CAPITOLUL 9

CODURI CONVOLUȚIONALE

9.1. CODAREA CODURILOR CONVOLUȚIONALE

Ca şi în cazul codării bloc, şirul biților generați de sursa de informație se segmentează în blocuri de câte k biți, iar la ieşirea codorului se formează blocuri de câte n biți; spre deosebire, însă, de codarea bloc, cei n biți de ieşire, care împreună constituie un *cadru*, nu sunt determinați numai de cei k biți de intrare, ci şi de un număr de variabile de stare. Un codor convoluțional poate fi considerat drept o *maşină automată cu stări finite*. În funcție de starea internă, unul şi acelaşi bloc de mesaj de k biți va determina la ieşire cadre diferite. Cuvântul de cod este şirul continuu de biți de ieşire de la începutul şi până la sfârşitul unei transmisiuni. Un exemplu de codor convoluțional este dat în figura 9.1.

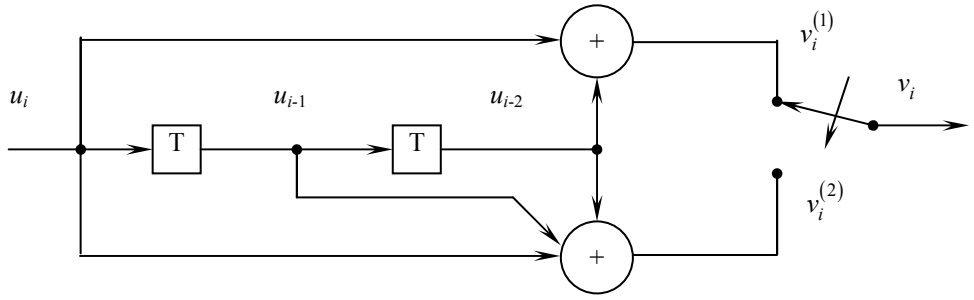

Fig. 9.1. Codor convoluțional de rată ½ şi memorie $\nu = 2$.

După cum se vede din figura 9.1, acest codor convoluțional are schema compusă dintr-un registru de deplasare cu două etaje, două porți

logice SAU EXCLUSIV şi un multiplexor de doi biți. La fiecare T secunde, un nou bit u_i se prezintă la intrarea codorului. Relația dintre timpul continuu t şi timpul discret i este dată de $t_i = iT$. Starea internă a codorului este determinată de valorile biților u_{i-1} şi u_{i-2}, biți care au fost la intrarea codorului cu T, respectiv cu $2T$ secunde mai înainte.

Având în vedere că perechea biților de stare u_{i-1}, u_{i-2} poate lua patru valori 00, 01, 10 şi 11, codorul acesta are $2^2 = 4$ stări. Codorul trebuie deci să memoreze doi biți de informație anteriori. Spunem că memoria codorului este ν = 2. La timpul discret i, codorul scoate la ieşire doi biți, notați cu $v_i^{(1)}$ şi $v_i^{(2)}$, la valorile acestora contribuind atât bitul de intrare u_i cât şi biții de stare internă u_{i-1} şi u_{i-2}. Biții $v_i^{(1)}$ şi $v_i^{(2)}$, care formează *cadrul* de ieşire (nu cuvântul de cod !), se calculează conform cu ecuațiile :

$$v_i^{(1)} = u_i + u_{i-2} \tag{9.1a}$$

$$v_i^{(2)} = u_i + u_{i-1} + u_{i-2}. \tag{9.1b}$$

În (9.1), cu „+“ am notat adunarea modulo 2. La ieşire, rezultă două şiruri $\cdots v_{i-1}^{(1)}, v_i^{(1)}, v_{i+1}^{(1)} \cdots$ şi $\cdots v_{i-1}^{(2)}, v_i^{(2)}, v_{i+1}^{(2)} \cdots$ care se multiplexează cu un comutator pentru a forma un singur şir de cod

$$\mathbf{v} = (\cdots v_{i-1}^{(1)}, v_{i-1}^{(2)}, v_i^{(1)}, v_i^{(2)}, v_{i+1}^{(1)}, v_{i+1}^{(2)} \cdots) \tag{9.2}$$

Utilizăm acest exemplu simplu de cod convoluțional pentru a introduce noțiuni care sunt valabile şi pentru scheme mai complicate. Astfel, vom descrie conexiunile dintre etajele registrului de deplasare şi sumatoarele modulo 2 cu ajutorul următoarelor două şiruri generatoare

$$\begin{aligned} \mathbf{g}^{(1)} &= (g_0^{(1)}, g_1^{(1)}, g_2^{(1)}) = (101) \\ \mathbf{g}^{(2)} &= (g_0^{(2)}, g_1^{(2)}, g_2^{(2)}) = (111) \end{aligned} \tag{9.3}$$

În (9.3), $\mathbf{g}^{(1)}$ reprezintă conexiunile de sus, iar $\mathbf{g}^{(2)}$ pe cele de jos, $g_0^{(1)}$ si $g_0^{(2)}$ fiind conexiunile cele mai din stânga. Un 1 reprezintă conexiune, un 0, lipsa de conexiune. Termenul de *cod convoluțional* se explică acum observând că şirul de ieşire $\mathbf{v}^{(l)}$, l = 1, 2, reprezintă convoluția şirului de intrare **u** cu şirul generator $\mathbf{g}^{(l)}$:

$$\mathbf{v}^{(l)} = \mathbf{u} * \mathbf{g}^{(l)}, \; l = 1, 2. \tag{9.4}$$

În (9.4), am notat cu $*$ operatorul de convoluție.

EXEMPLUL 9.1: Să presupunem că şirul de intrare este

$$\mathbf{u} = (1011100\cdots) \tag{9.5}$$

Cele două şiruri de ieşire sunt :

$$\mathbf{v}^{(1)} = (1001011\cdots) \tag{9.6a}$$

$$\mathbf{v}^{(2)} = (1100101\cdots) \tag{9.6b}$$

Şirul de cod emis este :

$$\mathbf{v} = (11,01,00,10,01,10,11,\cdots) \tag{9.7}$$

Şirurile generatoare (9.3) pot fi considerate drept „răspunsuri la impuls“, obținute aplicând la intrare şirul particular **u** = (1000···) şi observând cele două şiruri de ieşire $\mathbf{v}^{(1)}$ şi $\mathbf{v}^{(2)}$. Întrucât memoria acestui codor este $\nu = 2$, răspunsurile la impuls pot dura cel mult $\nu + 1 = 3$ unități de timp.

Să considerăm acum un codor convoluțional ceva mai complicat. Schema din fig.9.2 este asemănătoare cu cea din fig.9.1; ea are un etaj în plus, astfel încât memoria $\nu = 3$, iar numărul stărilor interne este $2^3 = 8$.

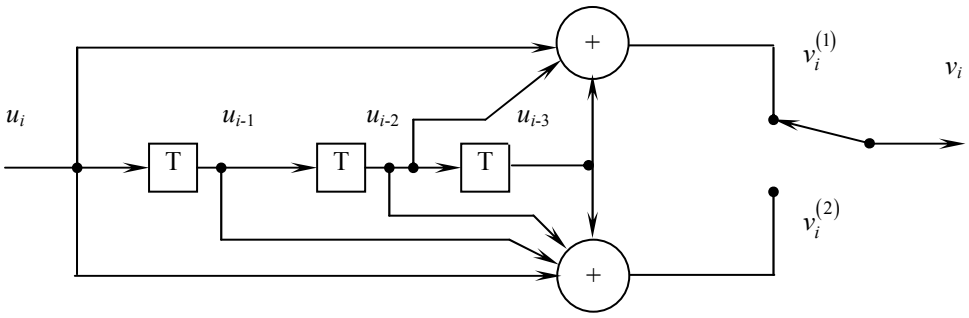

Fig. 9.2. Codor convoluțional de rată ½ şi memorie $\nu = 3$.

