

Coduri corectoare de erori utilizate în sistemele moderne de transmisiuni

- acest capitol al cursului va face o scurtă prezentare a câtorva clase de coduri corectoare de erori utilizate în sistemele moderne de transmisiuni numerice.

- vor fi prezentate codurile LDPC și turbocodurile, care alături de codurile convoluționale și codurile Reed-Solomon sunt utilizate în cea mai mare parte a sistemelor comerciale de transmisiuni.

1. Codurile LDPC – Low Density Parity Check Codes

1.1. Generalități

- Codurile LDPC, introduse în 1963 de Robert G. Gallager, reprezintă o familie de coduri bloc liniare care se obțin pornind de la o matrice „rară” de control a parității H , care conține un număr mic de elemente nenule și un număr mare de elemente de 0 („sparse matrix”).

- Codurile LDPC au fost reinventate de MacKay și Neal la mijlocul anilor 1990, la scurt timp după ce Berrou și alții au introdus turbocodurile și au demonstrat importanța tehnicilor iterative de decodare pentru apropierea performanțelor sistemelor practice de transmisie de capacitatea ideală a canalului ($C = \log_2(1+\rho)$ [bps/Hz]).

- Codurile LDPC pot fi privite ca și coduri bloc liniare, descrise de matricea de control a parității H , ale cărei dimensiuni sunt $M \times N$,

- La aceste coduri ecuația: $Hv^t = 0$ este satisfăcută pentru toate cuvintele de cod v .

- Fiecare linie a matricii H definește o ecuație de control a parității care trebuie să fie satisfăcută de fiecare cuvânt de cod v . Rezolvând sistemul format de aceste ecuații se determină biții de control specifici fiecărui cuvânt de cod.

- Spre deosebire de matricea de control H a unui cod Hamming, matricile de control a parității care determină codurile LDPC au următoarele proprietăți specifice:

- dimensiunile M și N ale matricii H sunt mult mai mari față de dimensiunile unei matrici de control specifice unui cod Hamming. S-au impus aceste dimensiuni astfel încât, prin codare, să ne apropiem cât mai mult de limita teoretică de codare impusă de teorema lui Shannon, în care se presupune un cod corector cu lungimea infinită a cuvântului de cod
- matricile H sunt definite, prin construcție, într-o formă (cuasi-)nesistematică și conțin un număr de elemente de 1 (pentru codurile binare) mult mai mic decât numărul de elemente de 0;
- matricea de control a parității ce definește un cod LDPC $H(j, k)$ are exact j elemente de 1 pe fiecare coloană și exact k elemente de 1 pe fiecare linie. Acestea înseamnă că fiecare bit al cuvântului de cod intră în j ecuații de control, iar fiecare ecuație de control conține k biți.

- Clasa matricilor de control H pentru codurile LDPC trebuie să satisfacă următoarele constrângeri:

- fiecare coloană trebuie să conțină un număr mic, fix, de elemente de 1, nota cu j ;
- fiecare linie trebuie să conțină un număr mic, fix, de elemente de 1, notat cu k .

- Aceasta echivalează cu descrierea matricii H de către un graf bipartit cu două tipuri de noduri:

- N noduri de simbol care corespund fiecărui bit din cuvântul de cod, (coloanele matricii) și
- M noduri de control care corespund ecuațiilor de control de paritate, noduri reprezentate de liniile matricii- *se va reveni asupra acestui aspect*

- Constrângerile menționate anterior impun ca fiecare nod de simbol (bit) să fie conectat cu j noduri de verificare a parității și fiecare nod de verificare a parității să fie conectat cu k noduri de simbol (bit).

- Codurile definite prin astfel de grafuri pot fi decodate cu algoritmul Sumă-Produs (Sum Product Algorithm – SPA sau Message-Passing –MP)). Decodarea cu acest algoritm asigură performanțele unui decodor bazat pe principiul ML (*Maximum-Likelihood*) numai dacă grafurile sunt fără cicluri cu lungime mică, sau altfel spus, sunt de tipul „4-cycle free” - *se va reveni asupra acestui aspect*.

- Această cerință este echivalentă cu condiția ca matricea H să nu aibă două linii (oricare două linii) în care să existe elemente de 1 suprapuse pe mai mult de o singură poziție.

- Pornind de la această condiție, limitele ce se impun dimensiunilor M (linii) și N (coloane) ale matricii H sunt interconținute de relația:

$$N \leq \frac{M(M-1)}{j(j-1)} \quad (1)$$

1.2. Tipuri de coduri LDPC

- După modul de generare a matricii de control H , se pot defini mai multe clase de coduri LDPC, dintre care amintim:

- construite aleator (random)

- construite pe baza unor structuri regulate: de tip „array-based” sau de tip “quasi-cyclic”
- pe baza unor construcții combinatoriale
- pe baza unor construcții geometrice
- construite pe baza grafurilor
- există și alte construcții mai recente

1.3. Coduri LDPC random

- în literatură sunt prezentate mai multe metode de generare a matricilor H care definesc codurile LDPC de tip random; prezentarea acestor metode depășește cadrul acestui curs

- pentru a asigura existența a k biți de „1” în fiecare linie (adică o ecuație de control să depindă de k biți de cod) și j biți de „1” în fiecare coloană (adică un bit de cod să fie inclus în j ecuații) ale matricii H , lungimea cuvântului de cod N și numărul de biți de control M trebuie să satisfacă, în cazul acestor coduri, condițiile (1'.a) și (1'.b).

- rata acestui tip de cod poate fi aproximată prin relația (1'.c).

$$N_1 = \frac{k(k+1)}{2}(j-1); \quad a.; \quad j \geq 2; \quad k > j; \quad N \geq N_1 \text{ și } M = \left\lceil N \cdot \frac{j}{k} \right\rceil \in N; \quad b.; \quad R_c \cong \frac{k-j}{k}; \quad c.; \quad (1')$$

1.3. Matrici de control a parității bazate pe coduri array pentru coduri LDPC

- Vom defini acest tip de coduri LDPC prin intermediul a trei parametri: un număr prim p și două numere întregi k și j astfel încât $j < k \leq p$.

