

CAPITOLUL IV

CODURI CORECTOARE DE ERORI

IV.1 Introducere

Un **cod corector de erori utilizat pentru transmisia datelor** transformă un canal discret zgomotos într-un canal sigur, de încredere adică indiferent de interferențele și distorsiunile de pe canal, secvența decodată este corectă. Aceste coduri sunt proiectate pentru canale zgomotoase fără constrângeri impuse secvenței transmise. Canalul de transmisii (CT) cu constrângeri poate fi transformat într-un CT fără constrângeri prin folosirea unui codor (decodor) de translare a datelor. Aplicațiile acestor tehnici de codare sunt diverse: comunicații mobile, comunicații prin satelit, modemuri de transmisie simultană voce-date, transmisii de date prin sistemele TV și radio existente etc.

Tehnicile de corecție a erorilor de transmisie în sistemele de comunicații digitale au prezentat interes practic abia din anii '80 când au fost introduse în comunicațiile prin satelit precum și în sistemele de stocare a datelor cu densitate ridicată, pe CD (*Compact Disc*). Strategia de proiectare a acestor coduri s-a modificat continuu. Codurile de tip 'bloc' au evoluat de la forma binară la varianta multibit, ceea ce a permis reducerea considerabilă a energiei necesare de transmisie precum și la simplificarea echipamentelor de transmisie-recepție.

În unele sisteme de comunicații, pentru scăderea complexității și a costului echipamentelor, se preferă realizarea doar a detecției erorilor de transmisie fără corecția lor, având ca urmare cererea de retransmisie a cadrelor eronate.

În cazul codului ASCII cu exprimare pe 7 biți a caracterelor se procedează la adăugarea unui așa-numit **bit de paritate** prin aplicarea unei anumite convenții:

- ◆ pentru **paritate pară** (*even-parity*) se calculează valoarea bitului excedentar astfel încât ponderea octetului transmis să fie de valoare pară (exemplu: 1 0011100);

- ◆ pentru **paritate impară** (*odd-parity*) se calculează valoarea noului bit astfel încât ponderea octetului transmis să fie de valoare pară (exemplu: 0 1110101);

- ◆ pentru **paritate de tip '1'** (*mark-parity*) se introduce un bit '1' în octetul transmis;

- ◆ pentru **paritate de tip '0'** (*space-parity*) se adaugă un bit '0' la sfârșitul octetului transmis.

În toate situațiile, receptorul verifică valoarea bitului de paritate în vederea depistării eventualelor erori de transmisie.

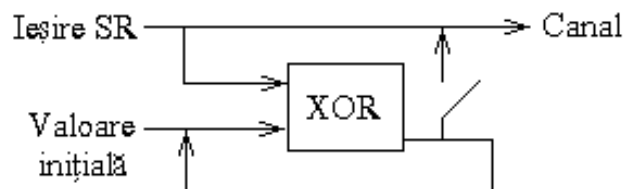


Fig. IV.1 Buclă de generare a bitului de paritate

Generarea bitului de paritate se face conform metodei adoptate:

- ◆ paritate pară: se sumează modulo-2 biții transmiși (Fig.IV.1) (folosind valoarea inițială '0');
- ◆ paritate impară: se neagă suma modulo-2 a biților din registrul de transmisie (SR - *Shift Register*) aplicând o valoare inițială '1'.

Întrucât aceste metode se limitează doar la detecția unor erori simple de transmisie, se poate realiza o verificare complexă a parității semnalului, lucrându-se pe matrici de biți (7×7 biți) cu paritate verificată simultan pe linii și pe coloane, cu posibilitatea determinării pozițiilor unor erori.

Exemplu: Fie șirul de date:

1001001.1111101.1110111.0000101.0010111.0000100.1100001.1010001

Se transformă vectorul de date corespunzător primilor 49 de biți într-o matrice pătratică, bidimensională și se introduc biții de paritate pe linia a 8-a și pe coloana a 8-a, pe principiul parității impare:

$$M = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Șirul binar transmis se obține prin citirea liniilor matricii codate M:

10010010.11111011.11101111.00001011.00101111.00001000.11000010.
10010111.

Dacă se recepționează eronat biții subliniați, matricea corespunzătoare secvenței recepționate este:

$$R = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad v_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad v_2 = [1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1]$$

Recalculând ponderile fiecărei linii și coloane, se obțin vectorii v_1 , v_2 . Se observă apariția erorilor pe liniile 1; 3; 4; 5 și pe coloanele 2; 4; 5; 7 (numerotarea s-a făcut de la 1 la 8). Biții determinați de intersecțiile acestora pot fi eronați.

Pentru o rată a erorilor uzuală de 10^{-6} și amplasare aleatoare a acestora, este posibil ca cel mult un bit din cei 49 ai matricii să fie eronat. Poziția acestuia se determină simplu prin intersectarea liniei cu coloana care rezultă prin recalcularea biților de paritate. O eroare pe al 10-lea bit din șir, va conduce la modificarea parității pe linia a 2-a și pe coloana a 3-a a matricii de date fiind astfel posibilă corecția sa.

Totuși sistemele complexe de comunicații digitale includ circuite de corecție a erorilor folosindu-se: coduri Hamming; coduri Reed-Solomon; coduri convoluționale; coduri ciclice; coduri TURBO, care vor fi studiate în acest capitol.

Codurile Hamming binare sunt utile pentru transmisiile afectate de zgomote aditive cu distribuție non-gaussiană dar sunt vulnerabile în cazul apariției fenomenului de fading

Sunt studiate în acest capitol performanțele codurilor Reed-Solomon definite în câmpuri Galois de diferite dimensiuni, prezentându-se codurile optime, adică cele care satisfac un compromis între creșterea debitului transmis și numărul de erori corectabile. Algoritmii de codare/decodare RS pot fi implementați cu ajutorul procesoarelor specializate pentru câmpuri Galois și pot fi executați în timp real. Aplicațiile acestor coduri cuprind transmisiile digitale cu fading și mai puțin cele cu perturbații aditive aleatoare.

Codurile convoluționale au fost analizate în variantă binară specificându-se codurile optime cu cel mult patru celule de memorie. A fost impusă această limită pentru a permite procesarea datelor în timp real printr-un număr limitat de operații aritmetice. În

variantă binară, aceste coduri sunt deosebit de performante pentru transmisii de date cu viteze mici și medii, afectate de zgomote aditive de tip gaussian. Limitarea capacității de corecție, determinată de creșterea excesivă a numărului de operații efectuate în decodor, impune proiectarea acestor coduri în câmp Galois, asemenea codurilor RS.

Sunt studiate codurile convoluționale multiple optime pentru câteva câmpuri algebrice și performanțele acestora. Evident capacitatea de corecție a erorilor a acestor coduri este mult mai mare față de codurile binare, pentru același număr de operații efectuate la decodare. Acest lucru permite utilizarea lor și pentru transmisii cu fading.

Metoda de decizie segmentată în aplicarea algoritmului Viterbi de decodare, pe blocuri de lungime variabilă, este studiată în cazul binar dar și pentru aplicarea pe simboluri de tip multibit.

În final, sunt analizate tehnicile combinate de codare, corectoare de erori, inclusiv cele TURBO care procesează semnalul întreșut, în matrici de date. Construcția codurilor TURBO poate fi realizată în câmp Galois, astfel încât datele să fie prelucrate direct pe structura de simboluri multibit. Sunt analizate câteva combinații posibile de coduri RS și convoluționale, pe 2, 3 și respectiv 4 biți. Câștigul mediu de codare pentru un simbol este de circa 8 dB. Creșterea dimensiunilor câmpului de definiție determină viteze de transmisie mai mari pentru aceiași parametri de sistem (lățime de bandă, rată de erori, raport semnal/zgomot).

Tehnicile de corecție a erorilor permit îmbunătățirea performanțelor modulatorilor digitale pentru diferite condiții de transmisie ceea ce impune analiza comună a acestora.

IV.2 Câmpuri Galois

Definiție: Un **câmp Galois** (GF - *Galois Field*) este prin definiție un câmp algebric finit în care se introduc două operații interne, adunarea și înmulțirea, pentru care există **element neutru** (0) respectiv **element unitate** (1):

$$\forall a \in GF, a + 0 = a \quad (IV.1)$$

$$a \cdot 1 = a \quad (IV.2)$$

Operațiile de scădere și împărțire se definesc implicit pe baza **elementului opus**

(-a) și a **elementului invers** (1/a, a nenul):

$$\forall a \in GF, \text{daca } a + b = 0 \text{ atunci } b = -a \quad (IV.3)$$

$$\forall a \in GF \setminus \{0\}, \text{daca } ab = 1 \text{ atunci } b = 1/a \quad (IV.4)$$

Operațiile de adunare și de înmulțire definite în GF au proprietăți de liniaritate, comutativitate, asociativitate și distributivitate asemenea celor definite în câmpul numerelor complexe.

Pentru $\forall m \in \mathbb{Z}^*, \exists GF(2^m)$ ale cărei elemente pot fi reprezentate pe m biți.

Adunarea: În GF, sumarea a două elemente se realizează prin sumarea modulo-2 bit cu bit.

Exemplu: $GF(16)$, $m = 4$, $1101 + 1110 = 0011$ adică $13 + 14 = 3$.

Înmulțirea: Produsul a două elemente din GF trebuie să fie element al câmpului. Este necesară deci introducerea unei reguli de evitare a depășirii (*overflow-rule*) în termeni polinomiali.

Fie $p(x)$ un polinom de grad m , ireductibil în $GF(2^m)$, având coeficienți binari (Tabel IV.1).

Tabelul IV.1

Polinoame ireductibile utilizate în $GF(2^m)$

Dimensiunea GF m	Polinoame ireductibile p(x)
2	x^2+x+1
3	$x^3+x+1; x^3+x^2+1$
4	$x^4+x+1; x^4+x^3+1$
5	$x^5+x^2+1; x^5+x^3+1$
6	$x^6+x+1; x^6+x^2+1; x^6+x^3+1; x^6+x^4+1; x^6+x^5+1$
7	$x^7+x+1; x^7+x^3+1; x^7+x^4+1; x^7+x^6+1$
8	$x^8+x^2+1; x^8+x^3+1; x^8+x^5+1; x^8+x^6+1$

Se asociază fiecărui element al câmpului trei forme de reprezentare: zecimală, binară și polinomială:

$$a \Leftrightarrow (a_0, a_1, \dots, a_{m-1}) \rightarrow a(x) = \sum_{i=0}^{m-1} a_i x^i \quad (IV.5)$$

Definim produsul a două elemente a și b din câmp, prin produsul:

$$c(x) = a(x)b(x) \bmod(p(x)) \quad (IV.6)$$

unde calculul coeficienților se realizează cu operații de întârziere și de sumare modulo-2.

