Lucrarea 5

Coduri Hamming grup

5.1 Obiectivul lucrării

În această lucrare, se vor studia codurile Hamming grup şi codurile Hamming grup extinse, principiul de codare şi posibilităţile de detecţie şi corecţie a erorilor.

5.2 Introducere teoretică

5.2.1 Codurile grup

Codurile grup sunt coduri bloc destinate codării surselor de informaţie discrete pentru canale cu perturbaţii, în vederea detecţiei sau corecţiei erorilor; fiecare cuvânt este un vector format din n simboluri binare, altfel spus, are lungimea n. Există 2^n vectori de forma:

$$\mathbf{w} = \left[\begin{array}{cccc} a_1 & a_2 & \dots & a_n \end{array} \right].$$

Dintre aceştia, numai 2^k sunt vectori cu sens (sau *vectori de cod*), care sunt puşi în corespondenţă cu mesajele sursei primare; aceşti vectori de cod alcătuiesc un spaţiu vectorial $\mathbf{V}$ (acest spaţiu are o structură de grup, ceea ce justifică denumirea acestor coduri).

Dintre simbolurile a_i ale unui cuvânt de cod, k sunt alese prin punerea în corespondenţă cu mesajele sursei primare şi se numesc *simboluri de informaţie*; celelalte $m = n - k$ simboluri din cuvântul de cod sunt calculate în funcţie de simbolurile de informaţie şi sunt *simboluri redundante*. Aşadar, operaţia de determinare a cuvintelor de cod se efectuează în două etape: în prima se stabileşte corespondenţa dintre mesajele sursei primare şi un set de vectori de lungime fixă k, iar în etapa a doua se trece de la vectorii având fiecare k simboluri de informaţie, la vectorii de lungime n prin calculul simbolurilor de control. În continuare, se va înţelege prin codare numai a doua etapă, adică calculul cuvintelor de cod $\mathbf{v}$ de lungime n ale codului pornind de la vectorii $\mathbf{i}$ de lungime k formaţi din biţii de informaţie. Acest calcul se poate face cu matricea de control $\mathbf{H}$ sau cu matricea generatoare $\mathbf{G}$.

Matricea de control $\mathbf{H}$ are m linii şi n coloane; un vector $\mathbf{v}$ aparţine codului dacă satisface condiţia:

$$\mathbf{H} \cdot \mathbf{v}^T = 0. \tag{5.1}$$

Această relaţie este echivalentă cu un sistem de m ecuaţii liniare cu ajutorul căruia se pot calcula în mod univoc m necunoscute, adică simbolurile de control.

Matricea generatoare $\mathbf{G}$ are k linii şi n coloane; relaţia de calcul a cuvântului de cod $\mathbf{v}$ pornind de la vectorul de biți de informaţie $\mathbf{i}$ se face cu relaţia:

$$\mathbf{v} = \mathbf{i} \cdot \mathbf{G}. \tag{5.2}$$

La momentul transmisiunii unui cuvânt de cod pe un canal zgomotos, presupus, de obicei, ca fiind binar simetric, o parte din simbolurile vectorului transmis sunt modificate (simbolul 0 se transformă în simbolul 1 şi viceversa) din cauza perturbaţiilor. Matematic, acest proces de eroare este descris prin însumarea vectorului transmis cu un vector de eroare ϵ, de aceeaşi lungime cu vectorul $\mathbf{v}$ transmis, care are unităţile pe poziţiile ce se modifică şi zerouri în rest. Cuvântul recepționat, cu posibile erori, se poate scrie astfel:

$$\mathbf{v_{rec}} = \mathbf{v} + \epsilon. \tag{5.3}$$

De exemplu, dacă

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

este cuvântul de cod transmis de sursă şi

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

este vectorul de eroare, vectorul rezultat la recepţie este

$$\mathbf{v_{rec}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

În cazul detecţiei erorilor, prin operaţia de decodare se determină dacă vectorul recepţionat $\mathbf{v_{rec}}$ aparţine codului $\mathbf{V}$ (adică transmiterea corectă a cuvântului recepţionat prin canal) sau nu. În cazul corecţiei erorilor, decodarea înseamnă determinarea vectorului eroare ϵ şi adunarea lui cu vectorul recepţionat în scopul corectării cuvântului recepționat:

$$\mathbf{v}_c = \mathbf{v_{rec}} + \epsilon. \tag{5.4}$$

Dacă eroarea este determinată în mod corect, $\mathbf{v}_c = v$.