- Se construiește mai întâi o matrice H de dimensiuni $M \times N$, cu $M = jp$, $N = kp$, de forma:

$$H = \begin{bmatrix} I & I & I & \dots & I \\ I & \alpha & \alpha^2 & \dots & \alpha^{k-1} \\ I & \alpha^2 & \alpha^4 & \dots & \alpha^{2(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & \alpha^{j-1} & \alpha^{2(j-1)} & \dots & \alpha^{(j-1)(k-1)} \end{bmatrix} \quad (2)$$

unde I este matricea unitate de dimensiune $p \times p$, iar α este o matrice de dimensiune $p \times p$ obținută prin deplasare la dreapta sau la stânga, cu câte o poziție, a coloanelor matricii I .

- Pentru $p = 5$ matricea α va avea una din următoarele forme:

$$\alpha_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{sau} \quad \alpha_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3)$$

- Această formă a matricii H nu are rangul M , pentru a permite generarea unică a M biți de control prin rezolvarea ecuației (4)

$$H \cdot v^t = 0 - \text{sistem de } N \text{ ec. cu } M \text{ nec. compatibil nedeterminat} \quad (4)$$

- Una din metodele prin care se poate asigura existența unei soluții unice a ecuației (4) constă în aducerea matricii H la o formă triangularizată

- Pentru aceasta se definește o matrice H_s' obținută prin deplasarea ciclică a coloanelor din matricea H (fiecare element al matricii H este de fapt o matrice de dimensiune $p \times p$). Deplasarea ciclică a fiecărei coloane se face astfel încât pe diagonala principală a submatricii de dimensiune $jp \times jp$ să avem ca elemente doar matrici unitate I de dimensiune $p \times p$; prin permutări de coloane (linii), rezultă matricea completă (5):

$$H'_s = \begin{bmatrix} I & I & I & \dots & I & I & \dots & I \\ \alpha^{k-1} & I & \alpha & \dots & \alpha^{(j-2)} & \alpha^{j-1} & \dots & \alpha^{k-2} \\ \alpha^{2(k-2)} & \alpha^{2(k-1)} & I & \dots & \alpha^{2(j-3)} & \alpha^{2(j-2)} & \dots & \alpha^{2(k-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \dots & \vdots \\ \alpha^{(j-1)(k-j+1)} & \alpha^{(j-1)(k-j+2)} & \dots & \dots & I & \alpha^{j-1} & \dots & \alpha^{(j-1)(k-1)} \end{bmatrix} \quad (5)$$

- Apoi se triangularizează matricea H_s' înlocuind elementele de sub diagonala principală, matrici $p \times p$, cu matrici O de dimensiune $p \times p$ care conțin numai elemente de 0. Astfel se va obține matricea de control triangularizată H_T (6) pentru codurile LDPC care este tot o matrice de tipul “4-cycle free”:

- Codul LDPC definit de matricea H_T , este de fapt definit de parametrii p, j, k ;

$$H_T = \begin{bmatrix} I & I & I & \dots & I & I & \dots & I \\ O & I & \alpha & \dots & \alpha^{j-2} & \alpha^{j-1} & \dots & \alpha^{k-2} \\ O & O & I & \dots & \alpha^{2(j-3)} & \alpha^{2(j-2)} & \dots & \alpha^{2(k-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \dots & \vdots \\ O & O & \dots & O & I & \alpha^{j-1} & \dots & \alpha^{(j-1)(k-1)} \end{bmatrix} \quad (6)$$

- Aceștia permit construcția matricii de control H_T , determinarea lungimii $N = k \times p$ a cuvântului de cod, a numărului de ecuații de control al parității $M = j \times p$ (egal cu numărul biților de control) și a lungimii blocului de informație $K = (k - j) \times p$.

- rata codului este dată de relația:
$$R = \frac{K}{N} = \frac{(k-j) \cdot p}{k \cdot p} = 1 - \frac{j}{k} \quad (7)$$

- matricea H_T poate fi calculată off-line și memorată, sau se pot determina pozițiile biților de „1”, în funcție de indecșii de linie și coloană, iar matricea poate fi generată dinamic în timp real, fără a fi necesară ridicarea la putere a matricii α .

1.5. Considerente privitoare la codurile de tip „quasi-cyclic”

- Matricile care definesc acest tip de coduri au trei tipuri de coloane:

- coloanele corespunzătoare biților informaționali
- coloanele biților de control principali (core)
- coloanelor biților de control ai extensiei codului.

- Aceste matrici au două tipuri de linii:

- linii de control principale (corespund primelor două tipuri de coloane)
- linii de control ale extensiei codului

- Structura matricii este complexă, modalitatea de construcție a ei urmărind să asigure un compromis între capacitatea de corecție și posibilitatea de procesare paralelă în cazul codării și decodării.

- O introducere în structura codurilor de acest tip care sunt prevăzute pentru interfața radio a sistemului 5G (5GNR) este prezentată în J.H. Bae, A.Abotabl, H-P. Lin, K-B. Song, J. Lee, “An overview of channel coding for 5G NR cellular communications”, APSIPA Transactions on Signal and Information Processing, vol. 8 2019, <https://www.cambridge.org/core/journals/apsipa-transactions-on-signal-and-information-processing/article/an-overview-of-channel-coding-for-5g-nr-cellular-communications/CF52C26874AF5E00883E00B6E1F907C7>

1.4. Codarea codurilor LDPC

- Considerând cuvântul de cod $v = [c_0, \dots, c_{M-1}, i_0, \dots, i_{(N-M)-1}]$, biții de control c_m se pot calcula, în funcție de biții informaționali i_l , prin rezolvarea sistemului de M ecuații liniare:
$$H \cdot v^t = 0 \quad (8)$$

- Această abordare are două dezavantaje majore:

- pentru valori mari ale parametrilor j și sau p , M ia valori mari necesitând un mare volum de calcule, ceea ce conduce la creșterea timpului și/sau resurselor hardware necesare procesării.
- are nevoie de toți biții informaționali i_l , $l = 0, \dots, (N-M)-1$, în același timp; această cerință induce o întârziere suplimentară de un cuvânt de cod în sistemul de transmisie.