Prin multiplicarea modulo- $p(x)$ a polinoamelor, se evită depășirea, adică:
 $\text{grad}(c(x)) \leq m - 1$, deci $c \in GF(2^m)$.

Exemplu: În $GF(16)$ cu $p(x) = x^4 + x + 1$, pentru $a = 7$ și $b = 11$, $c = ab = 1$.

Observații:

❶ Dacă $b_1 = -a$, $b_2 = 1/a$ atunci scăderea și împărțirea elementelor din GF se efectuează conform relațiilor:

$$c - a = c + b_1 \quad (\text{IV.7})$$

$$\frac{c}{a} = c \cdot b_2 \quad (\text{IV.8})$$

❷ Existența și unicitatea elementului invers sunt asigurate numai dacă polinomul $p(x)$, utilizat pentru definirea operației de multiplicare în GF, este ireductibil.

❸ Regulele de operare în GF se implementează ușor ca algoritmi.

În prezent există procesoare digitale de semnal pentru operare în câmp Galois. Acestea sunt utilizate pentru codarea și decodarea datelor pe baza unor reguli definite în câmp numeric finit (GF). În câmp Galois pot fi construite atât coduri de tip 'bloc', cât și coduri de tip 'arbore' (convoluționale).

❹ Comparativ cu câmpul numerelor complexe, în GF se păstrează:

◆ metodele de rezolvare a sistemelor liniare de ecuații;

◆ regulile de calcul matricial;

◆ transformata Fourier discretă (DFT) care există în $GF(N)$ numai pentru anumite dimensiuni (n) ale vectorilor de semnal și anume numai dacă n divide $N-1$.

Definiție: Transformata Fourier discretă directă și inversă se calculează în GF cu relațiile :

$$\text{DFT} : C_j = \sum_{i=0}^{n-1} c_i w^{ij}, j = \overline{0, n-1} \quad (\text{IV.9})$$

$$\text{IDFT} : c_i = \sum_{j=0}^{n-1} C_j w^{-ij}, i = \overline{0, n-1} \quad (\text{IV.10})$$

unde w este o rădăcină a unității de ordin n în $GF(N)$ dacă n divide N :

$$w^n = 1 \quad (\text{IV.11})$$

Spre deosebire de relația de calcul în câmpul numerelor complexe a transformatei Fourier discrete inverse, în GF nu apare divizarea prin n datorită sumării modulo-2 a coeficienților polinoamelor.

În $GF(2^m)$ sunt verificate următoarele condiții:

$$\text{P1. } y^{2^m-1} = 1, \forall y \in GF \setminus \{0\} \quad (\text{IV.12})$$

$$P2. y^{2^m} = y, \forall y \in GF. \quad (IV.13)$$

Circuitele de operare în GF (sumator, multiplicator, divizor și inversor) pot fi implementate ca circuite logice combinaționale, cu porți logice sau cu memorii cu viteză mare de acces. Avantajul acestor metode îl constituie timpul redus de prelucrare. Ele sunt utile în implementarea circuitelor de codare în GF care să lucreze în timp real. Aceleași operații pot fi implementate ca algoritmi folosind procesoare digitale de semnal.

Exemplu: Să analizăm câmpul Galois cu 8 elemente exprimabile:

$$\text{zecimal: } GF(8) = \{0; 1; 2; 3; 4; 5; 6; 7\}$$

$$\text{binar: } GF(8) = \{000; 001; 010; 011; 100; 101; 110; 111\}$$

$$\text{polinomial: } GF(8) = \{0; 1; D; D+1; D^2; D^2+1; D^2+D; D^2+D+1\}.$$

D - operatorul de întârziere cu un bit (*delay*).

Se alege pentru definirea operației de înmulțire în acest câmp, polinomul:

$$p(D) = D^3 + D + 1$$

Să efectuăm câteva calcule în acest câmp:

$$6 + 3 = 5 \Leftrightarrow 110 + 011 = 101$$

$$5 - 2 = 5 + 2 = 7 \Leftrightarrow 101 + 010 = 111$$

$$7 \cdot 3 = 2 \Leftrightarrow (D^2 + D + 1)(D + 1) \text{ modulo } [p(D)] = (D^3 + 1) \text{ modulo } [p(D)] = \\ = (D^3 + D + 1 + D) \text{ modulo } [p(D)] = D$$

$$2 \cdot 5 = 1 \Leftrightarrow D(D^2 + 1) \text{ modulo } [p(D)] = 1 \Rightarrow 2^{-1} = 5$$

$$\frac{7}{2} = 7 \cdot 5 = 6$$

Observație: Este avantajos modul de calcul matricial al transformatorilor DFT și

IDFT n-dimensionale în GF:

$$\bar{C} = n - DFT\{\bar{c}\} = \Omega_n \cdot \bar{c} \quad (IV.14)$$

$$\bar{c} = n - IDFT\{\bar{C}\} = \Omega_n^{-1} \cdot \bar{C} \quad (IV.15)$$

unde \mathbf{c} și \mathbf{C} sunt vectori-coloană cu n elemente aparținând câmpului, iar relațiile

de definiție a matricilor pătratice n*n utilizate sunt:

$$\Omega_n = [\omega_{ij}^{i \cdot j \text{ modulo } n}]_{i,j=0;n-1} \quad (IV.16)$$

$$\Omega_n^{-1} = [\omega_{ij}^{-i \cdot j \text{ modulo } n}]_{i,j=0;n-1} \quad (IV.17)$$

(ω este în acest caz rădăcină de ordin n a unității în GF considerat.)

Operațiile între coeficienți se efectuează în GF.

Exemplu: Continuând exemplul anterior, să determinăm rădăcina de ordin 7 a unității în GF(8) și să calculăm DFT pentru vectorul (1 0 2 4 0 1 0).

Calculăm puterile lui 2 în GF(8):

$$a = 2; 2 \cdot 2 \rightarrow D^2 \rightarrow 100 \rightarrow 2^2 = 4;$$

$$2 \cdot 4 \rightarrow D^3 \text{ modulo } (D^3 + D + 1) = D + 1 \rightarrow 011 \rightarrow 2^3 = 3;$$

$$2 \cdot 3 \rightarrow D(D+1) = D^2 + D \rightarrow 110 \rightarrow 2^4 = 6;$$

$$2 \cdot 6 \rightarrow D(D^2 + D) \text{ modulo } [p(D)] = (D^3 + D^2) \text{ modulo } [p(D)] = D^2 + D + 1 \\ \rightarrow 111 \rightarrow 2^5 = 7;$$

$$2 \cdot 7 \rightarrow D(D^2 + D + 1) \text{ modulo } [p(D)] = D^2 + 1 \rightarrow 101 \rightarrow 2^6 = 5;$$

$$2 \cdot 5 \rightarrow D(D^2 + 1) \text{ modulo } [p(D)] = 1 \rightarrow 2^7 = 1.$$

Deci 2 este element de ordin 7 în GF(8) cu p(D) precizat mai sus.

În general, într-un GF, există mai multe elemente având același ordin ceea ce conduce la obținerea unor coduri aparent diferite dar echivalente ca performanțe într-un anumit GF.

Pentru vectorul de date considerat, obținem:

$$\bar{C} = 7 - DFT\{\bar{c}\} = \Omega_7 \bar{c} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^1 & \omega^3 & \omega^5 \\ 1 & \omega^3 & \omega^6 & \omega^2 & \omega^5 & \omega^1 & \omega^4 \\ 1 & \omega^4 & \omega^1 & \omega^5 & \omega^2 & \omega^6 & \omega^3 \\ 1 & \omega^5 & \omega^3 & \omega^1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \\ 4 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \\ = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 & 6 & 7 & 5 \\ 1 & 4 & 6 & 5 & 2 & 3 & 7 \\ 1 & 3 & 5 & 4 & 7 & 2 & 6 \\ 1 & 6 & 2 & 7 & 4 & 5 & 3 \\ 1 & 7 & 3 & 2 & 5 & 6 & 4 \\ 1 & 5 & 7 & 6 & 3 & 4 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \\ 4 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 1 \\ 4 \\ 1 \\ 2 \\ 5 \end{bmatrix}$$

IV.3 Coduri-bloc corectoare de erori

Un cod-bloc corector de erori protejează secvența de simboluri transmisă față de zgomotele și erorile aleatoare introduse de canalul de comunicații. La fiecare **k** simboluri (biți) care se transmit, codul corector de erori adaugă (n-k) **simboluri (biți) de paritate**. Lungimea cuvântului de cod este de **n** simboluri (biți).

Pentru aceste coduri se calculează **rata de codare**:

$$R = k/n \quad (IV.18)$$

Prin introducerea simbolurilor de paritate, se urmărește creșterea distanței Hamming minime dintre cuvintele de cod transmise întrucât circuitul de decizie din receptor lucrează pe principiul distanței Hamming minime dintre cuvântul recepționat și cel estimat.

Dacă $d_{H \min} = 2t+1 = n-k$, atunci raza sferelor de decizie din planul semnalelor este de t biți, sferile nu sunt tangente și se pot corecta maxim t erori dintr-o secvență de lungime n .

Dacă $d_{H \min} = 2t = n-k$ atunci raza sferelor de decizie din planul semnalelor este de t biți, sferile sunt tangente, se pot corecta maxim $(t-1)$ erori și se detectează prezența a t erori într-un bloc de n simboluri.

Cele mai cunoscute coduri-bloc corectoare de erori sunt :

- ◆ codurile Hamming, organizate pe biți (*bit-organized codes*);
- ◆ codurile Reed-Solomon organizate pe simboluri (*byte-organized codes*).

IV.3.1 Coduri Hamming $H(n; k)$

Sunt coduri liniare descrise prin tabele de codare care specifică modul de asociere a cuvintelor de intrare de k biți cu cele de cod de n biți. Introducerea biților de paritate, respectiv creșterea redundanței semnalului, are rolul de a mări distanța Hamming minimă și medie dintre cuvinte astfel încât să fie posibilă detecția și corecția unor erori care apar la transmisia datelor pe CT zgomotos. Prin permutarea cuvintelor de cod se obțin coduri echivalente, cu aceeași valoare a distanței Hamming minime între cuvintele de cod. Se utilizează mai ales codurile **sistematice** pentru care cuvântul de intrare este parte componentă a cuvântului de cod, fie la începutul, fie la sfârșitul acestuia. În acest mod, nu mai este necesar un circuit de decodare în receptor, biții de date fiind culeși direct din semnalul recepționat. Se utilizează în acest caz doar blocul de detecție și corecție a erorilor.