Biții de ieşire se calculează conform cu ecuațiile:

$$\begin{aligned} v_i^{(1)} &= u_i + u_{i-2} + u_{i-3} \\ v_i^{(2)} &= u_i + u_{i-1} + u_{i-2} + u_{i-3} \end{aligned} \tag{9.8}$$

Şirurile generatoare sunt date de:

$$\begin{aligned} \mathbf{g}^{(1)} &= (g_0^{(1)}, g_1^{(1)}, g_2^{(1)}, g_3^{(1)}) = (1011) \\ \mathbf{g}^{(2)} &= (g_0^{(2)}, g_1^{(2)}, g_2^{(2)}, g_3^{(2)}) = (1111) \end{aligned} \tag{9.9}$$

Vom vedea că, având memorie mai mare (sau echivalent, mai multe stări interne), codorul din figura 9.2 este superior ca performanță celui din figura 9.1.

În general, codul are k şiruri de intrare şi n şiruri de ieşire. Rata codului, la fel ca pentru codurile bloc, este definită de raportul $R = k/n$. Notăm concis un cod convoluțional cu (n,k), dar numai aceşti parameti, n şi k, nu-l carecterizează univoc. Un exemplu de codor convoluțional (3,2) este dat în figura 9.3.

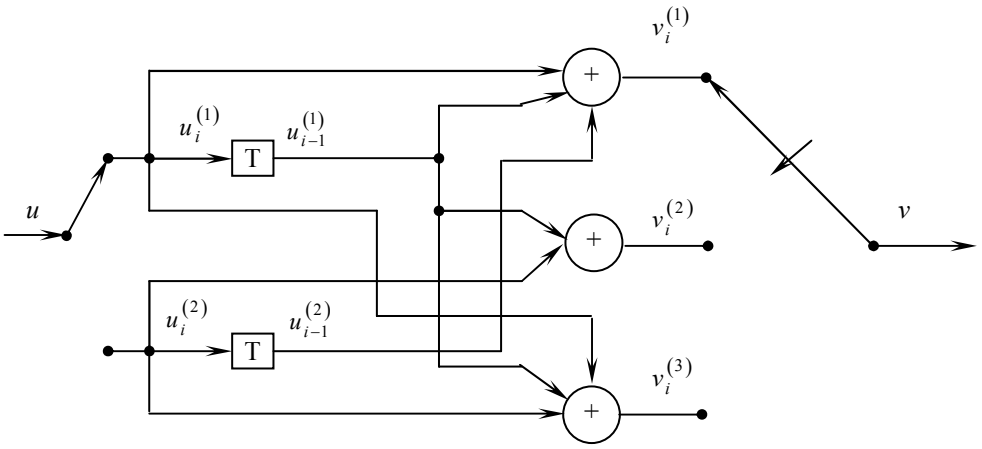

Fig. 9.3 Codor convoluțional de rată 2/3 şi memorie totală de $\nu = 2$.

Deoarece $k = 2$, codorul din fig. 9.3 constă din două registre de deplasare (de câte un singur etaj, în acest caz), împreună cu $n = 3$ sumatoare modulo 2 (porți logice SAU EXCLUSIV), un demultiplexor de intrare şi un muliplexor de ieşire. Biții de informație intră în codor câte $k = 2$ la un moment anume i şi putem scrie aceasta în două feluri :

1. ca un şir de intrare

$$\mathbf{u} = (u_0^{(1)}u_0^{(2)}, u_1^{(1)}u_1^{(2)}, u_2^{(1)}u_2^{(2)}, \cdots)$$

2. sau ca două şiruri de intrare

$$\mathbf{u}^{(1)} = (u_0^{(1)}, u_1^{(1)}, u_2^{(1)}, \cdots)$$

$$\mathbf{u}^{(2)} = (u_0^{(2)}, u_1^{(2)}, u_2^{(2)}, \cdots)$$

Există trei şiruri generatoare corespunzătoare fiecărui şir de intrare. Reprezentăm şirul generator corespunzător intrării j şi ieşirii l prin

$$\mathbf{g}_j^{(l)} = (g_{j,0}^{(l)}, g_{j,1}^{(l)}, \cdots, g_{j,\nu_j}^{(l)}). \tag{9.10}$$

Şirurile generatoare pentru codul (3,2) din fig.9.3 sunt:

$$\mathbf{g}_1^{(1)} = (1 \quad 1) \qquad \mathbf{g}_1^{(2)} = (0 \quad 1) \qquad \mathbf{g}_1^{(3)} = (1 \quad 1)$$

$$\mathbf{g}_2^{(1)} = (0 \quad 1) \qquad \mathbf{g}_2^{(2)} = (1 \quad 0) \qquad \mathbf{g}_2^{(3)} = (1 \quad 0).$$

Ecuaţiile de codare se pot scrie:

$$\begin{aligned} \mathbf{v}^{(1)} &= \mathbf{u}^{(1)} * \mathbf{g}_1^{(1)} + \mathbf{u}^{(2)} * \mathbf{g}_2^{(1)} \\ \mathbf{v}^{(2)} &= \mathbf{u}^{(1)} * \mathbf{g}_1^{(2)} + \mathbf{u}^{(2)} * \mathbf{g}_2^{(2)} \\ \mathbf{v}^{(3)} &= \mathbf{u}^{(1)} * \mathbf{g}_1^{(3)} + \mathbf{u}^{(2)} * \mathbf{g}_2^{(3)} \end{aligned} \tag{9.11}$$

Operaţia de convoluţie din (9.11) înseamnă că:

$$\begin{aligned} v_i^{(1)} &= u_i^{(1)} + u_{i-1}^{(1)} + u_{i-1}^{(2)} \\ v_i^{(2)} &= u_i^{(2)} + u_{i-1}^{(1)} \\ v_i^{(3)} &= u_i^{(1)} + u_i^{(2)} + u_{i-1}^{(1)} \end{aligned} \tag{9.12}$$

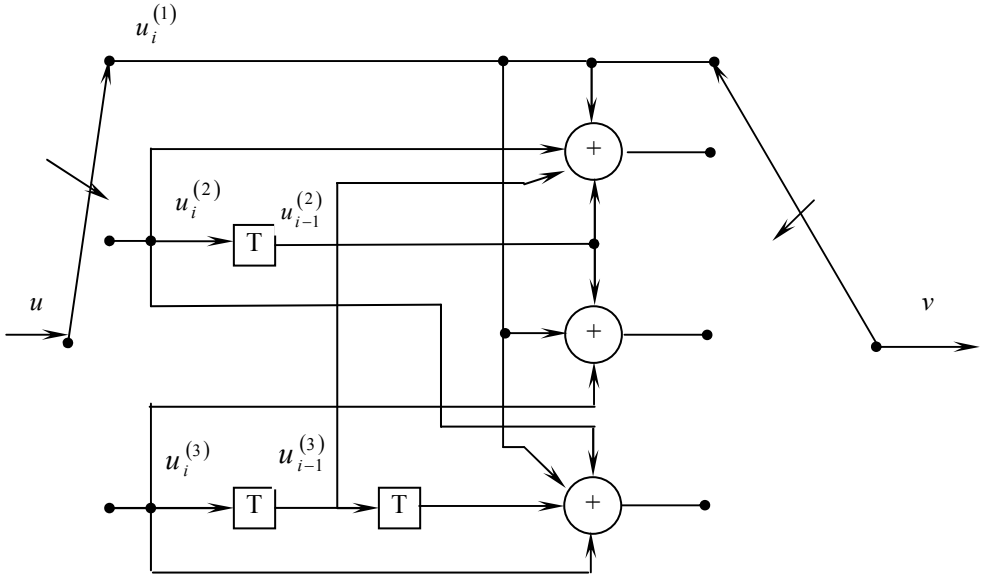

Fig. 9.4. Codor convoluţional de rată ¾ şi memorie totală $\nu = 3$.