- Aceste dezavantaje pot fi evitate de metoda de codare, mai simplă și mai rapidă, descrisă mai jos.

- Matricea H_T , ($M \times N$), este împărțită în două matrici D și E , reținând primele M coloane în matricea D și restul de $(N-M)$ coloane în matricea E . Cele două matrici au dimensiunile:

$$D \rightarrow (M \times M); \quad E \rightarrow (M \times (N-M)); \quad (9)$$

- Matricea D este pătrată și are determinant nenul, datorită construcției matricii H , deci are inversă D^{-1} .

- Rezultă că ecuația codării (8) poate fi rescrisă sub forma (10), unde matricea F ($M \times (N-M)$) este matricea de codare:

$$\begin{aligned} [H] \cdot [v]^t = [0] &\Leftrightarrow [D] \cdot \begin{bmatrix} c_0 \\ \vdots \\ c_{M-1} \end{bmatrix} + [E] \cdot \begin{bmatrix} i_0 \\ \vdots \\ i_{(N-M)-1} \end{bmatrix} = [0] \Rightarrow [D]^{-1} \cdot [D] \cdot \begin{bmatrix} c_0 \\ \vdots \\ c_{M-1} \end{bmatrix} + [D]^{-1} \cdot [E] \cdot \begin{bmatrix} i_0 \\ \vdots \\ i_{(N-M)-1} \end{bmatrix} = [0] \\ \Rightarrow \begin{bmatrix} c_0 \\ \vdots \\ c_{M-1} \end{bmatrix} &= [D]^{-1} \cdot [E] \cdot \begin{bmatrix} i_0 \\ \vdots \\ i_{(N-M)-1} \end{bmatrix} = [F] \cdot \begin{bmatrix} i_0 \\ \vdots \\ i_{(N-M)-1} \end{bmatrix}; [F] \Leftrightarrow [M \times (N-M)] \end{aligned} \quad (10)$$

- Prin dezvoltarea matricii de codare F , ($M \times (N-M)$), dacă notăm cu $[f_i]$ coloanele sale (fiecare din ele un vector de M biți, relația (10) poate fi exprimată sub forma:

$$[f_0] \cdot i_0 + \dots + [f_{(N-M)-1}] \cdot i_{(N-M)-1} = [C] \quad (11)$$

- Matricea F , calculată off-line, este stocată coloană cu coloană în memorie; fiecare bit de informație se înmulțește cu coloana cu același index, iar vectorii-produs rezultați sunt acumulați, astfel obținându-se vectorul biților de control $[C]$; pentru codurile de tip “array” sau cuasi-ciclice matricea F poate fi calculată în timp real, în cazul codurilor cu lungime relativ mică a cuvintelor de cod care au un număr relativ mic de biți/coloană

- Această metodă de codare este mai rapidă decât cea descrisă de ecuația (8) și calculează biții de control în mod paralel folosind un bit de informație la fiecare tact

- De asemenea volumul de calcule necesar determinării biților de control este repartizat uniform în timp.

- Pentru codurile lungi, această abordare necesită accesarea unui număr mare de locații de memorie pentru transferul unei coloane și dimensiuni mari ale acumulatorului. De aceea, pentru aceste coduri s-au ales metode de codare care calculează pozițiile biților de “1” în cadrul procesului de calcul al biților de control. Considerente privind acest tip de metode de codare sunt prezentate în referința menționată anterior.

- Matricea F_s a unui cod prescurtat, $K' = K - U$, se obține eliminând din matricea F un număr de U coloane care corespund biților informaționali care se elimină din cuvântul de cod complet;

- Cele U coloane pot fi :

- ultimele U coloane din partea dreaptă a matricii F ; ele corespund biților info cu index $N-U-1, \dots, N-1$

- coloanele ai căror indecși corespund biților care au “girth”-ul cel mai mic în graful Tanner asociat

- În standardul 5G NR este prevăzută și posibilitatea puncturării codurilor LDPC, fiind indicate unele “pattern-uri” de puncturare.

1.5. Decodarea codurilor LDPC – algoritmul Sum-Product (Message Passing)

- Algoritmul de decodare SP (MP) este un algoritm iterativ care nu urmărește determinarea cuvântului de cod cel mai apropiat de secvența recepționată, în sensul unei metrici;

- El urmărește decodarea fiecărui bit folosind informațiile primite privitoare la toți biții cuvântului de cod și utilizând corelațiile dintre acești biți impuse de ecuațiile de control

- Algoritmul MP utilizează ca date de intrare probabilitățile *a posteriori* ale biților cuvântului de cod, furnizate de funcția de demapare soft (vezi paragraful următor, pag. 10).

- La fiecare iterație, algoritmul modifică valorile acestor probabilități ale unui bit, în funcție de probabilitățile *a posteriori* ale celorlalți biți care intră în aceleași ecuații de control cu bitul dat. Aceste actualizări sunt făcute pentru fiecare bit.

- La sfârșitul fiecărei iterații, în pasul 3, se ia o decizie asupra valorii logice a biților obținuți în iterația respectivă, folosind criteriul lui Bayes, iar apoi, în pasul 4, cu biții obținuți se calculează sindromul, cu care se verifică dacă toți biții cuvântului de cod sunt corecți.

- Dacă acesta este nul, decodarea se consideră încheiată; dacă nu, atunci se trece la o nouă iterație, până la atingerea numărului maxim de iterații I .

- Dacă s-a atins numărul maxim de iterații permise, atunci se livrează biții decizi după ultima iterație, chiar dacă sindromul lor e nenul. Alternativ, în acest caz se poate “șterge” acest cuvânt de cod (erasure).