În general, pentru orice număr m întreg, pozitiv, nenul, există un cod Hamming de dimensiuni: $H(2^m - 1, 2^m - 1 - m)$. Se preferă folosirea codurilor cu parametru m impar întrucât în acest caz sferile de decizie sunt disjuncte și se pot corecta $t = (m-1)/2$ erori. Codurile cu parametru m par sunt utile în sistemele ARQ (*Automatic Request*).

Construcția codului poate fi realizată **matricial** într-un câmp numeric finit, câmp Galois binar, $GF(2)$.

Se scrie inițial **matricea de paritate** H de dimensiuni $m \times (2^m - 1)$, având pe coloane toate combinațiile binare nenule de m biți, eventual în ordine crescătoare a

ponderilor. Pentru proiectarea unui cod sistematic, primele m coloane ale matricii H trebuie să formeze matricea unitate (identitate) I_m :

$$H = [I_m \ P] \quad (\text{IV.19})$$

Matricea generatoare G a codului, determinată din condiția: $HG^T = 0$, (M^T semnifică transpusa matricii M), va fi de forma:

$$G = [P^T I_k] \quad (\text{IV.20})$$

Vectorul codat \mathbf{c} se obține din produsul vectorului-linie al datelor \mathbf{a} (de k biți) și matricea generatoare:

$$\bar{\mathbf{c}} = \mathbf{a}G \quad (\text{IV.21})$$

Semnalul recepționat este divizat în blocuri de lungime n urmând ca fiecare dintre acestea să fie corectat folosind **vectorul sindrom** calculat cu relația:

$$\bar{\mathbf{s}} = H\bar{\mathbf{r}}^T = H(\mathbf{c}^T + \mathbf{e}^T) = HG^T \mathbf{a}^T + H\mathbf{e}^T = H\mathbf{e}^T \quad (\text{IV.22})$$

(\mathbf{r} - vectorul recepționat; \mathbf{e} - vectorul de eroare).

Dacă vectorul transmis nu a fost afectat de erori (vector de erori nul) atunci rezultă vectorul sindrom nul. În cazul existenței unor erori de transmisie vectorul sindrom depinde doar de vectorul erorilor (nenul pe pozițiile corespunzătoare erorilor) fiind independent de vectorul de date.

Prin codarea tuturor combinațiilor binare de k biți, de intrare în codor, se obține un tabel de codare cu dimensiuni rezonabile în comparație cu tabelele care s-ar obține prin codarea Reed-Solomon.

Se poate realiza construcția codurilor Hamming și în varianta polinomială când se asociază fiecărui cuvânt, de intrare sau de cod, nu un vector, ci un polinom în variabila D (operatorul de întârziere cu un bit) iar în locul matricii generatoare se folosește un polinom generator. Metoda de proiectare matricială și cea polinomială sunt echivalente. Indiferent de metoda folosită, operațiile dintre coeficienți se efectuează în $GF(2)$.

Exemplu: Să proiectăm codul Hamming $H(7;4)$ în varianta sistematică.

Este un cod cu rata de codare $4/7$, corector de o eroare într-un bloc de 7 biți.

Matricile de paritate și cea generatoare sunt:

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

Codarea combinațiilor posibile de 4 biți se face aplicând relația (IV.21).

Pentru intrare $\mathbf{a} = (0\ 1\ 0\ 0)$ obținem vectorul codat $\mathbf{c} = (1\ 0\ 1\ 0\ 1\ 0\ 0)$ adică primii 3 biți sunt cei de paritate iar ultimii 4 sunt chiar biții datelor. Similar se determină biții de paritate pentru orice altă combinație de intrare. Codarea, respectiv decodarea se vor face pe baza tabelului IV.2.

Dacă se recepționează vectorul $\mathbf{r} = (1\ 0\ 1\ 1\ 1\ 0\ 0)$ cu o eroare de transmisie, vectorul sindrom este $\mathbf{s} = (0\ 1\ 1)$ și se observă că reprezintă coloana a patra din matricea de paritate. Deci vectorul de eroare este $\mathbf{e} = (0\ 0\ 0\ 1\ 0\ 0\ 0)$.

Vectorul corectat se obține prin sumarea în GF(2) a vectorului recepționat cu cel de eroare: $\mathbf{v} = (1\ 0\ 1\ 0\ 1\ 0\ 0)$. Datele transmise sunt culese de pe ultimele patru poziții: 0 1 0 0.

Tabelul IV.2 Cod Hamming H(7;4)

Biți de paritate	Biți de date
000	0000
111	0001
110	0010
001	0011
101	0100
010	0101
011	0110
100	0111
011	1000
100	1001
101	1010
010	1011
110	1100
001	1101
000	1110
111	1111

IV.3.2 Coduri Reed-Solomon

Codurile Reed-Solomon (RS) sunt organizate pe blocuri de biți ceea ce face necesară construcția lor în câmp algebric finit.

Aceste coduri realizează corecția erorilor care afectează global simbolurile de date deci îmbunătățesc probabilitatea de eroare de simbol și implicit pe cea de bit. Deși au o complexitate mai mare în comparație cu codurile Hamming, codurile RS sunt mai performante și sunt utilizate pe scară largă în sistemele de comunicații digitale.

Codurile RS se definesc pe baza DFT din câmpul de definiție a codului. În cazul câmpului complex crește numărul de biți necesari exprimării elementelor în binar sau precizia de calcul scade și apar erori de depășire asemănător operației de cuantizare.

Definiție: Un cod **RS(n; n-2t)** corector de **t** erori este definit ca fiind setul tuturor vectorilor **c** de lungime **n** care au **2t** componente spectrale nule, respectiv **2t** elemente ale vectorului obținut prin aplicarea DFT vectorului de date sunt zero.

Se preferă ca nulurile spectrale să fie plasate grupat la începutul sau sfârșitul vectorului obținut prin transformata Fourier discretă pentru simplificarea procedurii de codare.

Codarea RS poate fi realizată fie în varianta matricială, pe baza calculului matricial al DFT și IDFT, cu vectori de intrare și de cod, fie în varianta polinomială cu polinoame asociate vectorilor de intrare, vectorului codat și cu un polinom generator.

Avantajul proiectării codurilor RS în câmp complex este acela că nu apar restricții în ce privește dimensiunile vectorilor pentru calculul DFT și IDFT așa cum se întâmplă în câmpurile Galois.

Indiferent de dimensiunile câmpului algebric în care se construiește codul, există trei metode de proiectare, una directă, matricială, plecând de la definiția codului, și două indirecte, polinomiale, lucrând în domeniul timp, pe baza unui polinom generator. Doar o singură metodă asigură construcția unui cod sistematic.

Petru codurile RS nu se scriu tabele de codare întrucât acestea ar avea dimensiuni mult prea mari.

Metodele de codare RS sunt următoarele:

A. Codarea în domeniul frecvență

Se impun valorile componentelor vectorului spectral și prin aplicarea transformatei Fourier discrete inverse (IDFT) se obține vectorul de cod. Vectorul spectral rezultă direct din vectorul de date prin completarea acestuia cu $2t$ zerouri (*zero-padding*):

$$C_j = \begin{cases} a_j, & j = \overline{0, n-2t-1} \\ 0, & j = \overline{n-2t, n-1} \end{cases} \quad (\text{IV.23})$$

unde a_j sunt componentele vectorului de intrare.

Vectorul de cod este:

$$\bar{c} = IDFT(\bar{C}) \quad (IV.24)$$

$$c_i = \sum_{j=0}^{n-1} C_j \omega_n^{-ij}, \quad i = \overline{0, n-1} \quad (IV.25)$$

unde ω_n este rădăcina de ordin n a unității din câmpul numeric complex sau în GF.

Observații: Această metodă nu permite obținerea unui cod sistematic. Toate operațiile între coeficienți sunt cele interne, de sumare și multiplicare, definite în câmpul în care se lucrează (complex sau Galois). De asemenea, DFT are particularități de aplicare în cazul construirii codului în GF, când se restricționează dimensiunile codului în funcție de existența unei rădăcini a unității de un anumit ordin.

B. Codarea în domeniul timp

Se asociază vectorului de intrare un polinom :

$$\bar{a} = (a_0, a_1, \dots, a_{n-2t-1}) \rightarrow a(x) = \sum_{i=0}^{n-2t-1} a_i x^i \quad (IV.26)$$

Cu x s-a notat *operatorul de întârziere cu un simbol*.

Se construiește polinomul generator al codului, cu coeficienți binari (0 sau 1), cu $2t$ zerouri: $\omega_n^{n-2t}, \omega_n^{n-2t+1}, \dots, \omega_n^{n-1}$. Zerourile astfel alese plasează nulurile spectrale pe ultimele poziții ale vectorului obținut prin DFT.

$$g(x) = (x - \omega_n^{n-2t})(x - \omega_n^{n-2t+1}) \dots (x - \omega_n^{n-1}) \quad (IV.27)$$

Polinomul de cod se obține simplu ca produs:

$$c(x) = g(x)a(x) .$$

Metoda conduce într-adevăr la obținerea unui cod RS deoarece spectrul vectorului codat are $2t$ zerouri:

$$C_j = \sum_{i=0}^{n-1} c_i \omega_n^{ij} = c(\omega_n^j) = g(\omega_n^j)a(\omega_n^j) = 0, \quad j = \overline{n-2t, n-1} \quad (IV.28)$$

Operațiile între coeficienți sunt cele din câmpul de definire a codului.

Simbolurile de date nu apar explicit în vectorul de cod, deci metoda nu este sistematică.

C. Metoda sistematică de codare RS

Se calculează polinomul de cod cu relația:

$$c(x) = x^{2t}a(x) + r(x) \quad (IV.29)$$

unde $r(x)$ este opusul restului împărțirii polinomului $x^{2t}a(x)$ la polinomul generator $g(x)$.

Toate operațiile între coeficienți se efectuează în câmpul algebric de definiție a codului.

Polinomul obținut este divizibil cu polinomul generator ceea ce asigură existența celor $2t$ nuluri spectrale.

Se observă că simbolurile de date apar neschimbate în vectorul de cod pe pozițiile cele mai semnificative, deci metoda este sistematică și permite o mai simplă implementare a decodului.

Teoremă : Un cod RS($n, n-2t$) poate corecta t erori de simbol.