Pentru k >1, codorul este alcătuit din k registre de deplasare, care nu este necesar să aibe aceaşi lungime. Numărul de etaje ale registrului de

deplasare l, $1 \le l \le k$, este *memoria* intrării l a codorului. *Memoria* codorului se defineşte drept memoria maximă a intrărilor codorului

$$m = \max_{1 \le l \le k} \{\nu_l\}. \tag{9.13}$$

Memoria totală a codorului este suma memoriilor intrărilor codorului

$$\nu = \sum_{l=1}^{k} \nu_l \tag{9.14}$$

Un exemplu de codor convoluțional (4,3) în care lungimile registrelor de deplasare sunt 0,1 si 2 este dat în figura 9.4.

9.2. REPREZENTAREA ÎN DOMENIUL TRANSFORMATEI *D*

În orice sistem liniar, operațiile efectuate în domeniul timp unde intervine convoluția se pot înlocui cu operații mai convenabile din domeniul transformatei, unde se efectuează produse. În teoria datelor eşentionate, operatorul ce realizează o întârziere de T secunde se notează cu z^{-1}, unde z este o variabilă complexă. Noi vom nota acest operator de întârziere cu D. Fie **u** un şir de biți ce se succed în timp discret:

$$\mathbf{u} = \cdots u_{-1}, u_0, u_1, \cdots, u_i, \cdots \tag{9.15}$$

Transformata D a acestui şir este:

$$u(D) = \cdots + u_{-1}D^{-1} + u_0 + u_1 D + \cdots + u_i D^i + \cdots \tag{9.16}$$

Puterea lui D reprezintă numărul unităților de timp cu care este întârziat bitul în raport cu bitul din şir transmis la timpul discret 0.

EXEMPLUL 9.2: Pentrul codorul convoluțional reprezentat în fig. 9.2, şirurilor generatoare (9.9) le corespund polinoamele în D:

$$\begin{aligned} g^{(1)}(D) &= 1 + D^2 + D^3 \\ g^{(2)}(D) &= 1 + D + D^2 + D^3 \end{aligned} \tag{9.17}$$

Dacă polinomul de mesaj este

$$u(D) = u_0 + u_1 D + u_2 D^2 + \cdots \tag{9.18}$$

iar polinoamele de cod sunt

$$\begin{aligned} v^{(1)}(D) &= v_0^{(1)} + v_1^{(1)} D + v_2^{(1)} D^2 + \cdots \\ v^{(2)}(D) &= v_0^{(2)} + v_1^{(2)} D + v_2^{(2)} D^2 + \cdots \end{aligned} \tag{9.19}$$

ecuațiile de codare (9.4) devin

$$\begin{aligned} v^{(1)}(D) &= u(D) g^{(1)}(D) \\ v^{(2)}(D) &= u(D) g^{(2)}(D). \end{aligned} \tag{9.20}$$

Fie şirul de intrare (9.5), care are polinomul în D următor:

$$u(D) = 1 + D^2 + D^3 + D^4 \tag{9.21}$$

În acest caz particular, ecuațiile de codare (9.20) se scriu:

$$\begin{aligned} v^{(1)}(D) &= (1 + D^2 + D^3 + D^4)(1 + D^2) = 1 + D^3 + D^5 + D^6 \\ v^{(2)}(D) &= (1 + D^2 + D^3 + D^4)(1 + D + D^2) = 1 + D + D^4 + D^6. \end{aligned} \tag{9.22}$$

Rezultatul coincide cu (9.6). Şirul de cod emis este:

$$\begin{aligned} v(D) &= v^{(1)}(D^2) + D v^{(2)}(D^2) \\ &= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13} \end{aligned} \tag{9.23}$$

Acesta coincide cu (9.7).

Orice cod convoluțional (n,k) este specificat de $k \times n$ polinoame generatoare, care formează matricea generatoare $\mathbf{G}(D)$:

$$\mathbf{G}(D) = \begin{bmatrix} g_1^{(1)}(D) & g_1^{(2)}(D) & \cdots & g_1^{(n)}(D) \\ g_2^{(1)}(D) & g_2^{(2)}(D) & \cdots & g_2^{(n)}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_k^{(1)}(D) & g_k^{(2)}(D) & \cdots & g_k^{(n)}(D) \end{bmatrix} \tag{9.24}$$

Să considerăm k şiruri la intrare $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_l, \cdots, \mathbf{u}_k$. Fiecare din ele se poate reprezenta printr-o serie de puteri

$$u^{(l)}(D) = \cdots u_{-1}^{(l)} D^{-1} + u_0^{(l)} + u_1^{(l)} D + u_2^{(l)} D^2 + \cdots \tag{9.25}$$

pentru $l = 1, 2, \cdots, k$. Le putem dispune ca pe un vector linie de şiruri

$$\mathbf{u}(D) = [u^{(1)}(D), u^{(2)}(D), \cdots, u^{(k)}(D)] \tag{9.26}$$

Ieşirea constă dintr-un vector ale cărui componente sunt *n* polinoame de cod

$$\mathbf{v}(D) = [v^{(1)}(D), v^{(2)}(D), \cdots, v^{(n)}(D)]. \tag{9.27}$$

Cu acesta, operația de codare se poate reprezenta printr-un produs vector-matrice:

$$\mathbf{v}(D) = \mathbf{u}(D)\mathbf{G}(D). \tag{9.28}$$

Şirul de cod *j* se calculează astfel:

$$\mathbf{v}^{(j)}(D) = \sum_{l=1}^{k} u^{(l)}(D) g_{lj}(D). \tag{9.29}$$

Memoria intrării *l* a codorului este dată de

$$\nu_l = \max_{1 \le j \le n} \{\text{grad } g_{lj}(D)\}. \tag{9.30}$$

EXEMPLUL 9.3: Să considerăm codorul convoluțional (3,2) reprezentat în fig. 9.3. Matricea generatoare pentru acest codor este:

$$\mathbf{G}(D) = \begin{bmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{bmatrix} \tag{9.31}$$

Fie şirurile la intrare

$$\begin{aligned} u^{(1)}(D) &= 1 + D^2 \\ u^{(2)}(D) &= 1 + D \end{aligned} \tag{9.32}$$

Operația de codare se poate reprezenta astfel :

$$\begin{aligned} v(D) &= [1+D^2 \quad 1+D] \begin{bmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{bmatrix} \\ &= [1+D^3 \quad 1+D^3 \quad D^2+D^3] \end{aligned} \tag{9.33}$$

După multiplexare, întregul şir de cod este dat de :

$$\begin{aligned} v(D) &= v^{(1)}(D^3) + Dv^{(2)}(D^3) + D^2 v^{(3)}(D^3) \\ &= 1 + D + D^8 + D^9 + D^{10} + D^{11} \end{aligned} \tag{9.34}$$

9.3. FORMA SISTEMATICĂ A UNUI COD CONVOLUȚIONAL

Într-un cod convoluțional (*n,k*) *sistematic*, primele *k* şiruri de ieşire sunt identice cu şirurile de intrare. Codurile convoluționale sistematice nu sunt performante decât dacă au reacție. Codurile convoluționale considerate până acum sunt fără reacție: biții de intrare se deplasează în nişte registre de deplasare de la intrare spre ieşire, iar starea codului este dată de coținutul

acestor registre de deplasare la un moment dat. Vom arăta că un codor convoluțional nesistematic fără reacție poate fi pus în formă sistematică prin introducerea unei reacții de la ieşire la intrare.

