ordine de mărime mai repede decât decodificatorii digitali. Acest lucru este susţinut de rezultatele obtinuţe recent de la decodificatorii analogici.

Am observat în urma simulărilor repetate că, evitându-se prea multe suprapuneri ale biţilor de 1 în coloanele matricii de paritate H, circumferinţa creşte iar **decodarea este mai rapidă**. În acelaşi timp, eficienţa codurilor (la codificare) este permisă de proprietatea matricii de control a parităţii de a fi slab populată.

Complexitatea algoritmului este mai mică decât complexitatea algoritmilor altor metode de codare/decodare.

Ca performanţă ajung codurile TURBO, însă **costul de implementare** este cu mult mai scăzut.

Pentru performanţă este important ca **numărul de interaţii** să depăşească valoarea de 1000 spre deosebire de TURBO, unde după 20 de iteraţii îmbunătăţirea performanţelor este nesemnificativă.

Bibliografie

[1] Berrou, C., Glavieux, A., and Thitimajshima, P., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," *Proceedings of ICC 1993*, Geneva, Switzerland, pp. 1064-1070, May 1993.

[2] Bernard Sklar: "*A primer on turbo code concepts*", IEEE Comunications Magazine, vol.35, no. 12, pp. 94-98, Dec.1997

[3] Benedetto, S., and Montorsi, G., "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Transactions on Information Theory*, Vol. 42, No. 2, pp. 409-428, March 1996.

[4] P. Jung, M. Nasshan, and J. Blanz, "Application of turbo-codes to a CDMA mobile radio system using joint detection and antenna diversity", in *Proceedings of the IEEE Conference on Vehicular Technology*, pp. 770-774, 1994

[5]. R. G. Gallager, Low-density parity-check codes. Cambridge, Press, 1963.

[6]. ----, "Low-density parity check codes," IRE Trans. Inform. Theory, vol. IT-8, pp. 21-28, Jan. 1962.

[7] C. E, Shannon, "A. mathematical theory of communication," Bell System Technical Journal, vol 27, pp. 379-423, 623-665, Jury, Oct. 1948.

[8]. V.Zybalov and M. S. Pinsker, ―Estimation of the error-correcting complexity of Gallager low-density codes‖ Problems of Info. Trans, vol. 11, no. 1,pp. 23-26, 1975.

[9]. G. A. Margulis, "Explicit constractions of graphs without short cycles and low density codes‖, Combinatorica, vol. 2. no. 1, pp. 71--78, 1982.

[10] R. M. Tanner, "A recursive approach to low complexity codes," IEEE Trans, In- form. Theory, vol IT-27, pp. 533-547, Sept. 1981.

[11] M. G. Luby, M. A. Shokrolloahi, M. Mizenmacher, and D. A. Spielman, "Improved low-density parity-check cedes using irregular graphs", IEEE Tmns, Inform, Theory, vol. 47, no, 2, pp. 585-598, Feb. 2001.

[12] T. Tian, C. Jones, J. D. Villaseaor, and R. D. Wesel, "Construction of irregular LDPC codes with low error floors," in Proc. ICC 2003, vol. 5, Anchorage, AK, 2003, pp. 3125-3129.

[13]. C. E, Shannon, "A. mathematical theory of communication," Bell System Technical Journal,1948.

[14]. X. L, Massey, Threshold decoding, Cambridge, MA; MIT Press, 1963.

[15]. R. Kotter .and P. O. YoatobeL, —”

Graph-covers and iterative decoding of finite length codes," 2003

[16]. F.R Kschischang and B.J.Frey, **—**”Iteratice decoding of compound code by probability propagation in graphical models”, IEEE

[17]. R, L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics.

[18]. N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Univ. Linkoping, 1996. **[19]. N. Wiberg, R. Kotter, and H.-A. Loeliger**, **—**Codes and iterative decoding on general graphs." European Trans. on Telecomnmn,1995.

[20]. F. R Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm”,IEEE Trans. Inform. Theory, Feb.2001

[21]. D. J. C. MacKay,S. T. Wilson, and M. C. Davey, “Comparison of constructions of irregular Gallager codes”, IEE Trans. Communications, vol 47,no. 10,pp 1449- 1454,Oct 1999.

[22]. L. Ping, W. K. Leung, and N. Phamdo, "Low density parity check codes with .semi-random parity check matrix," Electron. Lett., vol. 35, no. 1, pp. 38-39, Jan.1999.

[23]. M. Rasbidpour and S, H. Jamali, "Low-density parity-check codes with simple irregular semi-random parity-check matrix for finite-length applications," in Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Communication, PIMRC2003, vol. 1, Beijing, China, 2003, pp. 439-443.