Demonstrație:

Conform definiției codului RS există $2t$ nuluri spectrale pentru orice cuvânt de cod.

$$C_j = 0, j = \overline{n-2t, n-1}$$

Aplicând IDFT se obțin componentele vectorului de cod :

$$c_i = K \sum_{j=0}^{n-1} \omega_n^{-ij} C_j = KC(\omega_n^{-i}) = K \sum_{j=0}^{n-2t-1} C_j \omega_n^{-ij}, i = \overline{0, n-1}$$

($K = 1/n$ în câmpul complex; $K=1$ în câmp Galois).

Întrucât $C(x)$ este de ordin $(n-2t-1)$ și are doar $(n-2t-1)$ zerouri rezultă că cel puțin $(2t+1)$ componente ale oricărui cuvânt de cod sunt nenule.

Datorită liniarității transformatei Fourier discrete, diferența oricăror două cuvinte de cod este de asemenea cuvânt de cod deci conține cel puțin $(2t+1)$ valori nenule și distanța Hamming între ele este de cel puțin $(2t+1)$ simboluri ceea ce permite corecția a t erori de simbol.Q.E.D.

Decodarea RS se realizează cu algoritmi de calcul a vectorilor de eroare cu număr minim de componente nenule având $2t$ componente spectrale cunoscute la recepție:

$$\bar{r} = \bar{c} + \bar{e} \Rightarrow R_j = C_j + E_j = E_j \quad \forall j = \overline{n-2t, n-1} \quad (\text{IV.30})$$

Algoritmul de determinare a DFT a cuvântului de eroare

Fie polinomul de eroare $E(x)$ și un polinom $L(x)$ de localizare a erorilor (grad maxim = t) necunoscut:

$$L(x) = \prod_{i=1}^t (1 - x\omega_n^{k_i}) \quad (\text{IV.31})$$

unde k_i reprezintă pozițiile simbolurilor eronate.

Dacă L este vectorul coeficienților polinomului $L(x)$ de dimensiune n , prin com-

pletare cu zerouri și prin aplicarea IDFT se obține vectorul \mathbf{I} atunci:

$$l_i = K \prod_{j=0}^{n-1} \omega_n^{-ij} L_j = L(\omega_n^{-i}) = K \prod_{j=0}^{n-1} (1 - \omega_n^{-i} \omega_n^{kj}) = 0, \forall i = k_i \Rightarrow$$

$$l_i e_i = 0 \quad \forall i = \overline{0, n-1} \Rightarrow L(x)E(x) \text{ modulo } (x^n) = 0 \quad (\text{IV.32})$$

Produsul polinoamelor este echivalent cu o convoluție circulară a vectorilor asociați lor:

$$L_0 E_j + \sum_{k=1}^t L_k E_{j-k} = 0 \Leftrightarrow E_j = -\sum_{k=1}^{n-1} L_k E_{j-k}, \quad j = \overline{0, n-1} \quad (\text{IV.33})$$

Ultima relație descrie un filtru autoregresiv (AR) cu t coeficienți. Dacă se cunosc cele $2t$ componente ale vectorului \mathbf{E} atunci din relația (IV.33) se obține un sistem de $2t$ ecuații cu $2t$ necunoscute care permite determinarea coeficienților polinomului de localizare a erorilor:

$$E_{n-k} = -\sum_{j=1}^t L_j E_{n-j-k}, \quad k = \overline{1, t} \quad (\text{IV.34})$$

Prin cunoașterea vectorului \mathbf{L} se pot determina recursiv din relația (IV.33) toate componentele vectorului \mathbf{E} și apoi ale vectorului \mathbf{C} . Prin aplicarea IDFT rezultă vectorul de cod transmis și corectat.

Observație: Sistemul (IV.34) poate fi scris în formă echivalentă matricială:

$$-\begin{bmatrix} E_{n-t} \\ \dots \\ E_{n-1} \end{bmatrix} = \begin{bmatrix} E_{n-t-1} & \dots & E_{n-2t} \\ \dots & \dots & \dots \\ E_{n-2} & \dots & E_{n-t-1} \end{bmatrix} \begin{bmatrix} L_1 \\ \dots \\ L_t \end{bmatrix} \quad (\text{IV.35})$$

Soluționarea acestui sistem necesită calculul inversei unei matrici de tip Toeplitz (cu elemente identice pe subdiagonalele paralele cu diagonala principală), care se poate efectua cu algoritmi rapizi, utilizați frecvent în procesarea digitală a semnalelor în proiectarea filtrelor AR și în domeniul analizei spectrale a semnalelor.

Exemplu: Să realizăm construcția în $\text{GF}(8)$ a codului $\text{RS}(7; 5)$ corector de o eroare de simbol într-un bloc codat de lungime 7.

$$n = 7; n - 2t = 5; R = \frac{5}{7}; m = 3.$$

Se utilizează ca rădăcină a unității de ordin 7: $\omega_7 = 2$.

Polinomul generator al codului este:

$$g(x) = (x + 2^5)(x + 2^6) = (x + 7)(x + 5) = x^2 + x(5 + 7) + 5 \cdot 7 = x^2 + 2x + 6$$

Întrucât se codează blocuri de 15 biți prin cuvinte de cod de 21 de biți, tabelul de codare ar trebui să cuprindă 2^{15} (32768) linii. Astfel codul RS nu poate fi aplicat, asemenea altor coduri-bloc, pe baza conținutului unor memorii, ci folosindu-se algoritmi pentru calculul cuvântului de cod asociat fiecărui cuvânt de intrare.

Să calculăm vectorul codat pentru secvența de intrare: 2 3 0 1 4, prin metoda nesistematică de construcție a codului RS în domeniul timp:

$$a(x) = 2 + 3x + x^3 + 4x^4;$$

$$c(x) = a(x)g(x) = 7 + 5x + 4x^2 + 5x^3 + 7x^4 + 2x^5 + 4x^6.$$

Deci cuvântul de cod este: 7 5 4 5 7 2 4.

Decodarea se va face simplu prin împărțirea polinomului vectorului recepționat la polinomul generator. Corecția erorilor se realizează în alt bloc al decodatorului.

În varianta sistematică se obține:

$$r(x) = R\{x^{2t}a(x)/g(x)\} = 1 + 4x.$$

Deci secvența codată sistematic este: 1 4 2 3 0 1 4.

IV.3.3 Performanțele codurilor-bloc corectoare de erori

Performanțele codurilor-bloc corectoare de erori se exprimă prin diferiți parametri.

Un decodor care corectează un număr fix de erori se numește **decodor cu distanță limitată**. În cazul apariției mai multor erori, în funcție de modul de proiectare al decodatorului, acesta poate să semnaleze apariția unui mesaj incorect (imposibil de corectat), cum este cazul sistemelor ARQ (*Automatic Repeat under Request*).

Dacă transmisia datelor se face pe un canal de transmisie discret, fără memorie, performanțele codului se exprimă în termenii probabilității de eroare la ieșirea decodatorului în funcție de probabilitatea de eroare a simbolului pe CT.

Prezintă interes determinarea următoarelor mărimi:

- ❶ Probabilitatea de eroare a unui bloc de simboluri;
- ❷ Probabilitatea de eroare a simbolului decodat ;
- ❸ Probabilitatea imposibilității efectuării decodării.

Deoarece un bloc eronat conține și simboluri corecte, probabilitatea de eroare de bloc este mai mare decât probabilitatea de eroare de simbol.

Teoremă: Pentru un cod-bloc corector de t erori utilizat pe un CT discret, fără memorie, probabilitatea decodării corecte de bloc este dată de relația:

$$P_c = 1 - P_e = \sum_{i=0}^t C_n^i p_e^i (1 - p_e)^{n-i} \quad (\text{IV.36})$$

p_e reprezintă probabilitatea de eroare de simbol pe CT și n este lungimea blocului codat.

Justificarea acestei teoreme este evidentă ținând seama de numărul cazurilor posibile ca cele i erori să afecteze blocul de n simboluri când cel mult t erori sunt corectabile.

În cazul în care procesul de transmisie este afectat de zgomot aditiv, alb, gaussian (AWGN - *Additive White Gaussian Noise*), probabilitatea de eroare se exprimă în funcție de raportul de puteri semnal/zgomot al semnalului analogic transmis:

$\frac{E_b}{N_0}$ - raportul puterilor semnalului util și a zgomotului;

E_b - energia medie a unui bit de informație la ieșirea sursei;

$E_b = \frac{n}{k} E_c$

unde E_c este energia medie necesară transmiterii unui bit pe CT.

Un demodulator proiectat pentru a elimina influența AWGN se caracterizează prin **câștigul de codare** (*coding gain*) care se definește ca diferența (exprimată în dB) dintre raportul S/N necesar obținerii unei anumite valori a probabilității de eroare la ieșirea decodurului, în prezența și respectiv în absența codării. Experimental s-a constatat că acest câștig de codare nu poate fi oricât de mult crescut adică performanțele codurului nu se mai îmbunătățesc semnificativ, respectiv puterea de emisie necesară nu mai scade mult, prin folosirea unor coduri mai complicate. Există așadar o valoare asimptotică a câștigului de codare pentru valori mari ale raportului S/N, pentru cazul în care codul tinde să corecteze toate erorile posibile.

În realitate, puterea de emisie este limitată, deci nu se obține câștigul asimptotic de codare iar probabilitatea decodării eronate este mai mare decât cea estimată.

Prin definiție, **câștigul asimptotic** al unui cod-bloc corector de t erori, cu rata de codare R , are expresiile:

$$\blacklozenge \text{ pentru decizie hard: } G = (t+1)R \quad (\text{IV.37})$$

$$\blacklozenge \text{ pentru decizie soft: } G = (2t+1)R \quad (\text{IV.38})$$

Observație: Câștigul de codare are semnificația creșterii artificiale a puterii semnalului util, respectiv a energiei de bit, deoarece prin folosirea unui cod corector de erori aceeași probabilitate de eroare la demodulator se obține cu o putere de emisie mai mică față de cazul neaplicării codului.

Pentru un demodulator coerent BPSK polar, cu decizie hard și cod corector de t erori, expresia probabilității de eroare este [Bla90]:

$$P_e \cong C_n^{t+1} Q \left[\sqrt{R(t+1) \frac{2E_b}{N_0}} \right] \quad (\text{IV.39})$$

unde $Q(x)$ este funcția normală de eroare:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \text{Exp}\left[-\frac{y^2}{2}\right] dy \quad (\text{IV.40})$$

iar energia echivalentă a unui bit transmis este:

$$E_b' = GE_b = R(t+1)E_b \quad (\text{IV.41})$$

În cazul deciziei soft, când demodulatorul retransmite date decodurii și acestea contribuie la corectarea unor erori ulterioare, câștigul de codare crește față de cazul deciziei hard fără reacție.