EXEMPLUL 9.4: Să considerăm codul convoluțional reprezentat în figura 9.1. Biții de ieşire sunt legați de biții de intrare prin ecuațiile (9.1). În formă sistematică, un bit de la ieşire, $v_i^{(1)}$ sau $v_i^{(2)}$, trebuie să fie identic cu bitul de intrare u_i. Totodată, forma şirului de ieşire trebuie să fie aceeaşi ca pentru codul nesistematic. Pentru aceasta, introducem o variabilă auxiliară s_i şi scriem că:

$$\begin{aligned} v_i^{(1)} &= s_i + s_{i-2} \\ v_i^{(2)} &= s_i + s_{i-1} + s_{i-2} \end{aligned} \tag{9.35}$$

Este clar că şirul de ieşire, sau cuvântul de cod, are aceeaşi formă ca şi cea exprimată de (9.1). Bitul de ieşire u_i trebuie să apară nemodificat în cadrul de ieşire $v_i^{(1)}, v_i^{(2)}$. Evident, avem două posibilități ; să le considerăm pe rând.

Dacă $v_i^{(1)} = u_i$, prima ecuație din (9.35) se scrie:

$$s_i = u_i + s_{i-2} \tag{9.36}$$

Aceasta se traduce imediat în schema de codor convoluțional sistematic din fig.9.5.

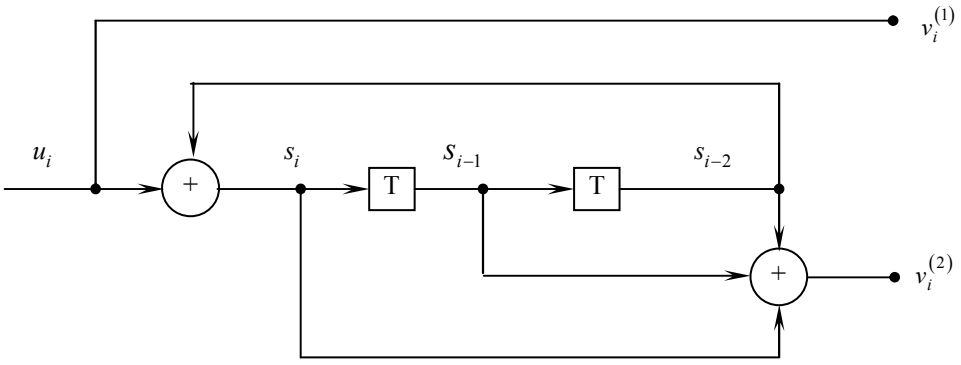

Fig. 9.5. Codor convoluțional sistematic corespunzător codorului nesistematic din fig.9.1.

Avem de făcut o observație importantă : deşi cele două codoare generează acelaşi cod, şirurile de intrare sunt diferite. Acest aspect este secundar, căci relația de intrare-ieşire este biunivocă în ambele cazuri.

Dacă însă $v_i^{(2)} = u_i$, ecuația a doua din (9.35) se scrie:

$$s_i = u_i + s_{i-1} + s_{i-2} \tag{9.37}$$

Conform acestei ecuații, se desenează schema din fig.9.6.

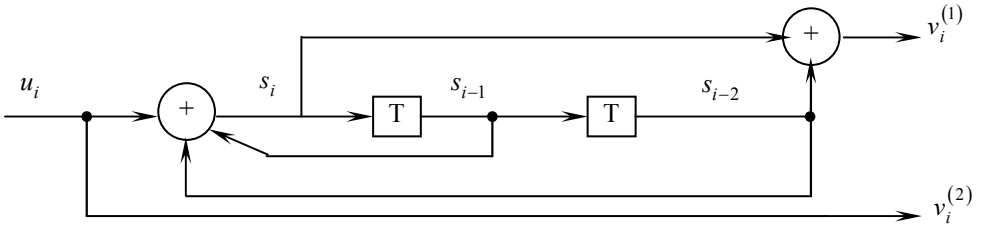

Fig. 9.6. A doua variantă de codor convoluțional sistematic corespunzător codorului nesistematic din fig .9.1.

Rămânând la exemplul 9.4, să arătăm acum că putem transforma un codor nesistematic într-unul sistematic prin operații elementare cu liniile matricei generatoare din domeniul transformatei D. Pentru codorul convoluțional din fig. 9.1, matricea generatoare este:

$$\mathbf{G}(D) = \left[1 + D^2 \quad 1 + D + D^2\right] \tag{9.38}$$

Cuvântul de cod generat de acest codor este dat de:

$$\begin{aligned} v(D) &= u(D)\left[1 + D^2 \quad 1 + D + D^2\right] \\ &= u(D)(1 + D^2)\left[1 \quad \frac{1 + D + D^2}{1 + D^2}\right] \\ &= u'(D)\left[1 \quad \frac{1 + D + D^2}{1 + D^2}\right] \\ &= u'(D)G_1(D) \end{aligned} \tag{9.39}$$

Putem scrie că

$$u'(D) = u(D)T(D) \tag{9.40}$$

unde

$$\mathbf{T}(D) = [1 + D^2] \tag{9.41}$$

Înmulțirea lui **u**(D) cu **T**(D) generează o versiune aleatorizată (scramblată) a şirului de intrare. Această operație se poate efectua cu un registru de deplasare având două etaje şi cu o poartă logică SAU EXCLUSIV aşa cum arată fig.9.7.

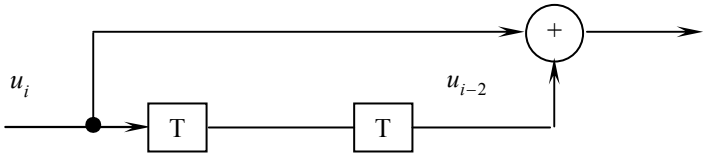

Fig. 9.7. Circuit de aleatorizare a şirului de intrare, utilizabil la intrarea codorului convoluțional sistematic reprezentat în fig. 9.5.