- Pentru a exemplifica acest concept se consideră codul LDPC ($k = 4$, $j = 3$, $p = 5$, $N=20$, $M=15$). Acest cod este descris de matricea H completă dată de (12); sistemul de ecuații de control a parității, construit pe baza (8) este descris de (13). **Notă: acest cod nu este de tip array.**

- Codul are $N-M = 5$ biți info și $M = 15$ biți de control

- Fiecare bit (x_i) intră în $j = 3$ ecuații de control (poziții de “1” pe coloana i)

- În fiecare ecuație l de control intră $k = 4$ biți (poziții de “1” pe linia l)

- Graful bipartit asociat codului LDPC ($k = 4, j = 3, p = 5$) este dat în figura 1

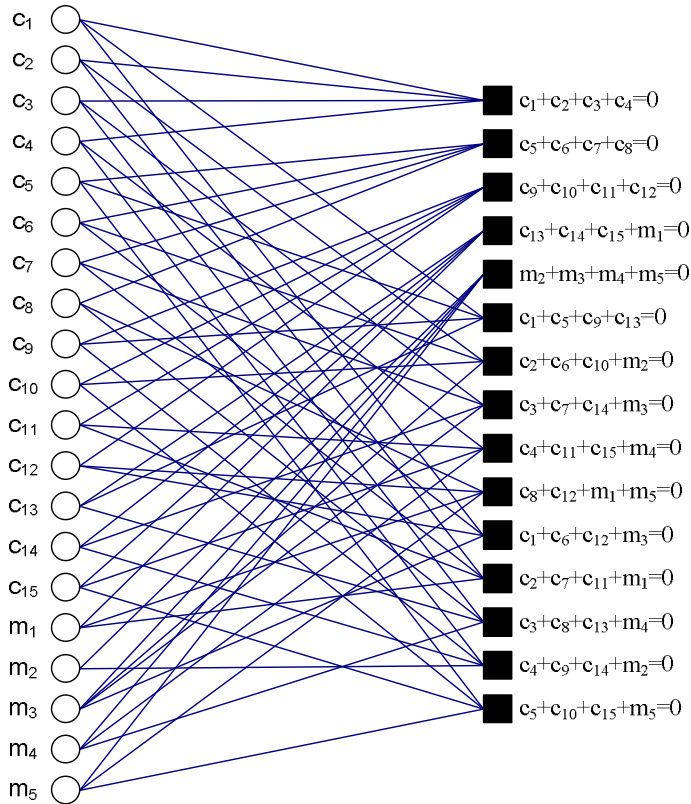


Figura 1 Graful bipartit corespunzător matricii de control a parității

Etapele algoritmului de decodare MP

Pasul 0 – inițializare

- Nodurile de bit sunt inițializate cu probabilitățile *a posteriori* inițiale, P_j^0 (probabilitatea ca bitul j din cuvântul recepționat să fie 0) și P_j^1 (probabilitatea ca bitul j din cuvântul recepționat să fie 1), determinate de blocul de demapare soft (vezi (26) - par. 1.6, pag. 10) pe baza semnalului recepționat.

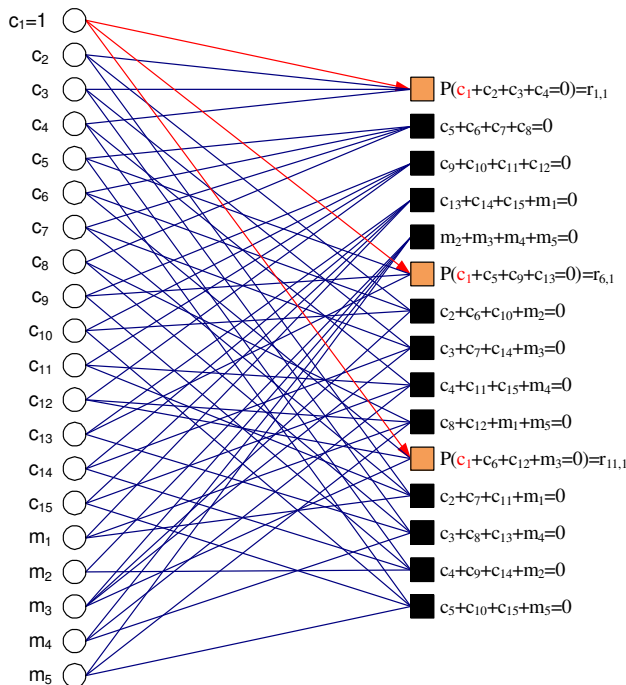


Figura 2 Actualizarea nodurilor de control

- Se poate arăta că probabilitatea $r_{m,n}^1$ ca din $k_m - 1$ biți independenți ai ecuației de control m , **diferiți de bitul n** , un număr impar de biți să fie nenuli este dată de (16);

Pasul 1- actualizarea nodurilor de control

- Se determină probabilitățile $r_{m,n}^x$, care reprezintă probabilitatea ca ecuația de control m să fie satisfăcută dacă bitul n are valoarea x , (0 sau 1).

- Probabilitatea ca prima ecuație de control a sistemului definit de (14) să fie satisfăcută, dacă $c_1=1$, este:

$$\begin{aligned}
 & P(c_1 + c_2 + c_3 + c_4 = 0 \mid c_1 = 1) \\
 &= P(c_2 + c_3 + c_4 = 1) = \\
 &= P_2^1 \cdot P_3^0 \cdot P_4^0 + P_2^0 \cdot P_3^1 \cdot P_4^0 + \\
 &+ P_2^0 \cdot P_3^0 \cdot P_4^1 + P_2^1 \cdot P_3^1 \cdot P_4^1
 \end{aligned} \tag{15}$$

- Din (15) rezultă că, dacă bitul n are valoarea 1, atunci ecuația m de control al parității este satisfăcută numai dacă între ceilalți $k-1$ biți care intră în această ecuație există un număr impar de biți care au valoarea 1. Dacă bitul n are valoarea 0, ecuația de control este satisfăcută doar dacă un par de biți ce intră în ecuație au valoarea 1.