În cazul transmisiei independente a biților, valoarea **probabilității de eroare de bit** (BER - *Bit-Error-Rate*) este mai mare decât probabilitatea de eroare de simbol raportată la lungimea blocului de biți.

Să analizăm performanțele codurilor Hamming. În tabelul IV.3 sunt specificate procentual creșterea redundanței semnalului, capacitatea de detecție respectiv de corecție a erorilor pentru fiecare cod în parte.

Tabelul IV.3

Performanțele codurilor Hamming H(n;k)

Parametru	Lungimea a secvenței codate	Lungimea a secvenței de intrare	Numărul de erori detectabile	Numărul erorilor corectabile	Creșterea redundanței	Capacitatea de detecție a erorilor	Capacitatea de corecție a erorilor
m	n	k	d	c	(%)	(%)	(%)
3	7	4	1	1	75,00	14,29	14,29
4	15	11	2	1	36,36	13,33	6,67
5	31	26	2	2	19,23	6,45	6,45
6	63	57	3	2	10,53	4,76	3,17
7	127	120	3	3	5,83	2,36	2,36

Se observă că în cazul codului H(7;4) se obține capacitatea maximă de detecție și corecție a erorilor dar creșterea redundanței este de asemenea cea mai ridicată dintre toate codurile. Un compromis acceptabil între diferiții parametri ai codului este realizat în cazul codului H(15,11), detector de erori duble, corector de o eroare. Codurile Hamming pot fi folosite în transmisii cu zgomot aditiv, gaussian, pentru corecția unor erori aleator amplasate în semnal, dar nu și în cazul transmisiilor cu fading, când apar grupuri de erori (*error-burst*) de diferite lungimi.

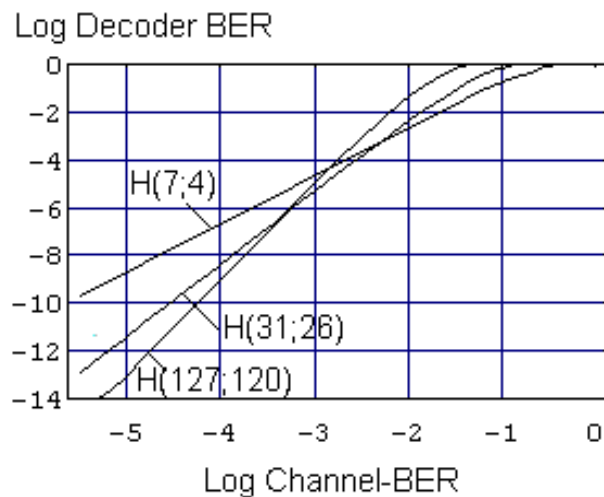


Fig. IV.2 Probabilitatea de eroare de bit a decodurii Hamming funcție de probabilitatea de eroare de bit a CT

În figura IV.2 sunt reprezentate graficele probabilității de eroare de bit a decodurilor Hamming corectoare de 1; 2 respectiv 3 erori, în funcție de probabilitatea de eroare de bit proprie canalului de transmisie afectat de zgomot alb, aditiv, gaussian.

Se observă că pentru o probabilitate de eroare de bit pe CT de 10^{-4} , decodorul asigură o rată de eroare de bit de 10^{-7} în cazul codului H(7;4), respectiv de 10^{-9} pentru H(127;120).

Performanțe superioare de corecție a erorilor se obțin prin folosirea codurilor Reed-Solomon, deosebit de puternice comparativ cu cele Hamming și cele convoluționale. Decodurile RS, implementate cu procesoare dedicate pentru câmpuri Galois, pot lucra în timp real, la viteze mari de transmisie, ceea ce le recomandă pentru utilizarea în sistemele de transmisie a datelor de mare viteză, cu precizie ridicată. De asemenea aceste coduri pot fi utilizate în cazul transmisiilor cu fading având capacitatea de corecție a unor grupuri mari de erori (*error-burst*).

În tabelul IV.4 sunt prezentate performanțele unor coduri RS, care au capacitate de corecție a erorilor de peste 6% și o creștere a redundanței de maxim 50% (corespunzător unei rate de codare 1/2: *half-rate*). Deosebit de puternice sunt codurile RS construite în GF(256), cu simboluri exprimate pe 8 biți.

Tabelul IV.4 Performanțele unor coduri Reed-SolomonRS($n;n-2t$) definite în $GF(2^m)$ ($m = 3;4;5;6;7;8$)

Numărul de biți dintr-un simbol	Lungimea secvenței codate (simboluri)	Lungimea secvenței de intrare (simboluri)	Numărul erorilor corectabile t(simboluri) b(biți consecutivi)		Cresterea redundanței (%)	Capacitatea de corecție a erorilor (%)	Numărul de biți de ieșire
m	n	k = n-2t	t	b	r	γ	mn
3	7	5	1	1	40,00	14,29	21
4	15	11	2	5	36,36	13,33	60
5	31	27	2	6	14,81	6,45	155
	31	25	3	11	24,00	9,68	
	31	23	4	16	34,78	12,90	
	31	21	5	21	47,62	16,13	
6	63	55	4	19	14,55	6,35	378
	63	51	6	31	23,53	9,52	
	63	47	8	43	34,04	12,70	
	63	45	9	49	40,00	14,29	
	63	43	10	55	46,51	15,87	
7	127	111	8	50	14,41	6,30	889
	127	103	12	78	23,30	9,45	
	127	95	16	106	33,68	12,60	
	127	85	21	141	49,41	16,54	
8	255	223	16	121	14,35	6,27	2040
	255	171	42	329	49,12	16,47	

Observație: Pentru transmisiile cu fading este necesară corecția unor grupuri de erori cu lungime care depinde atât de durata maximă a intervalelor cu fading, cât și de viteza de transmisie a datelor. În tabelele conținând codurile RS cele mai performante, este calculat numărul maxim de biți, respectiv de simboluri consecutive corectabile printr-un anume cod.

Cresterea dimensiunilor codului conduce la o capacitate mai mare de corecție a grupurilor de erori (*error-burst*), deci permit creșterea vitezei de transmisie și reducerea energiei emise în cazul transmisiilor digitale afectate de fading.

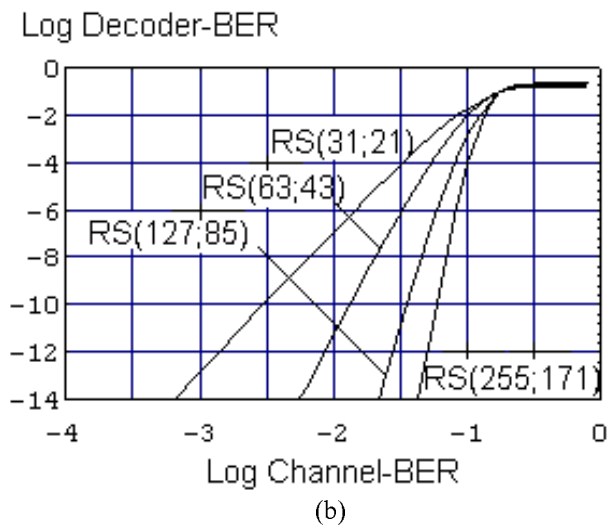
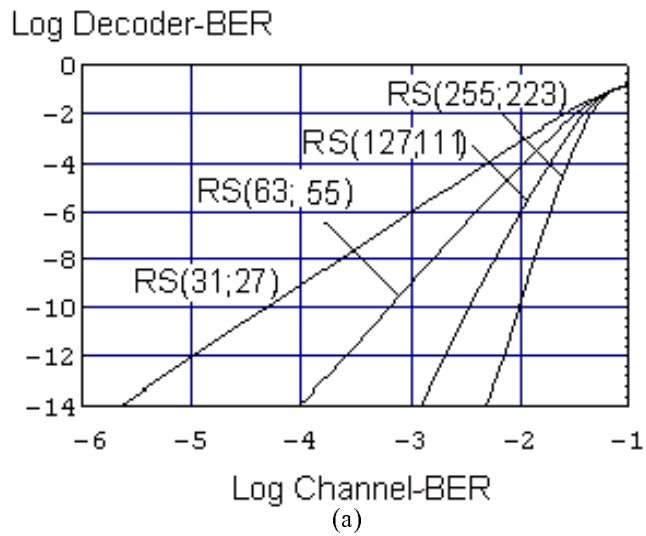


Fig. IV.3 Probabilitatea de eroare de bit a decodului RS funcție de probabilitatea de eroare de bit a CT afectat de zgomot aditiv, pentru o

De exemplu, prin folosirea codului RS(127, 95) se poate crește de două ori viteza de transmisie a datelor față de cazul utilizării codului RS(127,111), în aceleași condiții de zgomot aditiv, fading, la aceeași energie de bit, cu aceeași probabilitate de eroare (BER).

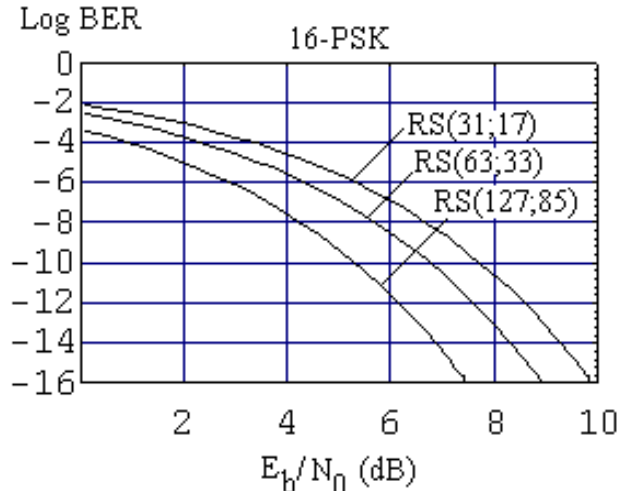


Fig.IV.4 Probabilitatea de eroare logaritmată a demodulatorului 16-PSK, cu precodor RS, pentru transmisie cu zgomot aditiv gaussian

Funcțiile de probabilitate de eroare de bit pentru ieșirile decodoarelor RS sunt reprezentate grafic în figura IV.3. Sunt comparate performanțele codurilor care conduc la aceeași creștere a redundanței semnalului.

Se observă că pentru o rată de eroare de bit pe CT de 10^{-2} , decodorul asigură o rată de eroare de bit de:

◆ 10^{-3} pentru codul RS(31;27); 10^{-4} pentru RS(63;55); 10^{-6} pentru RS(127;111); și de 10^{-10} pentru RS(255;223);

◆ 10^{-7} pentru RS(31;21); 10^{-11} pentru RS(63;43).