Prin urmare, mulțimea şirurilor de intrare $\mathbf{u}'(D)$, înmulțite de matricea generatoare $\mathbf{G}_1(D)$, produce aceeaşi mulțime de şiruri de ieşire ca şi matricea generatoare inițială **G**(D), astfel încât:

$$\mathbf{G}(D) = \mathbf{T}(D)\mathbf{G}_1(D) \tag{9.42}$$

Spunem că aceste două matrice sunt echivalente. Condiția pe care trebuie să o îndeplinească matricea **T**(D) este de a fi inversabilă. În acest exemplu, inversa lui **T**(D) este dată de:

$$\mathbf{T}^{-1}(D) = \left[\frac{1}{1+D^2}\right] \tag{9.43}$$

Să observăm, totuşi, că elementele matricei $\mathbf{G}_1(D)$ nu sunt polinoame, ci funcții raționale. Şirul de ieşire de paritate din ecuația (9.39) se obține înmulțind şirul de intrare cu polinomul $(1+D+D^2)$ şi împărțindu-l la polinomul $(1+D^2)$. Înmulțirea se poate efectua cu un singur registru de deplasare fără reacție, iar împarțirea, cu un registru de deplasare cu reacție. În general, înmulțirea şi împărțirea şirului de intrare $u(D)$ cu polinoamele $a(D)$ şi $q(D)$, respectiv,

$$v(D) = u(D)\frac{a(D)}{q(D)} \tag{9.44}$$

se pot efectua cu circuitele ilustrate în figurile 9.8. şi 9.9. Polinoamele cu coeficienți binari $a(D)$ şi $q(D)$ se pot scrie:

$$a(D) = a_0 + a_1 D + \cdots + a_\nu D^\nu$$
$$q(D) = 1 + q_1 D + \cdots + q_\nu D^\nu \tag{9.45}$$

Raportul

$$T(D) = \frac{a(D)}{q(D)} \tag{9.46}$$

reprezintă *o funcție de transfer rațională*. În fig.9.8, se arată realizarea în forma canonică de *controlor* a funcției raționale din ecuația (9.46).

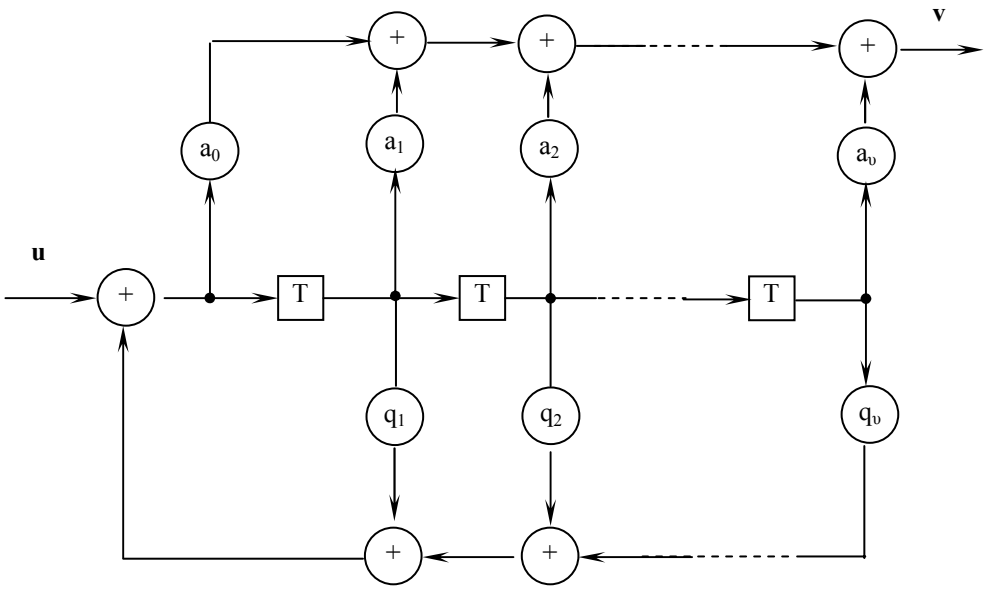

Fig. 9.8. Forma canonică de *controlor* a funcției de transfer *a(D)/q(D)*.

Forma canonică de *observator* de implementare a funcției de transfer raționale din ecuația (9.46) este ilustrată în fig. 9.9.

Întrucât circuitul din fig. 9.9, realizat numai cu elemente de întârziere şi porți logice SAU EXCLUSIV, este liniar, în virtutea proprietății de superpoziție a sistemelor liniare, avem:

$$v(D) = c(D)\left(a_0 + a_1 D + \cdots + a_\nu D^\nu\right) + v(D)\left(q_1 D + \cdots + q_\nu D^\nu\right) \tag{9.47}$$

Ecuația (9.47) este aceeaşi cu ecuația (9.44). În această implementare, elementele de întârziere nu formează un registru de deplasare, căci sunt separate de sumatoare modulo 2.

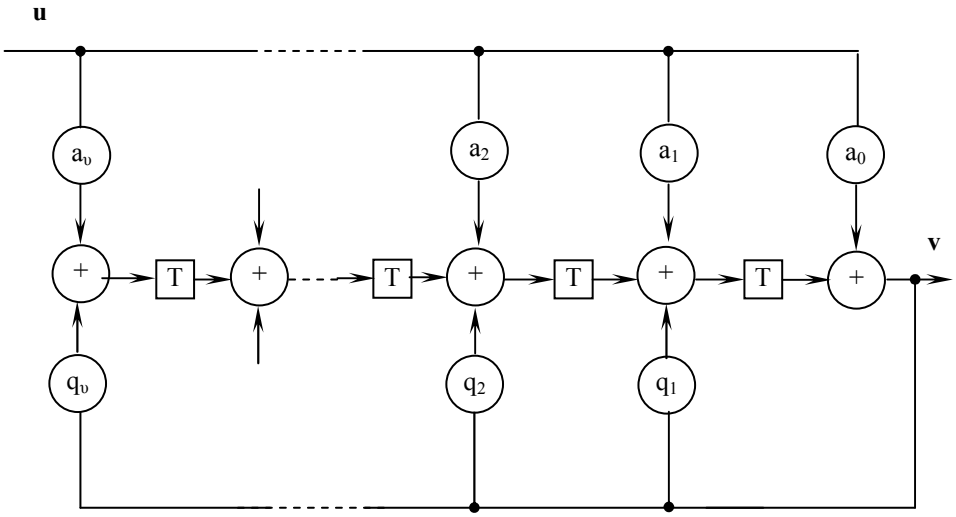

Fig. 9.9. Forma canonică de *observator* a funcției de transfer $a(D) / q(D)$.

Schema unui codor sistematic (2,1) în forma canonică de controlor pentru codorul cu matrice generatoare $\mathbf{G}_1(D)$ a fost deja arătată în figura 9.5. Schema corespunzătoare în forma canonică de observator este dată în figura 9.10.

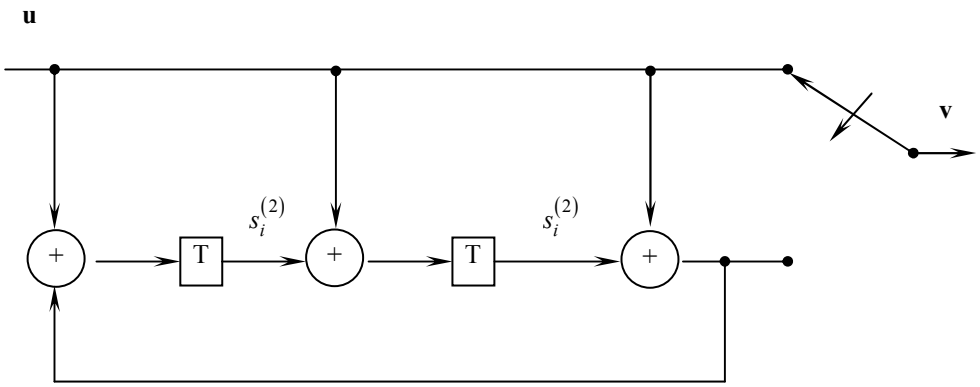

Fig. 9.10. Forma canonică de observator a codorului sistematic (2,1) cu matricea generatoare $\mathbf{G}_1(D)$.