În cazul detecţiei erorilor, se calculează, pentru fiecare vector recepţionat $\mathbf{v_{rec}}$, vectorul corector sau sindrom corespunzător, dat de relaţia:

$$\mathbf{z} = H \cdot \mathbf{v_{rec}}^T = H \cdot (\mathbf{v} + \epsilon)^T = H\mathbf{v}^T + H\epsilon^T = H\epsilon^T. \tag{5.5}$$

cu ajutorul căruia se ia decizia relativ la corectitudinea transmisiei:

- dacă $\mathbf{z} = 0$, transmisia s-a efectuat corect;
- dacă $\mathbf{z} \neq 0$, transmisia s-a efectuat eronat.

În cazul corecţiei erorilor, se mai parcurge o etapă, anume determinarea vectorilor ϵ, pornind de la vectorii $\mathbf{z}$. Modul de efectuare a acestei operaţii diferă de la un cod la altul.

Distanţa Hamming dintre doi vectori de aceeaşi lungime, este definită ca numărul poziţiilor din cadrul vectorilor pe care se află simboluri diferite.

De exemplu, distanţa Hamming (numită în continuare, simplu, *distanţa*) dintre vectorii $\mathbf{v_1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$ şi $\mathbf{v_2} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ este egală cu 2.

Ponderea Hamming (denumită, în mod curent, *ponderea*) unui cuvânt de cod este numărul de unităţi din cadrul acelui cuvânt de cod.

Dacă, în canalul de transmisie, asupra unui cuvânt de cod transmis $\mathbf{v}$ acţionează un cuvânt de eroare de pondere e (adică, vom avea e erori cauzate de zgomotul de pe canal), distanţa dintre

cuvântul recepţionat $\mathbf{v_{rec}}$ şi cuvântul initial $\mathbf{v}$ este egală cu e. Pentru ca vectorul $\mathbf{v_{rec}}$ să poată fi detectat ca eronat este, evident, obligatoriu ca el să nu aparţină codului; cu alte cuvinte, distanţa minimă dintre două cuvinte de cod ale unui cod detector de e erori trebuie să fie mai mare decât e, sau altfel spus, pentru cuvintele acestui cod se impune condiţia:

$$d_{min} = e + 1. \tag{5.6}$$

În cazul corecţiei erorilor, un cuvânt recepţionat eronat este interpretat ca fiind cuvântul de cod situat la distanţa cea mai mică de acesta. Prin urmare, distanţa dintre două cuvinte ale unui cod corector de e erori trebuie să fie mai mare decât $2e$, altfel spus, pentru acest cod:

$$d_{min} = 2e + 1. \tag{5.7}$$

Codurile care pot corecta e erori în orice poziţie, dar nu pot corecta nicio configuraţie particulară de $e + 1$ erori sau mai multe, se numesc *coduri perfecte*.

5.2.2 Codurile Hamming corectoare de o eroare

Codurile Hamming corectoare de o eroare reprezintă generalizarea codului Hamming H(7,4), introdus de către Richard Hamming în anul 1950.