În figura IV.4 sunt prezentate graficele probabilității de eroare a unui demodulator digital de fază cu 16 stări, cu precodor RS ca rată de codare 1:2 (*half-rate*).

Se observă comportarea bună la rapoarte S/N mai mici de 5 dB pentru codurile de mari dimensiuni.

IV.4 Coduri -arbore corectoare de erori

În contrast cu codurile-bloc, codurile de tip "arbore" (*tree-codes*) codează șirul simbolurilor de intrare, cu început și sfârșit nedeterminat, într-un șir de simboluri de ieșire.

Interes practic prezintă numai acele coduri care permit implementarea codorului ca automat cu număr finit de stări. În acest caz, codurile se numesc **coduri trellis**. Acestea pot fi reprezentate prin diagrame de tip "grilă" (*trellis*), desfășurate în timp, cu număr finit de noduri pe fiecare nivel. Un cod trellis liniar se numește **cod convoluțional**.

Șirul simbolurilor de intrare în codor este împărțit în segmente de **k** simboluri numite **cadre**. Codorul memorează ultimele **m** cadre, adică ultimele **mk** simboluri de date și în funcție de acestea și de cadrul curent generează **n** simboluri de ieșire. Orice combinație liniară de secvențe de cod convoluțional este de asemenea secvență de cod.

Condiția de liniaritate se exprimă astfel:

$\forall \bar{c}_1, \bar{c}_2$ cuvinte de cod ; $\forall \alpha, \beta \in GF$: $\alpha\bar{c}_1 + \beta\bar{c}_2$ este cuvânt de cod.

Această condiție se verifică pe fiecare nivel din 'trellis'.

Cuvintele de cod-arbore sunt infinit lungi și se citesc pe calea din 'trellis' specificată de secvența de intrare.

Aplicațiile codurilor convoluționale cuprind comunicațiile de viteză mică și medie în raport cu banda de transmisie, de exemplu sisteme digitale de transmisie vocală. Dezavantajul major al codurilor convoluționale este dat de numărul mare de operații efectuate pentru fiecare bit decodat, prin aplicarea algoritmului Viterbi în caz binar [Ber87]. O posibilă optimizare a acestor tehnici de codare este metoda de proiectare a lor în câmp Galois [Ugr93] utilizând simboluri exprimate pe un număr mai mare de biți. Evident este necesară aplicarea algoritmului Viterbi în câmpul algebric finit de definiție a codului.

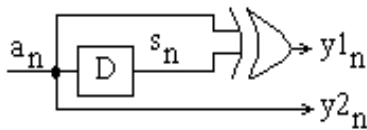


Fig.IV.6 Codor C(11,10)

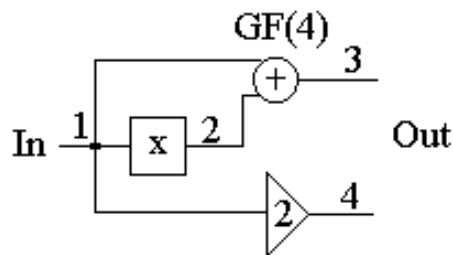


Fig.IV.7 Codor convoluțional C(11;20) în GF(4)

Dacă dimensiunea câmpului este 2 atunci avem de a face cu coduri binare. Pentru simplitatea implementării codorului și a decodorului se preferă proiectarea

codurilor în varianta sistematică. Un codor convoluțional binar este implementat ca filtru digital cu sau fără reacție, cu celule de memorie și sumatoare modulo-2 (exemplu: Fig.IV.6). În general, un codor convoluțional multinivel utilizează sumatoare și multiplicatoare specifice câmpului Galois de definiție a codului (exemplu: Fig.IV.7).

Observație: Notarea codurilor nu se face în mod unic. Am adoptat metoda citirii coeficienților intrărilor în sumatoare sau în ieșirile codorului, în ordine crescătoare a puterilor operatorului de întârziere (D sau x).

Cele două codoare folosite pentru exemplificare sunt structuri nerecursive.

Codurile convoluționale pot fi implementate în varianta sistematică, folosind structuri de codare recursive (RSC - *recursive systematic code*) (fig. IV.8). Astfel se elimină structura de decodare păstrându-se numai blocul de control al erorilor.

Observație: Nu întotdeauna prin sistematizare se obțin structuri recursive. Am notat aceste codoare nerecursive abreviat NSC (*Non-Recursive Systematic Convolutional Code*).

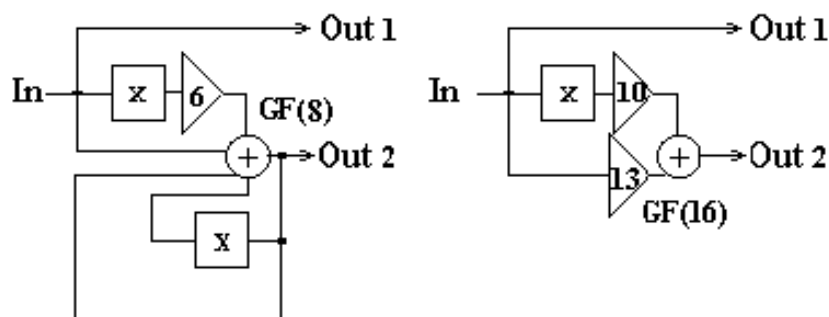


Fig.IV.8 Schemele-bloc ale unor codoare RSC și NSC definite în GF

IV.4.1 Descrierea matematică a codurilor convoluționale

Analiza codurilor convoluționale se face combinat, prin metode grafice și matriciale.

Parametrii codorului:

k - numărul simbolurilor de intrare;

n - numărul simbolurilor de ieșire;

$R = k/n$ - rata codorului;

M - numărul celulelor de memorie;

m - numărul de biti utilizați pentru exprimarea binară a simbolurilor;

$s = m^M$ - numărul stărilor circuitului;

v - **lungimea de constrângere** a codorului, definită ca numărul minim de celule de memorie necesare în circuit.

Exemplu: Codorul binar $C(11,10)$ are parametri:

$k = 1; n = 2; R = 1/2; M = 1; m = 2; v = 1$. Notăție echivalentă: $C(3;2)$.

Descrierea matricială a codurilor convoluționale se face fie cu o matrice generatoare cu elemente polinomiale, fie cu matrici de tip intrare-ieșire-stare cu coeficienți constanți.

Descrierea grafică a unui astfel de cod se realizează cu diagrame:

- ◆ de tranziții (de stări) ;
- ◆ de tip "arbore";
- ◆ de tip "grilă" (*trellis*).

Diagramele de stări sunt complicate în cazul codurilor cu număr mare de stări și cu cel puțin două simboluri de intrare. De asemenea diagramele "arbore" sunt mai puțin practice întrucât numărul nodurilor de pe un anumit nivel, crește exponențial de la un moment de timp la altul. Cele mai eficiente sunt diagramele 'trellis' care au același număr de noduri pe toate nivelele și devin periodice după un număr minim de pași, corespunzător regimului tranzitoriu. Și aceste diagrame sunt dificile pentru mai multe combinații intrare-stare.

Devine utilă în acest caz transcrierea trellisului sub formă de tabel în care să se specifice toate combinațiile posibile privind: starea codorului la momentul curent, cuvântul de intrare, secvența codată și starea în care trece circuitul la momentul următor.

IV.4.2 Analiza codului convoluțional pe baza matricii generatoare

Matricea generatoare a unui codor convoluțional este formată din $k \times n$ **polinoame generatoare** semnificând contribuția fiecărei intrări la fiecare variabilă de ieșire: $G(x) = [g_{ij}(x)]$, $i = \overline{1, k}$, $j = \overline{1, n}$ (IV.42)

Notații: x - operatorul de întârziere cu un simbol; $g_{ij}(x)$ - polinomul generator asociat intrării i și ieșirii j , cu coeficienți având valori din câmpul Galois de definiție a codului.

Lungimea de constrângere a codului se calculează ca:

$$v = \sum_{i=1}^k \max_j [\text{grad } g_{ij}(x)] \quad (\text{IV.43})$$

În cazul codurilor sistematice, matricea generatoare trebuie să conțină o submatrice de tip identitate cu k linii:

$$G(x) = [I \mid P(x)] \quad (\text{IV.44})$$

Se asociază fiecărei intrări câte un polinom. Toate aceste polinoame pot fi grupate într-un vector de intrare cu elemente polinomiale:

$$\bar{a}(x) = [a_1(x), \dots, a_k(x)] \quad (\text{IV.45})$$

Coeficienții acestor polinoame sunt simbolurile de date.

$$\text{Secvența codată se obține simplu: } \bar{c}(x) = \bar{a}(x)G(x) \quad (\text{IV.46})$$

Pentru corecția erorilor din secvența recepționată, se poate folosi **matricea de control** pentru calculul vectorului sindrom:

$$H(x) = [-P^T(x) \mid I] \quad (\text{IV.47})$$

Definiție: Un cod convolutional este **necatastrofic** dacă satisface relația:

$$\text{c.m.m.d.c. } [D_k(x)] = x^r \quad (\text{IV.48})$$

unde r este un număr întreg oarecare, $D_k(x)$ sunt determinanții submatricilor de dimensiuni $k \times k$ din matricea generatoare a codului.

Evident, matricea generatoare a unui cod sistematic conține o submatrice de tip 'matrice identitate' cu determinant de valoare 1, deci codul este necatastrofic.