9.4. MATRICEA DE CONTROL

La fel ca pentru un cod bloc, putem descrie un cod convoluțional cu ajutorul unei matrice de control. Aceasta este o matrice $(n-k)\times n$ de funcții raționale care satisface:

$$\mathbf{G}(D)\mathbf{H}^T(D)=\mathbf{0} \tag{9.48}$$

Pentru toate şirurile de cod $v(D)$, operația codare se poate scrie

$$\mathbf{v}(D)\mathbf{H}^T(D)=\mathbf{0} \tag{9.49}$$

Matricea generatoare a unui cod convoluțional sistematic este de forma

$$\mathbf{G}(D)=\left[\mathbf{I}\quad \mathbf{P}(D)\right] \tag{9.50}$$

unde **I** este o matrice identitate $k\times k$, iar $\mathbf{P}(D)$ este o matrice $k\times(n-k)$ale cărei elemente sunt funcții raționale gen (9.46). Matricea de control este de forma:

$$\mathbf{H}(D)=\left[\mathbf{P}^T(D)\quad \mathbf{I}\right] \tag{9.51}$$

unde **I** este o matrice identitate $(n-k)\times(n-k)$, iar $\mathbf{P}^T(D)$ este transpusa matricei $\mathbf{P}(D)$ din (9.50).

Pentru codorul sistematic de rată 1/2 reprezentat în fig. 9.5, matricea de control obținută din matricea generatoare $\mathbf{G}_1(D)$ este dată de:

$$\mathbf{H}(D)=\left[\frac{1+D+D^2}{1+D^2}\quad 1\right] \tag{9.52}$$

9.5. CODURI CATASTROFALE

Codurile sistematice prezintă avantajul că mesajul este conținut nemodificat în cuvântul de cod, astfel încât poate fi extras direct din şirul recepționat. În cazul codoarelor nesistematice, însă, este necesar un circuit de inversare pentru a recupera informația din şirul decodat. Să specificăm acest circuit de inversare prin matricea $\mathbf{G}^{-1}(D)$, astfel încât

$$\mathbf{v}(D)\mathbf{G}^{-1}(D)=\mathbf{u}(D)\mathbf{G}(D)\mathbf{G}^{-1}(D)=\mathbf{u}(D)D^l \tag{9.53}$$

În (9.53), D^l reprezintă o întârziere cu l intervale de bit. Pentru a obține şirul de mesaj $\mathbf{u}(D)$ sau o versiune întârziată a sa $\mathbf{u}(D)D^l$, trebuie satisfăcută relația

$$\mathbf{G}(D)\mathbf{G}^{-1}(D) = \mathbf{I}D^l \tag{9.54}$$

unde $\mathbf{I}$ este o matrice identitate $k \times k$. Cu cu alte cuvinte, matricea generatoare $\mathbf{G}$ trebuie să fie inversabilă.

Fie $\Delta_i(D),\ i = 1, 2, \cdots, \binom{n}{k}$, determinanții celor $\binom{n}{k}$ submatrice $k \times k$ distincte ale matricei generatoare $\mathbf{G}(D)$. Se demonstrează că matricea polinomială inversă $\mathbf{G}^{-1}(D)$ există dacă şi numai dacă

$$\text{c.m.m.d.c.} \left[\Delta_i(D),\ i = 1, 2, \cdots, \binom{n}{k} \right] = D^l,\ l \geq 0 \tag{9.55}$$

unde c.m.m.d.c. este prescurtarea de la „cel mai mare divizor comun“.

EXEMPLUL 9.5: Pentru codorul din fig.9.1, avem

$$\text{c.m.m.d.c.} \left[1 + D^2 \quad 1 + D + D^2 \right] = 1 \tag{9.56}$$

Inversa matricei generatoare este

$$\mathbf{G}^{-1}(D) = \begin{bmatrix} 1 + D \\ D \end{bmatrix} \tag{9.57}$$

Dacă matricea generatoare $\mathbf{G}$ nu are inversă, codul se numeşte *catastrofal*. Pentru codurile catastrofale, un număr finit de erori de transmisie cauzează un număr infinit de erori de decodare.

EXEMPLUL 9.6: O matrice generatoare care generează un cod catastrofal este

$$\mathbf{G}(D) = \left[1 + D \quad 1 + D^2 \right] \tag{9.58}$$

Într-adevăr, $\mathbf{G}^{-1}(D)$ nu există, căci

$$\text{c.m.m.d.c.} \left[1 + D \quad 1 + D^2 \right] = 1 + D \tag{9.59}$$

Dacă şirul de informație este

$$\mathbf{u}(D) = 1 + D + D^2 + \cdots = \frac{1}{1 + D} \tag{9.60}$$

şirurile de cod sunt

$$\mathbf{v}^{(1)}(D) = 1 \qquad \mathbf{v}^{(2)}(D) = 1 + D\,. \tag{9.61}$$

Ponderea şirului de cod este 3, în vreme ce ponderea şirului de intrare este infinită. Dacă şirul de cod se transmite pe un canal sistematic binar, în care apar trei erori, care schimbă cei trei biți de 1 în 0, şirul recepționat va conține numai zerouri. Întrucât acesta este cuvânt de cod, decodorul îl va considera valabil şi-l va furniza utilizatorului. Prin urmare, şirul decodat va avea un număr infinit de erori, deşi în canal nu s-au produs decât trei erori.

Este evident că trebuie să evităm codurile catastrofale.

9.6. DIAGRAMA DE STARE

Un codor convoluțional poate fi considerat un circuit liniar cu un număr finit de stări, astfel încât poate fi descris printr-o *diagramă de stare*. Starea codorului este prin definiție conținutul memoriei. Dacă ν este memoria totală a codorului, numărul stărilor este egal cu 2^{ν}. Starea curentă şi ieşirea codorului sunt determinate univoc de starea precedentă şi de intrarea curentă. Atunci când un bloc de mesaj se deplasează în codor, acesta suferă o tranziție de stare. Diagrama de stare este un graf orientat ale cărui noduri reprezintă stările codorului, iar arcele reprezintă tranzițiile de stare. Fiecare arc orientat este etichetat cu perechea intrare-ieşire. Dacă se dă o stare curentă a codorului, şirul de informație de la intrare determină drumul prin diagrama de stare şi şirul de ieşire.

EXEMPLUL 9.7: Diagrama de stare a codorului convoluțional nesistematic reprezentat în fig. 9.1 este arătată în fig. 9.11.

Codorul are patru stări, notate cu S_j, $j = 0, 1, 2, 3$, iar

$$j = 2u_{i-1} + u_{i-2}\,. \tag{9.62}$$

Din fiecare stare pleacă două drumuri, corespunzătoare celor două valori posibile ale bitului de mesaj de intrare. Codorul sistematic echivalent pentru acest cod în forma canonică de observator este arătat în fig. 9.10, unde starea curentă S_j are indicele j dat de

$$j = 2s_i^{(1)} + s_i^{(2)} \tag{9.63}$$

Diagrama de stare pentru acest codor este ilustarată în fig. 9.12.