$$r_{m,n}^1 = \frac{1 - \prod_{\substack{l=1 \\ l \neq n}}^{k_m} (1 - 2q_{l,m}^1)}{2} = \frac{1}{2} \left(1 - \prod_{\substack{l=1 \\ l \neq n}}^{k_m} (q_{l,m}^0 - q_{l,m}^1) \right) \quad (16)$$

unde $q_{n,m}^x$ este probabilitatea ca bitul n al cuvântului de cod recepționat, care intră în ecuația m , să aibă valoarea $x = „0”$ sau „1”, calculată pe baza fazorului y recepționat, vezi (.26). Practic $q_{n,m}^x$, (17), reprezintă mesajul primit de nodul de control m de la nodul de bit n pe parcursul unei iterații, dacă s-a recepționat y :

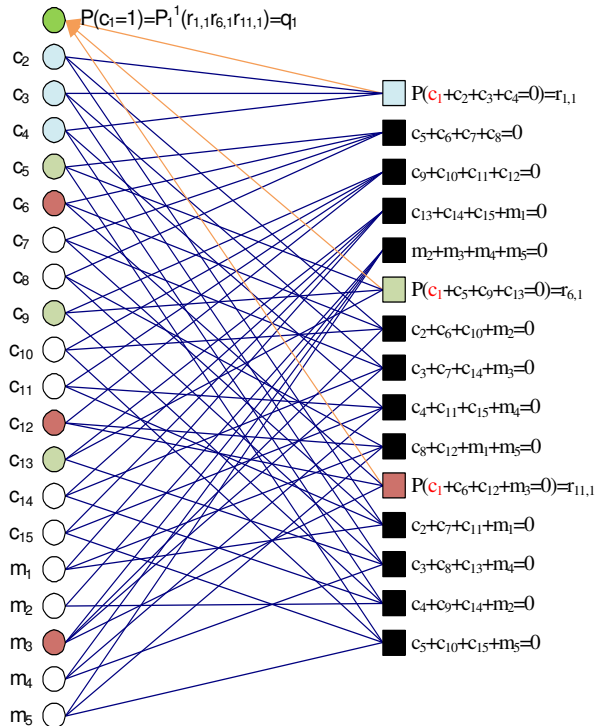
$$q_{n,m}^x = P_n^x = p(y | b_n = x) \quad (17)$$

- Similar se poate arăta că probabilitatea $r_{m,n}^0$ ca din $k_m - 1$ biți independenți ai ecuației de control m , **diferiți de bitul n** , un număr par de biți să fie nenuli este dată de prima ecuație din (18)
- Pentru fiecare nod de control m se determină $r_{m,n}^1$ și $r_{m,n}^0$ adică probabilitățile ca ecuația de control m să fie satisfăcută dacă bitul corespunzător nodului de bit n are valoarea „1” (număr impar de biți 1), respectiv „0” (număr par de biți 1), pentru fiecare nod de bit n care este legat cu nodul de control m . Notând cu $M(n)$ mulțimea ecuațiilor de control în care intra bitul n și cu $N(m)$ mulțimea biților care intră în ecuația m , $r_{m,n}^1$ și $r_{m,n}^0$ se calculează cu (18):

$$\begin{aligned} & \text{for } n = 0, \dots, N - 1 \\ & \quad \text{for } m \in M(n) \\ & \quad r_{m,n}^0 = \frac{1}{2} \left(1 + \prod_{\substack{t \in N(m) \\ t \neq n}} (q_{t,m}^0 - q_{t,m}^1) \right); \\ & \quad r_{m,n}^1 = \frac{1}{2} \left(1 - \prod_{\substack{t \in N(m) \\ t \neq n}} (q_{t,m}^0 - q_{t,m}^1) \right); \\ & \quad \text{end} \\ & \text{end} \end{aligned} \quad (18)$$

Pasul 2 Actualizarea nodurilor de bit

- La sfârșitul fiecărei iterații se determină probabilitățile *a posteriori* pentru fiecare bit, adică probabilitățile $P_n^{l(0)}$ ca bitul transmis pe poziția n să fie „1” (sau „0”) condiționat de setul de simboluri y recepționate (din care se obțin probabilitățile *aposteriori* ale biților) și de evenimentul $S_l(S_0)$ ca cele j ecuații de control, în care intră bitul n , $\{M(n)\}$, să fie satisfăcute pentru $b_n = l(0)$, vezi figura 3.



- Ecuațiile de control fiind independente, probabilitatea ca cele $\{M(n)\}$ ecuații de control în care intră bitul n să fie satisfăcute este produsul probabilităților ca fiecare din aceste ecuații să fie satisfăcută, adică (19).

Notă: $\text{card}(M(n)) = j$, dar mulțimea $\{M(n)\}$ diferă pentru fiecare bit.

$$P(x_n = 1 | S_1, \{y\}) = \alpha_n \cdot P_n^1 \cdot \prod_{i=1}^{M(n)} (r_{i,n}^1) \quad (19)$$

$$P(x_n = 0 | S_0, \{y\}) = \alpha_n \cdot P_n^0 \cdot \prod_{i=1}^{M(n)} (r_{i,n}^0)$$

- În (19) constanta de scalare α_n se alege astfel încât:

$$P(x_n = 1 | S_1, \{y\}) + P(x_n = 0 | S_0, \{y\}) = 1 \quad (20)$$

- α_n are rolul de a repartiza proporțional erorile generate de numărul finit (mic) de biți pe care se face reprezentarea valorilor P_n^x și prelucrarea lor

- Acest pas reprezintă actualizarea probabilităților *aposteriori* ale biților

Figura 3 Actualizarea nodurilor de bit

- Noul mesaj transmis de nodul de bit n spre nodul de control m în iterația următoare va fi (21), în care se

consideră doar efectul ecuațiilor în care intră bitul n , mulțimea $\{M(n)\}$, diferite de ecuația curentă m :

$$q_{n,m}^x = \alpha_{nm} P_n^x \prod_{\substack{l=1 \\ l \neq m}}^{M(n)} r_{l,n}^x \quad (21)$$