Pentru orice cod Hamming, lungimea cuvintelor de cod este determinată de numărul simbolurilor de control, conform relaţiei:

$$n = 2^m - 1, \tag{5.8}$$

de unde rezultă că numărul simbolurilor de informaţie dintr-un cuvânt de cod este $k = 2^m - m - 1$. Primele 3 coduri din această clasă au parametri:

$$\begin{array}{lll} m = 3 & n = 7 & k = 4 \\ m = 4 & n = 15 & k = 11 \\ m = 5 & n = 31 & k = 26 \end{array} \tag{5.9}$$

Un cod Hamming, fiind un cod grup, este complet determinat de matricea de control, care poate fi scrisă în formă necanonică sau canonică. În cazul scrierii sub formă necanonică, coloanele matricei H sunt reprezentările în binar ale numerelor naturale de la 1 la $2^m - 1$. Pentru cazul $m = 3$, matricea de control necanonică are forma:

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{5.10}$$

Dintre proprietăţile codurilor Hamming, menţionăm:

1. sunt coduri corectoare de o eroare într-un cuvânt de cod. Într-adevăr, oricare două coloane însumate dau un vector coloană nenul. În plus, distanţa minimă între cuvintele de cod este 3 şi, conform relaţiei (5.7), numărul de erori corectabile este egal cu 1.

2. sunt coduri perfecte, deoarece corectează orice vector eroare de pondere 1, dar niciun vector eroare de pondere 2 sau mai mare.

   Pentru a demonstra această proprietate, este suficient a arăta că numărul vectorilor corectori $\mathbf{z}\,(2^m - 1)$ este egal cu numărul vectorilor eroare de pondere 1 (adică, n). Într-adevăr, trecând ultima coloană (a n-a) din binar în zecimal, obţinem $n = 1 + 2^1 + ... + 2^{m-1} = 2^m - 1$.

În cazul scrierii sub formă necanonică a matricei **H**, simbolurile de informaţie şi de control sunt amestecate în cuvântul de cod:

1. poziţiile coloanelor ce conţin o singură valoare de 1 (numărul coloanei este o putere a lui 2) vor fi poziţiile simbolurilor de control în cuvântul de cod rezultat;
2. celelalte poziții, în număr de $k = n - m$, corespund simbolurilor de informaţie.

Exemplu: Pentru codul Hamming $H(7, 4)$, matricea de control are următoarea formă:

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{5.11}$$

Coloanele ce conţin o singură valoare egală cu 1 sunt pe poziţiile 1, 2 şi 4. Acestea vor fi pozţiile simbolurilor de control în cuvântul de cod format.

Pentru a obţine cuvântul de cod, folosind matricea de control **H**, se foloseşte următoarea relaţie:

$$\mathbf{H} \cdot \mathbf{v}^T = 0, \tag{5.12}$$

unde **v** este cuvântul de cod rezultat. Pentru matricea **H** din relaţia (5.16), relaţia (5.12) devine:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ i_3 \\ c_4 \\ i_5 \\ i_6 \\ i_7 \end{bmatrix} = 0. \tag{5.13}$$

În acest exemplu, vectorul de biți de informație are $k = 4$ simboluri:

$$\mathbf{i} = \begin{bmatrix} i_3 & i_5 & i_6 & i_7 \end{bmatrix}$$

Din relaţia (5.13), rezultă:

$$\begin{bmatrix} c_4 + i_5 + i_6 + i_7 \\ c_2 + i_3 + i_6 + i_7 \\ c_1 + i_3 + i_5 + i_7 \end{bmatrix} = 0. \tag{5.14}$$

Relaţia anterioară poate fi scrisă sub forma unui sistem de ecuaţii:

$$\begin{cases} c_4 + i_5 + i_6 + i_7 = 0 \\ c_2 + i_3 + i_6 + i_7 = 0 \\ c_1 + i_3 + i_5 + i_7 = 0 \end{cases} \Rightarrow \begin{cases} c_4 = i_5 + i_6 + i_7 \\ c_2 = i_3 + i_6 + i_7 \\ c_1 = i_3 + i_5 + i_7 \end{cases} \tag{5.15}$$

Cuvântul de cod poate fi obţinut şi cu ajutorul matricei generatoare **G**, folosind relaţia (5.2) și matricea generatoare:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{5.16}$$

Efectuând operația de înmulțire, va rezulta:

$$\mathbf{v} = \mathbf{i} \cdot \mathbf{G} = \left[\begin{array}{ccccccc} i_3 + i_5 + i_7 & i_3 + i_6 + i_7 & i_3 & i_5 + i_6 + i_7 & i_5 & i_6 & i_7 \end{array} \right]$$

Ținând cont de ordinea biților de control, se verifică astfel relațiile de mai sus.