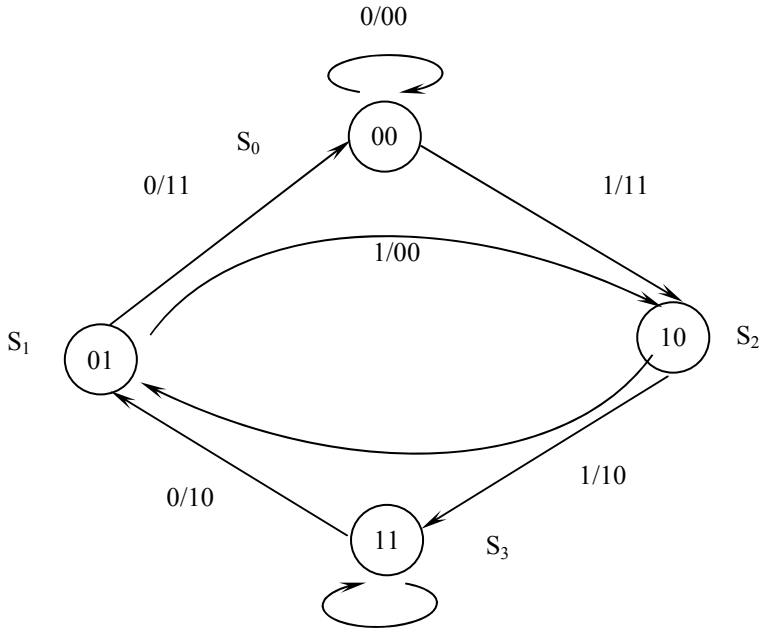

Fig. 9.11. Diagrama de stare pentru codorul convoluțional nesistematic din fig.9.1.

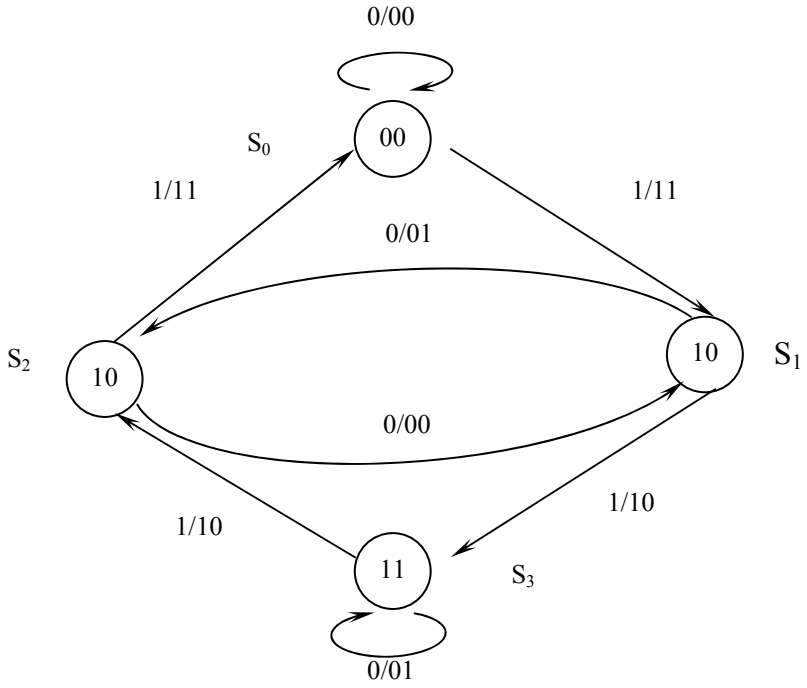

Fig. 9.12. Diagrama de stare pentru codorul sistematic (2,1) din fig. 9.10.

Să observăm că şirurile de ieşire din cele două diagrame de stare arătate în figurile 9.11 şi 9.12 sunt identice, căci codoarele sunt echivalente, dar că diferă corespondența dintre biții de mesaj de la intrare şi cadrele de ieşire.

9.7. DIAGRAMA TRELLIS

Cuvântul *trellis* înseamnă în limba engleză, în general, împletitură de nuiele, leasă, spalier, grătar de zăbrele. În particular, în teoria codurilor, prin diagrama trellis se înțelege un graf orientat derivat din diagrama de stare şi deosebit de util în studiul codurilor convoluționale. Am văzut deja în capitolul 6 că un cod bloc admite descrierea printr-o diagramă trellis. Aplicarea diagramei trellis la codurile bloc s-a făcut, însă, abia după ce s-a dovedit marea sa utilitate pentru tratarea codurilor convoluționale.

O *secțiune trellis* include toate stările posibile la timpul curent i reprezentate ca puncte dispuse pe verticală, toate stările posibile la timpul următor $i + 1$, reprezentate similar, precum şi toate tranzițiile de stare permise între fiecare stare curentă şi stările următoare. Secțiunea de trellis corespunzătoare diagramei de stare din fig. 9.11 este arătată în fig. 9.13. Fiecare tranziție este etichetată cu $u_i / v_i^{(1)} v_i^{(2)}$.

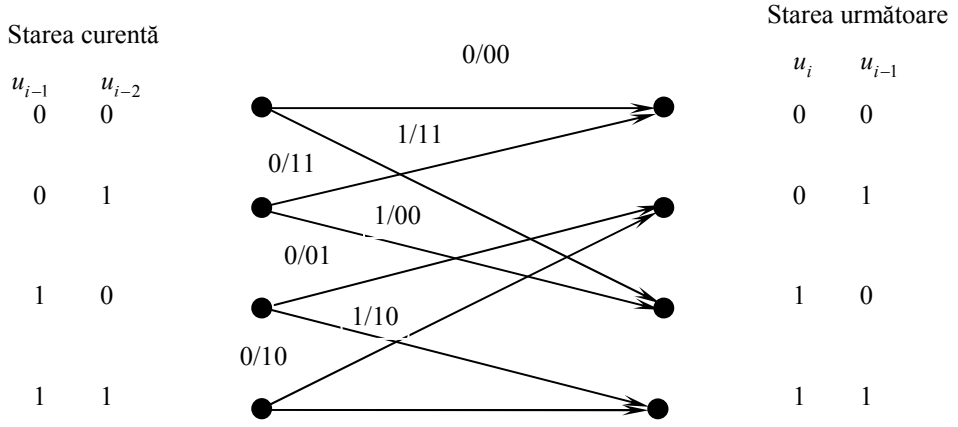

Fig.9.13. Secțiune de trellis pentru codorul convoluțional nesistematic (2,1) din fig. 9.1.

Înaintea unei sesiuni de transmisiune de date, codorul convoluțional este inițializat în starea S_0. Primul bit aplicat la intrare face ca starea următoare să fie S_0 sau S_2, după cum valoarea sa este 0 sau 1, respectiv. Aşadar, primul bit de intrare nu poate duce codorul în stările S_1 şi S_3. Din starea S_2, al doilea bit de intrare face codorul să treacă într-una din aceste două stări, în S_1 sau în S_3. În general, codorul atinge numărul maxim posibil de stări, 2^{ν}, după ν unități de timp. În exemplul nostru, ν = 2. După aceasta, secțiunea de trellis se repetă până la terminarea transmisiunii. Diagrama trellis pentru codorul convoluțional nesistematic (2,1) având secțiunea de trellis arătată în fig. 9.13 este reprezentată în fig. 9.14.