- În (21) $q_{n,m}^0, q_{n,m}^1$ reprezintă probabilitățile ca bitul n , „0” sau „1”, să satisfacă toate grupurile (mulțimile) de $M(n) - 1$ ecuații de control. Aceste valori sunt trimise de către nodurile de bit și vor fi folosite în iterația următoare la pasul 1 (actualizarea nodurilor de control). Notă: mulțimile sunt diferite pentru fiecare bit, dar cardinalul lor e același fiind egal cu $j-1$

- În cadrul acestui pas se mai calculează și probabilitățile Pl_n^0, Pl_n^1 ca bitul n , „0” sau „1”, să satisfacă toate ecuațiile de control, care vor fi folosite pentru decizia bitului pe baza criteriului lui Bayes.

- Pe baza relațiilor (19) și (21) pasul 2 al algoritmului de decodare va fi:

for $n = 0, \dots, N - 1$

actualizarea nodurilor de bit →

for $m \in M(n)$

- nu conțin ecuația de control m

$$q_{n,m}^0 = \alpha_{nm} P_n^0 \prod_{\substack{t \in M(n) \\ t \neq m}} r_{t,n}^0$$

$$q_{n,m}^1 = \alpha_{nm} P_n^1 \prod_{\substack{t \in M(n) \\ t \neq m}} r_{t,n}^1 \quad (22)$$

end

calculul probabilităților pt. criteriul lui Bayes →

$$Pl_n^0 = \beta_n P_n^0 \prod_{t \in M(n)} r_{t,n}^0$$

- conțin și ecuația de control m

$$Pl_n^1 = \beta_n P_n^1 \prod_{t \in M(n)} r_{t,n}^1$$

end

- Factorii de scalare β_n sunt calculați cu o relație similară cu (20) pentru valorile expresiilor din ultimele două rânduri ale (22).

- Metoda descrisă mai sus actualizează valorile tuturor nodurilor de control în paralel („flooding schedule”) într-o etapă a iterației și valorile tuturor nodurilor de bit în paralel în a doua etapă a unei iterații. Studiile prezentate în literatură, vezi și referința menționată, arată că pentru codurile de tip „quasi-cyclic” decodarea structurată („layered decoding”) permite o procesare mai rapidă. Implementarea de tip „layered decoding” implică actualizarea succesivă a fiecărui nod de control sau de bit.

Pasul 3 Decizia biților decodați, decizia privind corectitudinea biților decodați și decizia de terminare a decodării

- Valoarea logică a fiecărui bit al cuvântului de cod se determină cu o formă a criteriului lui Bayes, $Pl_n^1 > 0.5$, atunci bitul decodat este 1, altfel bitul decodat este 0.

- Corectitudinea biților decodați se determină prin verificarea ecuației $Hv^T = 0$.

- Această înmulțire dintre o matrice și un vector poate fi realizată prin metoda descrisă de relația (11)

- Decizia de terminare a decodării depinde de valoarea sindromului și de numărul de iterații efectuate:

• Dacă sindromul este nul, decodarea este încheiată, iar biții decodati sunt livrați către straturile superioare.

• Dacă sindromul nu este nul, $Hv^T \neq 0$, se efectuează o nouă iterație, adică se transmit noile mesaje de la nodurile de bit spre nodurile de control în care, pe baza acestor mesaje, se calculează noile mesaje spre nodurile de bit; cu aceste mesaje se determină noile probabilități *aposteriori* Pl_n^0, Pl_n^1 , numai dacă numărul de iterații efectuate nu a atins numărul maxim permis I ; I este un parametru al algoritmului MP.

- Conform considerațiilor de mai sus, următorul pas al algoritmului este exprimat de (23).

for $n = 0, \dots, N - 1$

daca $Pl_n^1 \geq 0.5$ atunci $x_n = 1$

altfel $x_n = 0$

end

Sind = Hv^T

daca Sind = 0 decodare terminata

altfel daca $i = I$ decodare terminata

altfel mergi la pasul 1

- Acest algoritm iterativ continuă până când se găsește un cuvânt de cod valid (cu sindrom nul) sau până când numărul de iterații atinge o valoare I , pentru care graful de decodare va deveni un arbore cu I nivele, vezi figurile 4 și 5, caz în care se consideră că decodarea a eșuat -fie se livrează așa, fie se șterge (erasure)
- Trebuie însă menționat că obținerea unui cuvânt de cod valid nu este echivalentă cu obținerea cuvântului de cod corect, cel transmis.
- În figurile 4 și 5 sunt prezentate grafurile vecinilor bitului c_1 după una, și respectiv, două iterații, pentru codul definit în relația (14)

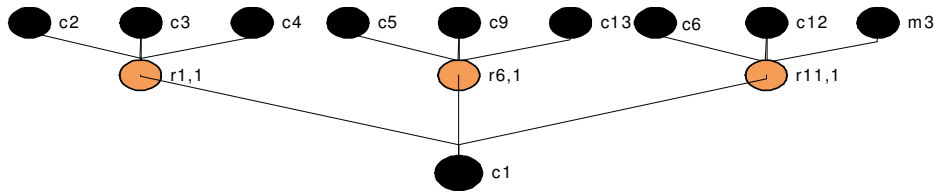


Figura 4 Graful vecinilor nodului de bit c_1 după prima iterație (de jos în sus)