Se poate observa din structura cuvintelor de cod (5.13), că simbolurile de informaţie sunt amestecate cu cele de control. Pentru a optimiza procesul, se preferă utilizarea formei sistematice a codului, în care simbolurile de informaţie şi cele de control sunt grupate în structura cuvântului de cod. Pentru a se obţine cuvinte de cod în formă sistematică, matricea de control şi cea generatoare se scriu sub formă canonică.

Dacă presupunem că biții de control se grupează înaintea biților de informație, $\mathbf{v}_s = \begin{bmatrix} \mathbf{c} & \mathbf{i} \end{bmatrix}$ ($\mathbf{i}$ este vectorul biților de informație, $\mathbf{c}$ este vectorul biților de control), matricele de control și generatoare vor fi:

$$\mathbf{H}_c = \left[\begin{array}{cc} \mathbf{I}_m & \mathbf{Q} \end{array} \right] \qquad \mathbf{G}_c = \left[\begin{array}{cc} \mathbf{Q}^T & \mathbf{I}_k \end{array} \right] \tag{5.17}$$

în care $\mathbf{I}_m$ este matricea identitate $m \times m$, iar $\mathbf{I}_k$ este matricea identitate $k \times k$.

În practică, pentru un anumit cod, forma canonică a matricii de control se obține din matricea de control H în care se reordonează coloanele în funcție de poziția biților de control.

Folosind (5.17), relația de codare

$$\mathbf{H}_c \cdot \mathbf{v}_s^T = 0 \tag{5.18}$$

devine:

$$\mathbf{c} = \mathbf{i} \cdot \mathbf{Q}^T \tag{5.19}$$

Egalitatea furnizează o relație directă între biții de control și cei de informație.

Exemplu: Pentru codul Hamming $H(7,4)$, forma canonică a matricelor de control şi generatoare va fi:

$$\mathbf{H}_c = \left[\begin{array}{ccccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array} \right] \qquad \mathbf{G}_c = \left[\begin{array}{ccccccc} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right].$$

Pentru a păstra relațiile de codare (5.15) și a respecta reordonarea coloanelor între $\mathbf{H}$ și $\mathbf{H}_c$, vom considera: $\mathbf{v}_s = \begin{bmatrix} c_4 & c_2 & c_1 & i_3 & i_5 & i_6 & i_7 \end{bmatrix}$.

În continuare, vom determina cuvintele de cod pentru vectorul de biți de informaţie $\mathbf{i} = \left[\begin{array}{cccc} 0 & 1 & 1 & 0 \end{array} \right]$, în formă nesistematică şi sistematică.

a) *Folosirea matricei de control sub formă necanonică*

Se poate folosi sistemul de ecuaţii (5.15):

$$\begin{cases} c_4 = i_5 + i_6 + i_7 = 1 + 1 + 0 = 0 \\ c_2 = i_3 + i_6 + i_7 = 0 + 1 + 0 = 1 \\ c_1 = i_3 + i_5 + i_7 = 0 + 1 + 0 = 1 \end{cases}$$

Rezultă cuvântul de cod:

$$\mathbf{v} = \left[\begin{array}{ccccccc} c_1 & c_2 & i_3 & c_4 & i_5 & i_6 & i_7 \end{array} \right] = \left[\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{array} \right].$$

b) *Folosirea matricei generatoare sub formă necanonică*

Se va utiliza relaţia (5.2):

$$\mathbf{v} = \mathbf{i} \cdot \mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

adică:

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

obținându-se, așa cum era de așteptat, rezultatul de mai sus.

c) *Folosirea formei canonice a matricei de control*

Vom considera că biții de control sunt plasați înaintea biților de informație în ordinea menționată mai sus:

$$\mathbf{v}_s = \begin{bmatrix} c_4 & c_2 & c_1 & i_3 & i_5 & i_6 & i_7 \end{bmatrix}$$

Se va utiliza relaţia (5.12):

$$\mathbf{H}_c \cdot \mathbf{v}_s^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_4 \\ c_2 \\ c_1 \\ i_3 \\ i_5 \\ i_6 \\ i_7 \end{bmatrix} = 0$$

şi vor rezulta aceleași relații de codare ca și în cazul nesistematic, însă ordinea biților în cuvântul de cod va fi schimbată:

$$\begin{cases} c_4 = i_5 + i_6 + i_7 = 1 + 1 + 0 = 0 \\ c_2 = i_3 + i_6 + i_7 = 0 + 1 + 0 = 1 \\ c_1 = i_4 + i_5 + i_7 = 0 + 1 + 0 = 1 \end{cases}$$

Cuvântul de cod astfel obţinut va fi:

$$\mathbf{v}_s = \begin{bmatrix} c_4 & c_2 & c_1 & i_3 & i_5 & i_6 & i_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

d) *Folosirea formei canonice a matricei generatoare*

Se va utiliza relaţia (5.2):

$$\mathbf{v}_s = \mathbf{i} \cdot \mathbf{G}_c = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

adică:

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

Se observă că între forma sistematică şi cea nesistematică, diferă doar poziţiile simbolurilor de control în interiorul cuvântului de cod.

Cuvintele de cod obţinute prin codarea cu matricea **H** si **G** sunt identice. Distanţa Hamming minimă a codului este, în ambele situaţii, egală cu 3 şi de aceea, pot fi detectate două erori sau poate fi corectată una singură.

Decodarea cuvintelor de cod se face cu ajutorul matricei de control **H**, care trebuie sa fie cunoscută atât la transmisie cât şi la recepţie.

Dacă transmisia s-a realizat fără erori, atunci sindromul va fi egal cu 0. Daca au apărut erori, el va fi un vector coloană diferit de 0.

Dacă decodorul este setat doar pentru detecţia erorilor, codul Hamming poate sa detecteze o eroare dublă (DED – Double Error Detection). Dacă este setat pentru corecţia erorilor, atunci poate să corecteze o singură eroare (SEC – Single Error Correction).

În cazul detecţiei erorilor, orice valoare a sindromului diferită de 0, va semnala existenţa uneia sau mai multor erori si se va cere retransmisia mesajului.

În cazul corecţiei, sindromul este egal cu coloana din matricea de control **H** care indică poziţia erorii în cuvântul recepţionat.

Exemplu: Fie cuvântul de cod obţinut cu ajutorul matricei **H** sub formă necanonică:

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

şi cuvântul eroare:

$$\epsilon = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Cuvântul de cod recepţionat este:

$$\mathbf{v}_{rec} = \mathbf{v} + \epsilon = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Se determină sindromul cu ajutorul relaţiei (5.5):

$$\mathbf{z} = \mathbf{H} \cdot \mathbf{v}_{rec}^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Faptul că au fost introduse erori în canal este dovedit de valoarea diferită de zero a corectorului (detecţia erorilor). Pentru corecţia erorilor, se ţine seama de faptul că sindromul **z** este egal cu a treia coloană a matricei **H** necanonice, coloană care este reprezentarea în binar a numărului 3, poziţia în cuvântul de cod a simbolului eronat. Acest lucru se poate verifica folosind relația de mai jos, în care s-a considerat un cuvânt-eroare ϵ ce are 1 pe poziția a treia:

$$\mathbf{z} = \mathbf{H} \cdot \epsilon^{\mathbf{T}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

Dacă se transmite cuvântul de cod în formă sistematică:

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

iar vectorul de eroare este acelaşi:

$$\epsilon = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

sindromul va avea expresia:

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

Sindromul este egal cu coloana a treia a matricei de control în formă canonică, dar nu mai este reprezentarea în binar a numărului 3.