În cazul codurilor necatastrofice, matricea generatoare admite inversă și decodarea se poate face pe baza relației:

$$\bar{a}(x)x^r = \bar{c}(x)G^{-1}(x) \quad (\text{IV.49})$$

Exemplu: Fie matricea generatoare:

$$G(x) = \begin{bmatrix} 1 & x^2 & 1 \\ 0 & 1 & x+x^2 \end{bmatrix}; D(x) = [1; x+x^2; 1+x^3+x^4] \text{ deci codul este necatastrofic.}$$

Matricea inversă a decodului:

$$G^{-1}(x) = \begin{bmatrix} x^3+x^4 & 1+x^2+x^3+x^4 \\ x+x^2 & 1+x+x^2 \\ 1 & 1 \end{bmatrix}$$

IV.4.3 Analiza codorului convolutional cu ecuații de stare

Descrierea codorului în varianta matricială poate fi făcută cu matrici fixe caracteristice pornind de la ecuațiile codorului în varianta paralel-paralel [Bed89]:

$$\textcircled{1} \text{ Ecuatia intrare-iesire: } \bar{c}_p = A_{n \times k} \bar{a}_p + B_{n \times M} \bar{s}_p \quad (\text{IV.50})$$

$$\textcircled{2} \text{ Ecuatia de stare: } \bar{s}_{p+1} = C_{M \times k} \bar{a}_p + E_{M \times M} \bar{s}_p \quad (\text{IV.51})$$

Notatii:

c, a, s - vectorii-coloană de iesire, intrare și stare;

A - matricea ieșire-intrare;

B - matricea iesire - stare;

C - matricea stare - intrare;

E - matricea stare - stare;

p - variabila discretă de timp;

Deducerea matricii generatoare polinomiale pe baza matricilor caracteristice ale codorului se face iterativ:

$$\bar{c}_n = \mathbf{A}\bar{a}_n + \mathbf{B}\bar{s}_n = \mathbf{A}\bar{a}_n + \mathbf{B}(\mathbf{C}\bar{a}_{n-1} + \mathbf{E}\bar{s}_{n-1}) = \mathbf{A}\bar{a}_n + \mathbf{BCD}\bar{a}_n + \mathbf{BED}\bar{s}_n$$

$$\bar{c}_n = \mathbf{A}(\mathbf{I} + \mathbf{BCD} + \mathbf{BECD}^2 + \mathbf{BE}^2\mathbf{CD}^3 + \dots + \mathbf{BE}^{v-1}\mathbf{CD}^v)\bar{a}_n$$

Rezultă expresia matricii generatoare:

$$\mathbf{G}(D) = \mathbf{A} + \sum_{i=0}^{v-1} (\mathbf{BE}^i\mathbf{C})D^{i+1} \quad (\text{IV.52})$$

Observație:

$$\text{grad}[\mathbf{G}(D)] = v \text{ si } \mathbf{E}^v = 0$$

Exemple:

Codul C(11;10) poate fi descris cu matricile fixe: $\mathbf{A} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$; $\mathbf{B} = [1]$; $\mathbf{C} = [1]$; $\mathbf{E} = [0]$.

Codul C(111;101) binar este descris de matricile:

$$\mathbf{A} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \mathbf{E} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

IV.4.4 Algoritmul de decodare Viterbi

Se preferă însă decodarea prin folosirea algoritmului Viterbi [For73], [Vit79] care lucrează pe principiul distanței Hamming minime între cuvântul de cod recepționat și cel estimat pentru a realiza și corecția unor eventuale erori.

Algoritmul Viterbi cuprinde următorii pași :

① Se inițializează codorul în starea nulă pe nivelul $k = 0$.

② Se inițializează ponderile stărilor: $w_s^0 = 0$

③ Se incrementează variabila de nivel: $k=k+1$.

④ Se calculează metrica fiecărei căi ce intră în nodurile de pe nivelul k al trellisului.

⑤ Se reține numai calea de intrare în fiecare stare care are metrica minimă (*surveiver*): d_s .

⑥ Se calculează ponderile stărilor pe nivelul k : $w_s^k = w_s^{k-1} + d_s$ (IV.53)

⑦ Dacă L este lungimea secvenței recepționate și $k < L+M$, se revine la pasul 3.

⑧ Se decide că s-a recepționat secvența corespunzătoare căii ce intră în starea cu normă minimă.

Pentru o secvență de L biți, pentru aplicarea algoritmului Viterbi se efectuează $L2^{v-1}$ operații. În practică, se limitează lungimea de constrângere a codorului la 5.

Algoritmul Viterbi reprezintă un caz particular de cercetare operațională pentru găsirea drumului minim într-un graf. În prezent, decodoarele Viterbi sunt implementate în tehnologie VLSI.

Reducerea numărului de operații efectuate pentru decodarea unui simbol (bit) prin aplicarea algoritmului Viterbi, poate fi redus prin **segmentarea procesului de decizie**. Aceasta constă în luarea de decizii parțiale pe durata decodării, de îndată ce s-a obținut pe un anumit nivel din trellis o stare unică de pondere minimă, cu condiția ca de la ultima reinițializare să fi fost analizate minimum L nivele (L - **lungimea de decizie** a codului).

Pentru primul nivel, când se analizează pe baza algoritmului Viterbi numai secvențele de cod asociate unei stări (**inițiale**), numărul de operații efectuate este substanțial redus față de alte nivele pentru care analiza cuprinde toate stările codorului.

Exemplu: Codorul $C(11,10)$ binar are matricea generatoare: $G(D) = \begin{bmatrix} 1+D \\ 1 \end{bmatrix}$.

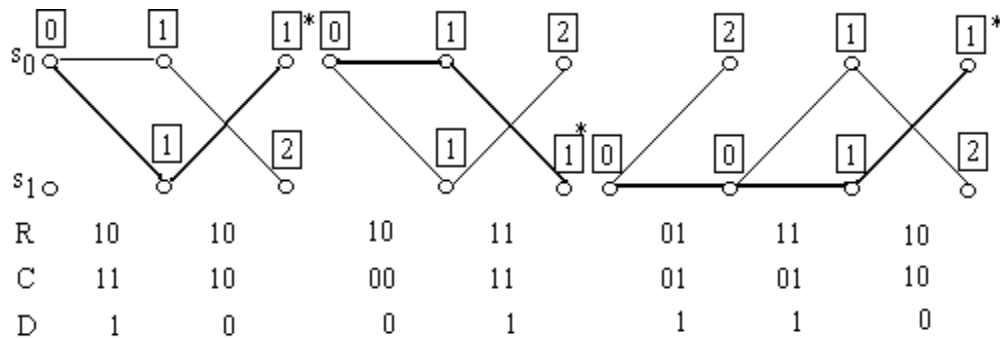
În tabelul IV.5 se specifică tranzițiile între stări, pentru codorul considerat.

Tabelul IV.5 Codor convoluțional binar $C(3;2)$

s_n	a_n	$y_{1n}y_{2n}$	s_{n+1}
0	0	00	0
	1	11	1
1	0	10	0
	1	01	1

Considerăm o secvență binară particulară a_n , cu vectorul de stare asociat și secvențele codate, transmisă respectiv recepționată (cu erori):

a_n 1 0 0 1 1 1 0
 s_n 0 1 0 0 1 1 1
 $y_{1n}y_{2n}$ 11 10 00 11 01 01 10
 $r_{1n} r_{2n}$ 10 10 10 11 01 11 10.



Vom aplica algoritmul de decodare Viterbi cu decizie segmentată. Se obțin 3 segmente de diagramă cu 2, respectiv 3 nivele.

(R - secvență recepționată; C - secvență corectată; D - secvență decodată).

Pentru codul convoluțional binar C(11;10) cu 2 stări, rată de codare 1:2 și lungime de decizie 2, se vor efectua în cazul aplicării algoritmului Viterbi clasic, pentru fiecare bit decodat, 4 operații de adunare. Dacă se segmentează procesul de decizie și se reinițializează trellis-ul la fiecare două nivele atunci se vor efectua în medie doar 3 operații pentru un bit decodat, ceea ce conduce la o reducere de 25% a numărului total de operații efectuate pentru decodare precum și a timpului afectat procesului. Acest fapt este deosebit de util în cazul decodării în timp real a datelor.

Observații:

- ❶ În cazul codurilor binare, metrica unei căi din trellis se determină pe baza **distanței Hamming** dintre grupurile de biți.
- ❷ În general, pentru decodarea Viterbi a codurilor construite în câmp Galois, se va folosi **distanța euclidiană pătratică** dintre grupurile de simboluri multibit.
- ❸ **Lungimea de decizie a codului (L)** va fi definită în paragraful următor. În esență, aceasta corespunde numărului minim de nivele din graf care asigură capacitatea maximă de corecție a erorilor pe baza codului analizat.
- ❹ Dacă pe ultimul nivel există mai multe stări cu pondere de valoare minimă, se decide detecția erorilor și imposibilitatea corecției lor pe acel nivel. Sunt necesare în acest caz, date suplimentare pentru a realiza corecția erorilor.

IV.4.5 Performanțele codurilor convoluționale

Un cod convoluțional este în principiu infinit lung și are scopul de a corecta erorile distribuite aleator, pe măsura apariției lor. În cazul în care numărul erorilor care afectează un segment codat depășește capacitatea de corecție a decodorului, se marchează intervalul respectiv și se cere retransmisia lui. Acest eveniment se numește **eveniment de nedecodare**.

Capacitatea unui cod convoluțional de detecție și corecție a erorilor se studiază prin:

- ◆ structura distanțelor codului (**profilul distanțelor**);
- ◆ probabilitatea de decodare eronată ca funcție de distanța Hamming minimă (*free Hamming distance*) dintre oricare două secvențe codate infinit lungi.

Definiție: Distanța minimă de ordin L a codului convoluțional este egală cu distanța Hamming minimă dintre oricare două segmente de L cadre ale cuvântului de cod care nu corespund în cadrul inițial.

Secvența monoton crescătoare a distanțelor Hamming minime de ordin 1; 2; 3; ... se numește **profilul distanțelor** codului.

În cazul codurilor liniare, procedura de calcul a distanțelor codului se simplifică prin raportarea tuturor segmentelor codate la calea fundamentală corespunzătoare secvenței infinite de zerouri (*all-zero word*). Trebuie calculată în acest caz doar ponderea minimă a segmentelor codate de L cadre care nu sunt nule pe primul nivel.

Limita superioară a șirului distanțelor codului se numește **distanța Hamming minimă** (*free-Hamming distance*) a codului.

Pentru un cod convoluțional cu distanța minimă de ordin L de valoare d_L se pot corecta t erori în primele L cadre dacă $2t+1 \leq d_L$.

Exemplu: Să determinăm profilul distanțelor și distanța Hamming minimă a codului convoluțional binar C(3;2) (Fig.IV.9).

Pentru primul nivel al trellisului se calculează distanța Hamming între cuvintele de cod: 00 și 11. Rezultă distanța de ordin 1: $d_1 = 2$.

Pentru distanța de ordin 2, se vor compara secvențele de două cadre care nu coincid în primul cadru: setul1 (0000;0011) cu setul al doilea (1110;1101). Din cele 4 combinații posibile rezultă: $d_2 = 3$.

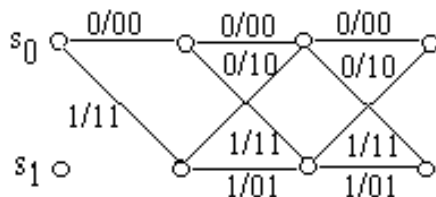


Fig.IV.9 Diagrama 'trellis' a codului C(11;10)

Același rezultat se obține simplu prin calculul ponderii minime a secvențelor din setul al doilea, nenule în primul cadru.