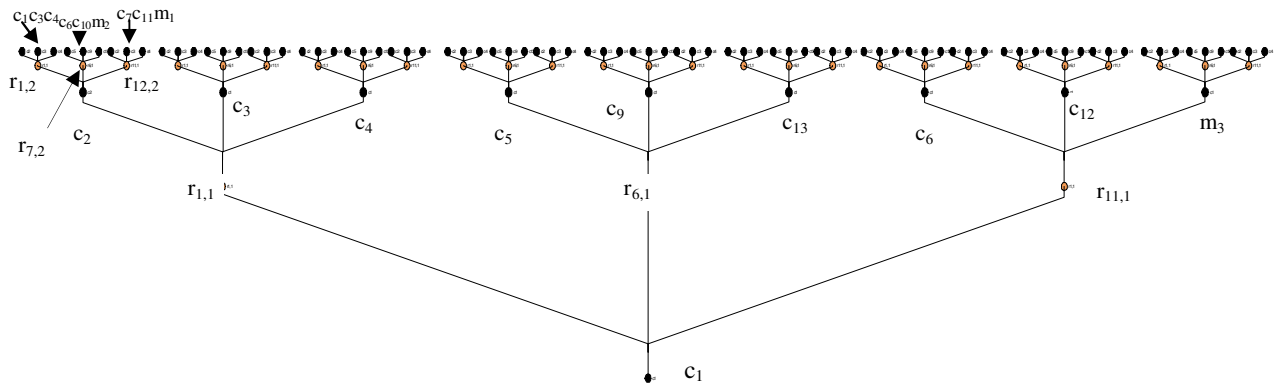


Figura 5 Graful vecinilor nodului de bit c_1 după a doua iterație (de jos în sus)

- Generalizând reprezentarea din fig.5, pentru ca toți vecinii de gradul L ai unui nod de bit să fie independenți (să apară o singură dată în graful vecinilor după iterația L), lungimea N a cuvântului de cod ar trebui să fie:

$$N \geq [(k-1) \cdot j]^L \quad (24)$$

- $c_1 + c_2 + c_3 + c_4 = 0$ (1)
- $c_4 + c_6 + c_7 + c_8 = 0$ (2)
- $c_9 + c_{10} + c_{11} + c_{12} = 0$ (3)
- $c_{13} + c_{14} + c_{15} + m_1 = 0$ (4)
- $m_2 + m_3 + m_4 + m_5 = 0$ (5)
- $c_1 + c_5 + c_9 + c_{13} = 0$ (6)
- $c_2 + c_6 + c_{10} + m_2 = 0$ (7)
- $c_3 + c_7 + c_{14} + m_3 = 0$ (8)
- $c_4 + c_{11} + c_{15} + m_4 = 0$ (9)
- $c_8 + c_{12} + m_1 + m_5 = 0$ (10)
- $c_1 + c_6 + c_{12} + m_3 = 0$ (11)
- $c_2 + c_7 + c_{11} + m_1 = 0$ (12)
- $c_3 + c_8 + c_{13} + m_4 = 0$ (13)
- $c_4 + c_9 + c_{14} + m_2 = 0$ (14)
- $c_5 + c_{10} + c_{15} + m_5 = 0$ (15)

- Deoarece lungimea cuvintelor de cod în situații reale nu respectă (24), subgraful vecinilor nodului de bit n rezultat în urma decodării iterative, după un număr de I iterații, nu va fi un arbore fără bucle.
- Aceasta înseamnă că mesajele de bit care contribuie la determinarea anumitor mesaje de control nu vor mai fi independente, deci ecuația (16) nu mai este validă (pt. că biții nu mai sunt independenți), iar valorile probabilităților $q_{n,m}^0, q_{n,m}^1$ nu se vor mai modifica (semnificativ).
- Considerând ecuațiile de control (14) ale codului definit de (12) și (13), în figura 6 (care reprezintă tot graful vecinilor bitului c_1) este figurată o buclă de 4 cicli, definită de:

$$c_1 \rightarrow r_{11} \rightarrow c_6 \rightarrow r_2 \rightarrow c_8 \rightarrow r_{13} \rightarrow c_{13} \rightarrow r_6 \rightarrow c_1 - 8 \text{ pași} - 4 \text{ cicli}$$

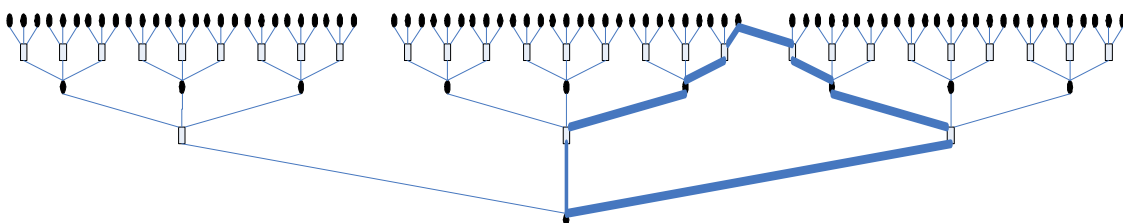
Alte bucle în graful Tanner:

$$c_1 \rightarrow r_1 \rightarrow c_2 \rightarrow r_7 \rightarrow c_6 \rightarrow r_{11} \rightarrow c_1 - 6 \text{ pași} - 3 \text{ cicli în graful Tanner}$$

$$c_1 \rightarrow r_1 \rightarrow c_3 \rightarrow r_8 \rightarrow m_3 \rightarrow r_{11} \rightarrow c_1 - 6 \text{ pași} - 3 \text{ cicli în graful Tanner}$$

(14)

Fig.6 ↓ Buclă în subgraful vecinilor; pentru indecșii nodurilor vezi fig. 5 și (14)