5.2.3 Coduri Hamming extinse

Codurile Hamming sunt, prin definiţie, coduri corectoare de o eroare (SEC) sau detectoare de două erori (DED). Pentru a extinde posibilităţile de detecţie si corecţie, se poate forma codul Hamming extins, prin adăugarea unui bit suplimentar de control, numit şi *bit de paritate*.

Aceast bit de paritate va transfoma codul Hamming din corector de o eroare sau detector de două erori în corector de o eroare şi detector de două erori (SEC-DED). Acest lucru este posibil, crescând distanţa minimă între cuvintele de cod la $d = 4$.

Matricea de control a codurilor extinse, notată $\overline{\mathbf{H}}$ se obţine din matricea codului Hamming nesistematic în felul următor:

$$\overline{\mathbf{H}} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ 1 & 1 \end{bmatrix}. \tag{5.20}$$

Exemplu: Pentru codul $H(7,4)$, matricea de control a codului extins $\overline{H}(8,4)$ va avea urmatoarea formă:

$$\overline{\mathbf{H}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{5.21}$$

În cazul în care un singur bit este eronat, sindromul $\mathbf{z}$ va fi egal cu coloana din $\overline{\mathbf{H}}$ corespunzătoare poziției eronate. Se observă că, în acest caz, sindromul va avea ultimul element egal cu 1 (toate coloanele se termină în 1). În cazul în care doi biți sunt eronați, sindromul va fi diferit de zero, dar ultimul său element va fi 0. Acest lucru se poate verifica adunând oricare două coloane din $\overline{\mathbf{H}}$.

Cuvântul de cod va avea forma:

$$\mathbf{v} = \begin{bmatrix} c_1 & c_2 & i_3 & c_4 & i_5 & i_6 & i_7 & c_8 \end{bmatrix} \tag{5.22}$$

Folosind relația de codare:

$$\overline{\mathbf{H}} \cdot \mathbf{v}^T = 0 \tag{5.23}$$

obținem:

$$\begin{aligned} c_4 &= i_5 + i_6 + i_7 \\ c_2 &= i_3 + i_6 + i_7 \\ c_1 &= i_3 + i_5 + i_7 \\ c_8 &= c_1 + c_2 + i_3 + c_4 + i_5 + i_6 + i_7 = i_3 + i_5 + i_6 \end{aligned}$$

adică:

$$\mathbf{v} = \begin{bmatrix} i_3 + i_5 + i_7 & i_3 + i_6 + i_7 & i_3 & i_5 + i_6 + i_7 & i_5 & i_6 & i_7 & i_3 + i_5 + i_6 \end{bmatrix} \tag{5.24}$$

Matricea generatoare corespunzătoare este:

$$\overline{\mathbf{G}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{5.25}$$

Se poate verifica ușor că $\mathbf{v} = \mathbf{i} \cdot \overline{\mathbf{G}}$, unde $\mathbf{i} = \begin{bmatrix} i_3 & i_5 & i_6 & i_7 \end{bmatrix}$.