Distanța de ordin 3 corespunde ponderii minime a secvențelor: (111000;111011;110110;110101): $d_3 = 3$.

Întrucât această valoare se obține pentru calea 111000 care se unește cu cea fundamentală, rezultă că s-a obținut limita superioară a valorilor profilului distanțelor {2; 3; 3 ...}, adică distanța Hamming minimă a codului: $d_{Hfree} = 3$.

Câștigul de codare asimptotic al unui cod convoluțional binar cu rata de codare R și distanța Hamming minimă d_{Hfree} se calculează cu relația:

$$G_a = R d_{Hfree} \quad (IV.54)$$

Observații:

❶ În cazul semnalelor modulate, performanțele transmisiei depind de distanța euclidiană dintre semnale și nu de cea Hamming. Distanța euclidiană pătratică dintre două stări are semnificația energiei necesare pentru a se efectua tranziția respectivă.

Se calculează pentru fiecare constelație de semnal distanța euclidiană minimă, respectiv medie, care determină probabilitatea de eroare maximă, respectiv medie, a demodulatorului.

❷ Performanțele de corecție a erorilor ale codurilor convoluționale se exprimă prin:

- ◆ profilul distanțelor și distanța Hamming minimă;
- ◆ lungimea minimă a unui segment codat care asigură corecția numărului maxim de erori. Acest parametru permite realizarea unei comparații între codurile-bloc corectoare de erori și cele convoluționale.

Exemplu: Codul convoluțional binar C(31;13;12) [Scr95/1] cu parametrii:

- ◆ $k = 2$ intrări paralele;

- ◆ $n = 3$ ieșiri paralele;
- ◆ $R = 2/3$ rata de codare;
- ◆ $M = 2$ celule de memorie;
- ◆ $s = 4$ stări;
- ◆ $v = 1$ (lungimea de constrângere);
- ◆ $H = 4$ (distanța Hamming minimă);
- ◆ $c = 1$ eroare corectabilă;
- ◆ $d = 2$ erori detectabile;
- ◆ $L = 2$ cadre (lungimea minima necesară a segmentului recepționat pentru corecția unei erori);

Matricea generatoare:

$$G(D) = \begin{bmatrix} 1+D & 1 \\ 1 & 1+D \\ 1 & D \end{bmatrix}$$

Codul are un câștig de codare $G = 8/3$ (4,771dB) fiind cel mai performant din familia codurilor convoluționale binare, cu două variabile de stare și un nivel, cu structură de stare simetrică (total 4096 coduri posibile).

Codul prezentat corectează o eroare și detectează două erori dintr-un segment codat recepționat de 6 biți.

Definiții:

❶ **lungimea de decizie:** (L) este numărul minim de cadre după care în profilul distanțelor se obține distanța Hamming minimă totală a codului;

❷ **eficiența de detecție a erorilor** se definește prin relația $\delta = d/nL$;

❸ **eficiența de corecție a erorilor** are relația de definiție $\gamma = c/nL$;

(d - numărul erorilor detectabile în L cadre recepționate; c - numărul erorilor corectabile în L cadre recepționate.)

Pentru codul $C(31;13;12)$ se obțin: $\delta = 2/6 = 33,33\%$ și $\gamma = 1/6 = 16,66\%$, comparativ cu codurile-bloc corectoare de erori:

◆ Hamming:

$H(7,4)$ $\delta = \gamma = 1/4 = 25\%$;

$H(15, 11) \delta = 2/11 = 18,18\% \quad \gamma = 1/11 = 9.09\%$;

◆ Reed-Solomon:

$RS(7,5) \delta = \gamma = 1/5 = 20\%$;

$RS(15,11) \delta = \gamma = 2/11 = 18,18\%$.

Observatii:

◆ Codurile convoluționale sunt mai simple din punctul de vedere al implementării circuitelor de codare/decodare decât codurile-bloc corectoare de erori .

Codurile convoluționale binare optime, cu 1; 2; 3 sau 4 variabile de stare sunt prezentate în tabelul IV.6.

TABELUL IV.6

Coduri convoluționale binare optime

Cod convoluțional	Numărul variabilelor de stare	Rata codorului	Distanța Hamming minimă	Număr de erori corectabile	Câștigul de codare [dB]
C(3;1)*	1	1:2	3	1	1,76
C(11;01;31)	1	2:3	3	1	3
C(5;7)	2	1:2	5	2	3,98
C(31;11;32)	2	2:3	4	1+d	4,26
C(7;11)	3	1:2	6	2+d	4,77
C(71;53;35)	4	2:3	5	2	5,23

(* citirea coeficienților polinoamelor generatoare se face în ordine descrescătoare a puterilor operatorului D).

◆ Codurile convoluționale multinivel, construite în câmpuri Galois, asigură performanțe superioare de corecție a erorilor [Ugr93].

Algoritmul Viterbi se va aplica în cazul transmisiei multinivel folosind pentru calculul metricii unei căi din trellis funcția '**distanța euclidiană pătratică**' dintre simbolurile componente ale diferitelor secvențe:

$$d_E^2(\bar{x}; \bar{y}) = \sum_{i=1}^n d_E^2(x_i; y_i) = \sum_{i=1}^n (x_i - y_i)^2 \quad (\text{IV.55})$$

Notatii: $\bar{x} = [x_1 x_2 \dots x_n]$; $\bar{y} = [y_1 y_2 \dots y_n]$ secvențe de n simboluri din GF.

Determinarea distanței euclidiene pătratice minime și a lungimii minime de de-

cizie se va face pe baza **profilului distanțelor codului**. Câștigul de codare a codului convoluțional se calculează pe baza '**distanței euclidiene pătratice minime**' ($d_{E\ free}^2$) dintre oricare două secvențe codate de lungime nelimitată, care nu coincid în primul cadru (R - rata de codare):

$$G = R \cdot d_{E\ free}^2 \quad (IV.56)$$

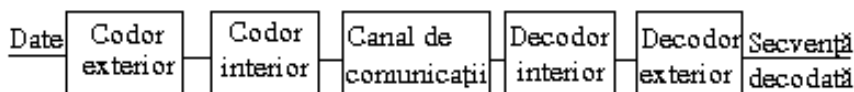


Fig.IV.10 Codor mixt corector de erori

În concluzie, pentru transmisiile de mică viteză, afectate doar de zgomot aditiv aleator, codurile convoluționale binare sunt mai avantajoase decât codurile-bloc corectoare de erori fiind și mai simple din punctul de vedere a implementării circuitelor de codare/decodare. În transmisiile de mare viteză sau în cele afectate de fading, sunt mai avantajoase codurile Reed-Solomon și cele convoluționale construite în câmp Galois (pe simboluri exprimabile pe un număr mare de biți), cu capacitate mare de detecție și corecție a erorilor, aleatoare sau grupate, cu posibilitate de decodare în timp real. În prezent se utilizează metode mixte de codare (*concatenated-codes*) care să combine avantajele celor două tehnici (Fig.IV.10).

Aceste coduri au fost prima oară propuse de Forney (1966) dar metoda a evoluat prin îmbunătățirea principiului de intercalare a simbolurilor. Utilizarea în serie a două sau mai multe codoare, în vederea obținerii unui câștig de codare superior, este frecventă în sistemele de comunicație cu putere de transmisie limitată, ca în cazul transmisiilor prin satelit. Recent (1993) au fost introduse codurile de tip **turbo** care utilizează un produs de

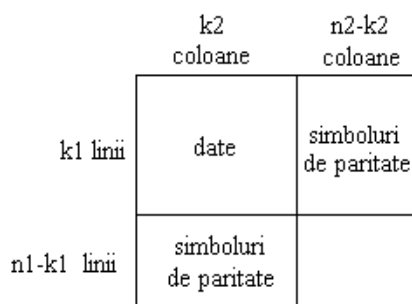


Fig.IV.11 Principiul codării turbo

doă coduri corectoare de erori, cu simbolurile de date intercalate (fig.IV.11) [Sk197]. Decodarea datelor se poate face pe principiul estimatorului de maximă probabilitate. Codurile folosite, 'bloc' sau 'arbore', sunt în general sistematice pentru a ușura decodarea.

IV.5 Aplicații

P1. Codăți ASCII pe 8 biți, cu paritate pară, secvența de caractere: a1b2c3d4.

P2. Calculați 3-DFT a șirului de simboluri (10; 4; 1; 5; 2; 6) în GF(16) cu $p(x)=x^4+x+1$ și $\omega_3=6$.

P3. Calculați 5-DFT a secvenței (12; 8; 3; 0; 1) în GF(16) cu $p(x)=x^4+x+1$ și $\omega_5=8$.

P4. Calculați 7-DFT a secvenței (1; 6; 5; 0; 2; 0; 1) în GF(8) cu $p(x)=x^3+x+1$ și $\omega_7=5$.

P5. Codăți în domeniul timp secvența (3 1 2 7 5) în cod RS(7;5) în GF(8) cu $p(x)=x^3+x+1$ și $\omega_7=4$.

P6. Codăți secvența (11 9 7) în cod RS(5;3) în GF(16) cu $p(x)=x^4+x+1$ și $\omega_5=10$ prin metoda IDFT.

P7. Codăți secvența (2 5 0 4 1) în cod sistematic RS(7;5) în GF(8) cu $p(x)=x^3+x+1$ și $\omega_7=2$.

P8. Scrieți matricea de paritate și cea generatoare a codului Hamming H(15;11). Calculați (în dB) câștigul codului.

P9. Pe un canal de transmisie binar cu probabilitate de eroare a bitului de 10^{-4} , se folosește codul Hamming H(7;4). Calculați:

- probabilitatea de de codare corectă;
- probabilitatea de nedetectare a erorilor.

P10. Codul RS(7;5) este utilizat pentru transmisie pe un canal octal, cu probabilitate de eroare de simbol 10^{-3} . Calculați probabilitatea de eroare a decodului.

P11 Un codor convoluțional binar cu polinoamele genratoare $g1(x)=x^2+x+1$ și $g2(x)=x^3+1$ este pus în cascadă cu un codor-bloc (regula de codare: 00 -> 010; 01 -> 011; 10 -> 101; 11 -> 110). Desenați trellisul codului obținut. Calculați distanța Hamming minimă și câștigul de codare al acestuia.

P12. Determinați profilul distanțelor și câștigul codului convoluțional cu rată 1:2, C(11;20) (ordine crescătoare a puterilor operatorului) construit în GF(4).