- Un parametru care afectează semnificativ capacitatea de corecție a codului este diametrul minim (numărul de tranziții) al buclelor din graful Tanner, numit „girth”, și numărul buclelor cu diametru minim, fiecare buclă corespunzând unui bit al cuvântului de cod.
- Trebuie reținut că acest algoritm nu încearcă să găsească cel mai apropiat cuvânt de cod, față de secvența recepționată, ci încearcă să corecteze fiecare bit al secvenței recepționate, folosind informațiile oferite de ceilalți biți cu care bitul dat intră în ecuațiile de control. Sindromul este folosit pentru verificarea corectitudinii cuvântului decodat; deci acest tip de coduri are CRC inclus în structura sa.
- Datorită acestui fapt, numărul de biți eronați după o decodare nereușită este mai mic decât numărul biților eronați înaintea decodării (spre deosebire de codurile bloc ciclice sau de cele convoluționale), dacă cuvântul de cod este recepționat în integralitatea sa pe un canal cu parametrii aproximativ constanți.
- Numărul maxim de iterații este un parametru care se stabilește în funcție de durata permisă pentru decodare de către aplicația în care este utilizat codul.
- Varianta în probabilități *a posteriori* a algoritmului MP are dezavantajul că necesită un număr foarte mare de înmulțiri. Ea folosește pentru decizie o formă echivalentă a raportului de plauzibilitate, adică raportul probabilităților *a priori* ca bitul n să fi fost „1” sau „0”, dacă s-a recepționat fazorul y :

$$LR_n = \frac{P_1^n}{P_0^n} > 1 \text{ bitul } n \text{ este } 1 \quad (25)$$

- Pentru reducerea resurselor de calcul necesare s-a dezvoltat o variantă a acestui algoritm care utilizează logaritmul raportului de plauzibilitate, Log-Likelihood ratio (LLR), al fiecărui bit ca variabilă în algoritm.
- Ea se bazează pe exprimarea logaritmului raportului de plauzibilitate sub forma (25’):

$$LLR = \ln \frac{P_0}{P_1} \Rightarrow e^{LLR} = \frac{P_0}{P_1} \Rightarrow \tanh\left(\frac{LLR}{2}\right) = \frac{e^{LLR} - 1}{e^{LLR} + 1} = P_0 - P_1 \text{ - comparată cu } 0 \quad (25')$$

- Studiul acestei variante depășește cadrul cursului de față.
- În general codurile LDPC random cu cuvinte de cod de lungime mai mare (la aceeași rată) au bucle cu girth-uri minime mai mari în graful Tanner asociat, ceea ce scade probabilitatea de eroare după decodare.
- Codurile LDPC de tip array au valoarea girth-ului minim independentă de lungimea cuvântului de cod
- Lungimea cuvântului de cod trebuie aleasă și în funcție de timpul de coerență al canalului și de volumul resurselor timp-frecvență (simboluri QAM) ce pot fi alocate pentru transmiterea sa.

1.6. Demaparea soft

- Deoarece semnalele demodate cu metoda QAM reprezintă coordonatele (afectate de canal) ale unui fazor pe care au fost mapați p biți, probabilitățile *a posteriori* ca dacă s-a recepționat fazorul f_r , de coordonate I_r și Q_r , bitul j să aibă valoarea „1” sau „0”, se obțin prin operația de demapare soft descrisă de:

$$P(b_j = 1) = \frac{\sum_{k=1}^{2^p} \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\left(-\frac{(I_r - I_k)^2 + (Q_r - Q_k)^2}{2\sigma^2}\right) \cdot b_{jk}}{\sum_{k=1}^{2^p} \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\left(-\frac{(I_r - I_k)^2 + (Q_r - Q_k)^2}{2\sigma^2}\right)} \quad (26)$$

$$P(b_j = 0) = \frac{\sum_{k=1}^{2^p} \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\left(-\frac{(I_r - I_k)^2 + (Q_r - Q_k)^2}{2\sigma^2}\right) \cdot \overline{b_{jk}}}{\sum_{k=1}^{2^p} \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\left(-\frac{(I_r - I_k)^2 + (Q_r - Q_k)^2}{2\sigma^2}\right)}$$

unde I_k și Q_k reprezintă coordonatele celor $N = 2^p$ fazori ai constelației, b_{jk} valoarea logică a bitului cu index j din grupul de biți mapat pe fazorul k , $k = 0, \dots, N-1$, iar b_j bitul cu index j din grupul de biți 'demapat' de pe fazorul recepționat (I_r, Q_r), vezi figura 7.

- Trebuie menționat faptul că probabilitățile de eronare ale biților depind de poziția bitului pe simbol dacă maparea este de tip Gray; probabilitățile de eronare ale biților cei mai semnificativi fiind mai mici decât cele ale biților mai puțin semnificativi.

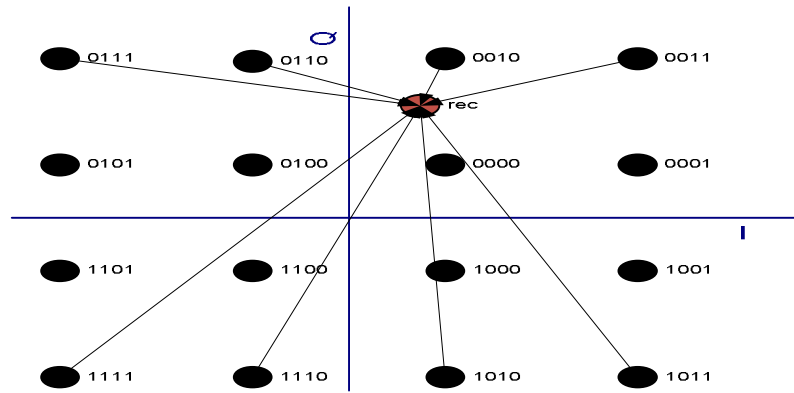


Figura 7 Fazorii din care poate să provină fazorul recepționat dacă al 3-lea bit este „1”

- Aceste probabilități reprezintă P_j^1 și P_j^0 care sunt introduse ca variabile de intrare pentru fiecare bit în algoritmul MP
- Datorită erorilor de prelucrare, în cazurile practice suma celor două probabilități nu va fi egală cu 1, și aceea valorile obținute din calcule trebuie normate la suma lor, așa cum se face și în ecuațiile (20) și (22).
- Valorile probabilităților *aposteriori* depind de valoarea SNR (și implicit a puterii zgomotului σ_2).
- În standardul 5G NR se prevede ca biții cuvântului de cod să fie introduși într-un interleaver înainte de maparea pe simbolurile QAM, pentru a poziționa biții informaționali pe pozițiile cele mai semnificative ale simbolului.