Pentru a putea deduce forma canonică a matricii de control corespunzătoare codului $\overline{H}(8,4)$ nu vom mai putea folosi, în mod direct, faptul că primele coloane corespund matricii identitate $\mathbf{I}_4$ (ce corespund biților de control), iar restul sunt coloanele rămase din matricea de control (ce corespund biților de informație) deoarece ultima linie din matricea $\overline{\mathbf{H}}$ are toate elementele egale cu 1. Pentru a păstra similitudinea cu matricea canonică corespunzătoare codului $H(7,4)$, pentru codul extins, se va considera cuvântul de cod în formă sistematică:

$$\mathbf{v}_s = \begin{bmatrix} c_4 & c_2 & c_1 & c_8 & i_3 & i_5 & i_6 & c_7 \end{bmatrix} \tag{5.26}$$

iar forma canonică a matricii de control corespunzătoare codului $\overline{H}(8,4)$ este:

$$\overline{\mathbf{H}}_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{5.27}$$

Se poate verifica că $\overline{\mathbf{H}}_c \cdot \mathbf{v}_s^T = 0$ conduce la aceleași relații de codare ca mai sus.

Conform relaţiilor (5.17), forma canonică a matricii generatoare este:

$$\overline{\mathbf{G}}_c = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.28}$$

Pentru decodarea codurilor extinse, se calculează sindromul, conform relaţiei (5.5). Dacă acesta este egal cu 0, mesajul a fost transmis fără eroare. Dacă este diferit de 0 şi se află printre coloanele matricei de control extinse, atunci a apărut o eroare ce poate fi corectată. Dacă sindromul are o valoare diferită de 0, dar nu se regăseşte printre coloanele matricei de control $\overline{\mathbf{H}}$, atunci au apărut două erori detectabile. În acest ultim caz, sindromul are ultimul element egal cu 0.

5.3 Desfăşurarea lucrării

În continuare se vor exemplifica capacităţile de detecţie şi corecţie a erorilor ale codurilor grup, în particular codurile Hamming şi codurile Hamming extinse.

Aplicaţia va consta în mai multe etape, precum cele descrise mai jos:

1. Formarea matricelor de control şi generatoare pentru codurile grup;
2. Utilizarea matricelor obţinute la pasul anterior pentru codarea mesajelor în formă nesistematică şi sistematică;
3. Transmiterea cuvântului de cod printr-un canal cu erori;
4. Calcularea sindromului (sau, corectorului);
5. Verificarea corectitudinii operaţiei de decodare/detecţie a erorilor.