[24]. M. Yang, W. E. Ryan, and. L. Yan, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes”, IEEE Tmns. Communications, vol. 52, no. 4, pp, 564-571, Apr. 2004.

[25]. R. Lucas, M. P. C. Fossorier, Y, Kou, and S, Lin, "Iterative decoding of one-step majority logic deductible codes based on belief propagation," IEEE Trans. Communications, vol. 48, no. 6, pp. 931-937, June 2000.

[26]. Y. Kou, S, Lin, and M, P. Foasorier, "Low density parity check codes based on finite geometries: A rediscovery and new results," IEEE Trans. Inform: Theory, vol. 47, no. 7, pp. 2711-2736, Nov. 2001.

[27]. S. J. Johnson and S. R. Weller, “A family of irregular LDPC codes with low encoding complexity," IEEE Commun. Lett., vol. 7, no. 25 pp. 79-81, Feb. 2003.

[28] . W. W. Peterson, „Error Correcting Codes” Cambridge, MA: MIT Press, 1961.

[29]. M. Sipser si D. A. Spielman, „Expander codes", IEEE Trans, Inform, Theory, ol. 42, no. 6, pp. 1710-1722, Nov. 1996

[30]. D. A. Spielman, "Linear-time encodable and decodable error-correcting codes", IEEE Trans. Inform. Theory, vol. 42, no. 6, pp. 1723 -1731, Nov. 1996.

[31] M. G. Luby, M. A. Shokrolloahi, M. Mizenmacher, and D. A. Spielman, "Improved low-density parity-check cedes using irregular graphs", IEEE Tmns, Inform, Theory, vol. 47, no, 2, pp. 585-598, Feb. 2001.

[32]. R. Echard si S. C. Chang, „The -rotation low-density parity check code", In Proc. GLOBECOM 2001, vol. 2, San Antonio, TX, 2001, pp. 980-484.

[33]. _____ "The extended irregular ar-rotation low-density parity check codes", IEEE Commun. Lett, vol. 7, no. 5, pp. 230-232, May 2003.

[34]. T. Zhang si K. K. Parhi, "Joint (3,k)-regular LDPC code and decoder/encoder design", IEEE Trans. Sig. Proc., vol. 52, no. 4, pp. 1065-1079, Apr. 2004.

[35]. T. J. Richardson si R. L, Urbanke, "Efficient encoding of low-density parity-check codes". IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 638-656, Feb. 2001.

[36]. T. J. Richardson, A. Shokrollahi, R. Urbanke, "Design of provably good low-density parity-check codes", in Proc. Int. Symp. Information Theory, Sorrento., Italy, 2000, p. 199.

[37] F. Lustenberger, "On the design of analog iterative VLSI decoders," Ph.D. dissertation, ETH, Zikieh, Switzerland, 2000.

[38] S. B. Wicker, Error Control Systems for Digital Communication and Storage. Prentice Hall, 1995.

[39] M. G. Luby, M. A. Shokrolloahi, M. Mizenmacher, and D. A. Spielman, "Improved low-density parity-check cedes using irregular graphs", IEEE Tmns, Inform, Theory, vol. 47, no, 2, pp. 585-598, Feb. 2001.

[40] C.Di, D. Proietti, I. E. Teletar, T. J. Richardson, and R. L. Urbanke, —Finite-length analysis of low-deasity parity-check codes on the binary erasure channel", IEEE Trans. Inform. Theory, vol. 48, no. 6, pp. 1570-1579, Jun. 2002.

[41] T. J. Richardson and R. L. Urbanke, —The capacity of low-density parity-check codes under message-passing decoding', IEE, Feb 2001

[42] Ivan B. Djordjevic, "PMD compensation in fiber-optic communication systems with direct detection using LDPC-coded OFDM"

[43]. Xiao Song, Li Yunsong, Bai Baoming, ZhouYouxi, "The Key Technologies of Deep Space Communications**"**

[44]. Sony Corporation, Tokio Institute of Technology, Press release, February 17, 2012
[45] Da Xinyu, Wang Yanling, Xie Tiecheng, "Performance of LDPC Codes for Satellite Communication in Ka Band"
[46] Prof. Warren J. Gross, "Forward Error-Correction in High-Speed Optical Communications", International Workshop on Trends in Optical Technologies CPqD, Campinas, Brazil, May 9, 2012;
[47] Murat Arabaci, "High-Rate Nonbinary Regular Quasi-Cyclic LDPC Codes for Optical Communications", Journal of Lightwave Technology, vol. 27, no. 23, December 1, 2009, pp. 5261-5267;
[48] Ivan B. Djordjevic, "Nonbinary LDPC Codes for Optical Communication Systems", IEEE Photonics Technology Letters, vol. 17, no. 10, October 2005, pp. 2224-2226;