În continuare va fi prezentată o aplicație de codare, detecție, respectiv corecție a erorilor, folosind un cod Hamming grup H(7,4), în varianta nesistematică. Așa cum se poate observa din exemplu, acest cod, poate corecta orice configurație de o eroare (din cele 7 posibile), dar nu poate corecta 2 sau mai multe erori. La final, în urma decodării, se va putea recupera mesajul transmis de sursă prin extragerea biților de informație din cuvântul corectat.

```
import numpy as np
import random

class codH74():
    def __init__(self):
        self.n = 7
        self.k = 4
        self.m = self.n - self.k
        self.H = np.array([[0, 0, 0, 1, 1, 1, 1], [0, 1, 1, 0, 0, 1, 1], [1, ...
            0, 1, 0, 1, 0, 1]])

    def __repr__(self):
        return 'Matricea de control pentru codul H(7,4):\n ...
            {}'.format(repr(self.H))

    def codare(self, i):
        [i3, i5, i6, i7] = i
        c1 = (i3+i5+i7) % 2
        c2 = (i3+i6+i7) % 2
        c4 = (i5+i6+i7) % 2
        return np.array([c1, c2, i3, c4, i5, i6, i7])

    def detectie_erori(self, v_rec):
        z = np.mod(np.matmul(self.H, v_rec), 2)
        return (z == np.zeros(shape=(self.m,))).all()

    def corectie_erori(self, v_rec):
        z = np.mod(np.matmul(self.H, v_rec), 2)
        if (z == np.zeros(shape=(self.m,))).all():
            return v_rec
        else:
            cuvant_eroare_estimat = np.zeros(shape=(self.n, ), dtype=int)
            # OBSERVATIE!
            # La codurile Hamming nesistematice, z reprezinta pozitia eronata ...
                in binar
```

```
            # Pentru restul codurilor, se identifica pozitia eronata prin ...
                identificarea coloanei din H egala cu z
            pozitie_eroare_estimata = bin2dec(z)
            print('Eroarea se produsese pe pozitia: ...
                {}'.format(pozitie_eroare_estimata))
            cuvant_eroare_estimat[pozitie_eroare_estimata-1] = 1
            v_corectat = np.mod(v_rec + cuvant_eroare_estimat, 2)
            return v_corectat


def bin2dec(a):
    b = 0
    for p in range(len(a)):
        b += a[-p-1] * (2**p)
    return np.int(b)


def transmitere_canal(v, e):
    cuvant_eroare = np.zeros(shape=(len(v),), dtype=int)  # initial
    pozitie_eronata = random.sample(range(len(v)), k=e)
    cuvant_eroare[pozitie_eronata] = 1
    v_rec = np.mod(v + cuvant_eroare, 2)  # o pozitie aleasa aleator este eronata
    return v_rec

cod = codH74()
print(cod)

# Codare
print('\nCODARE')
i = [0, 1, 1, 0]
v = cod.codare(i)
print('Cuvantul de cod pentru {} este {}'.format(i, v))

# (a) Transmitere printr-un canal cu zgomot (1 eroare)
# random.seed(32)
print('\nCAZUL (a)')
e = 1
print('\nPe canal se transmite {}'.format(v))
v_rec = transmitere_canal(v, e)
print('La receptor, ajunge cuvantul {}'.format(v_rec))
print('Exista erori? {}'.format(~cod.detectie_erori(v_rec)))
v_corectat = cod.corectie_erori(v_rec)
print('Cuvantul corectat este: {}'.format(v_corectat))
print('Este cuvantul corectat identic cu cuvantul de cod transmis? ...
    {}'.format((v_corectat == v).all()))
# Decodare
i_decodare = [v_corectat[2], v_corectat[4], v_corectat[5], v_corectat[6]]
print('Mesajul recuperat: {}'.format(i_decodare))

# (b) Transmitere printr-un canal cu zgomot (2 erori)
print('\nCAZUL (b)')
print('\nPe canal se transmite {}'.format(v))
e = 2
v_rec = transmitere_canal(v, e)
print('La receptor, ajunge cuvantul {}'.format(v_rec))
print('Exista erori? {}'.format(~cod.detectie_erori(v_rec)))
v_corectat = cod.corectie_erori(v_rec)
print('Cuvantul corectat este: {}'.format(v_corectat))
print('Este cuvantul corectat identic cu cuvantul de cod transmis? ...
    {}'.format((v_corectat == v).all()))
```

5.4 Exerciţii

Pornind de la algoritmul prezentat în platformă, rezolvați următoarele exercitii:

1. Implementați funcțiile:

   h31_ encode(text: str) -> np.ndarray
   h31_ decode(code_ matrix: np.ndarray) -> str

   Funcția *h31_ encode* primește ca parametru un text format doar din caracterele '0', '1' și returnează o matrice în care fiecare linie reprezintă codarea caracterului respectiv folosind un cod Hamming(3, 1).

   Funcția *h31_ decode* primește ca parametru o matrice în care fiecare linie reprezintă codarea unui caracter folosind un cod Hamming(3, 1) (codarea este afectată de cel mult 1 eroare) și întoarce textul decodat.

2. Implementați funcțiile:

   h84_ encode(text: str) -> np.ndarray
   h84_ decode(code_ matrix: np.ndarray) -> str

   Funcția *h84_ encode* primește ca parametru un text format doar din primele 16 litere mari ale alfabetului englezesc (de la 'A' la 'P') și returnează o matrice în care fiecare linie reprezintă codarea literei respective folosind un cod Hamming(8, 4).

   Funcția *h84_ decode* primește ca parametru o matrice în care fiecare linie reprezintă codarea unui caracter folosind un cod Hamming(8, 4) (codarea este afectată de cel mult 2 erori) și întoarce textul decodat. Dacă un cod este afectat de 2 erori, se va decoda în caracterul '*'.