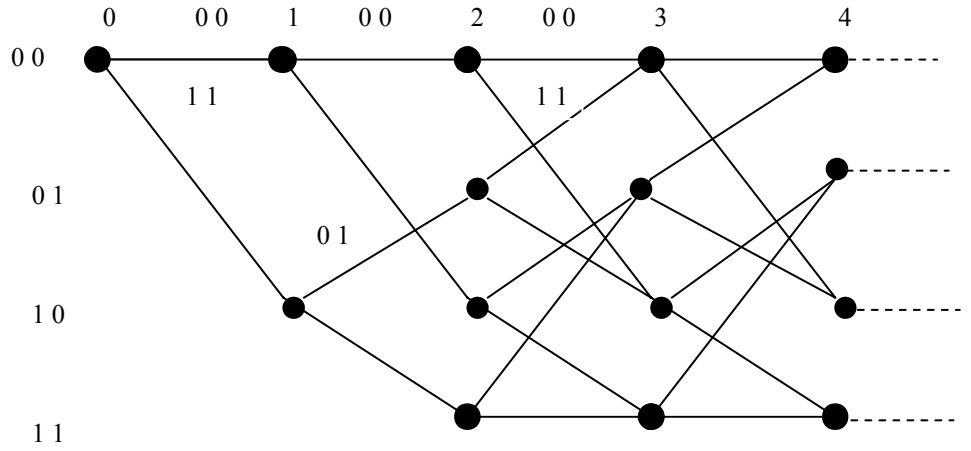

Fig. 9.14. Diagrama trellis pentru codorul convoluțional nesistematic (2,1) din fig. 9.1.

În general, trellisul unui cod convoluțional (*n,k*) are 2^k ramuri plecând din fiecare stare şi 2^k ramuri intrând în fiecare stare.

9.8. DISTRIBUȚIA DE PONDERE A CODURILOR CONVOLUȚIONALE

Prin utilizarea unui cod convoluțional într-un sistem de comunicație, urmărim să reducem probabilitatea de eroare. Aceasta depinde de proprietățile de distanță. După cum decodarea se bazează pe o decizie fermă sau suplă, sunt două tipuri de distanță pe care le considerăm.

În cazul decodării bazate pe o decizie fermă, decodorul operează cu simboluri binare, iar performanța codului se măsoară cu ajutorul distanței Hamming. Prin definiție, *distanța liberă minimă* a unui cod convoluțional, notată cu d_{lib}, este distanța Hamming minimă dintre oricare două cuvinte de cod. Codurile convoluționale fiind liniare[1], distanța Hamming dintre două cuvinte de cod este egală cu ponderea sumei lor modulo 2, care este un alt cuvânt de cod. De aceea, distanța liberă minimă este ponderea minimă a tuturor cuvintelor de cod diferite de zero. Cu alte cuvinte, cuvântul de cod reprezentat prin acel drum prin trellis ce trece numai prin starea zero poate fi utilizat drept cuvânt de referință la determinarea distanței libere minime.

EXEMPLUL 9.8: Să vedem împreună cum a fost proiectat codorul reprezentat în figura 9.1.

Primii paşi de proiectare sunt opțiuni influențate de sistemul de comunicație în care includem codorul convoluțional. Astfel, alegem rata $R = \frac{1}{2}$, k = 1, n = 2 şi ν = 2. Optând apoi pentru un codor nesistematic, urmează că starea este dată de doi biți de intrare anteriori, u_{i-1} şi u_{i-2}. Deci, starea curentă este $u_{i-1}\ u_{i-2}$, iar bitul de intrare u_i determină trecerea în starea următoare $u_i\ u_{i-1}$. Acesta determină secțiunea de trellis reprezentată în fig. 9.13, dar fără etichetarea tranzițiilor de stare. Un trellis fără etichete pe tranziții se numeşte *trellis topologic*. (Este ca într-un oraş nou constituit în care străzile nu au încă nume.) Urmează să etichetăm tranzițiile de stare astfel încât să maximizăm distanța Hamming liberă a codului. Pentru acesta, partiționăm mulțimea stărilor $\{S_0, S_1, S_2, S_3\}$ astfel: pentru stările de origine, $\Sigma_0 = \{S_0, S_1\}$ şi $\Sigma_1 = \{S_2, S_3\}$ iar pentru stările de ieşire $\Sigma_0' = \{S_0, S_2\}$ şi $\Sigma_1' = \{S_1, S_3\}$. Partiționăm de asemenea mulțimea valorilor pe care le poate lua cadrul de ieşire $v_i^{(1)} v_i^{(2)}$ astfel: $F_0 = \{00, 11\}$ şi $F_1 = \{01, 10\}$. Această din urmă partiție nu s-a făcut întâmplător, ci astfel încât să se maximizeze distanța Hamming a mulțimilor F_0 şi F_1, în cazul nostru egală cu 2. Aplicăm acum următoarele reguli:

[1] Trebuie mentionat că, pentru a se asigura invariația rotațională a schemelor de modulație codată trellis, se utilizează ca blocuri componente coduri convoluționale neliniare.

1. Tranzițiile ce-şi au origine într-una din stările din Σ_0 primesc etichete diferite între ele din F_0, iar tranzițiile ce pleacă din stările din Σ_1 primesc etichete diferite între ele, dar din F_1.
2. Tranzițiile ce ajung într-una din stările din $\Sigma_0^{'}$ primesc etichete diferite între ele din F_0, iar tranzițiile ce ajung în sările din $\Sigma_1^{'}$ primesc etichete diferite între ele din F_1.
3. Tranziția de la starea 0 la starea 0 este determinată de $u_i = 0$ şi are eticheta $v_i^{(1)} v_i^{(2)} = 00$.

Cu ajutorul acestor reguli, întocmim tabla logică de mai jos:

Tabelul 9.1

Intrare			Ieşire	
u_i	u_{i-1}	u_{i-2}	$v_i^{(1)}$	$v_i^{(2)}$
0 1	0	0	0 1	0 1
0 1	0	1	1 0	1 0
0 1	1	0	0 1	1 0
0 1	1	1	1 0	0 1

Din tabla logică, obținem fără dificultate

$$v_i^{(1)} = u_i \oplus u_{i-2}$$
$$v_i^{(2)} = u_i \oplus u_{i-1} \oplus u_{i-2}.$$

Cu aceasta, proiectarea este încheiată. Pentru a determina distanța Hamming liberă, examinăm diagrama trellis din fig. 9.14 şi observăm că şirurile de stări $\cdots 0, 0, 0, 0, \cdots$ şi $\cdots 0, 2, 1, 0, \cdots$ corespund unor cuvinte de cod ce diferă printr-o distanță Hamming egală cu 5. Aceasta este, deci, distanța Hamming liberă pentru acest cod.

Un decodor bazat pe decizii suple primeşte de la demodulator semnale analogice sau cuantizate pe mai mult decât două nivele, iar operația de decodare se bazează pe distanța euclidiană. *Distanța euclidiană liberă minimă*, notată cu d_{Elib}, este distanța euclidiană minimă dintre oricare două cuvinte de cod. Ea depinde atât de structura diagramei trellis a codului

convoluțional cât şi de tipul de modulație. Spre exemplu, dacă se utilizează modulația binară de fază (BPSK) ce transmite simboluri binare din alfabetul{-1,1}, distanța euclidiană liberă minimă pentru codul (2,1) având diagrama trellis din fig. 9.14 este $d_{Elib} = 2\sqrt{5}$.

Pentru a calcula performanța unui cod convoluțional în ce priveşte erorile, avem nevoie de distribuția ponderilor. Prin definiție, A_i este numărul cuvintelor de cod de pondere *i* din trellis ce diverg la un nod de la drumul zero şi se reunesc cu aceasta pentru prima oară într-un alt nod.

Mulțimea

$$\left\{A_{d_{lib}}, A_{d_{lib+1}}, \cdots, A_i, \cdots\right\}$$

se numeşte distribuția de pondere a codului convoluțional.

Distribuția de pondere se poate calcula modificând diagrama de stare a codului. Diagrama de stare modificată se obține despicând starea S_0 într-o stare inițială, S_{in} , şi o stare finală, S_{ies} , eliminând totodată auto – bucla din jurul lui S_0 . Orice drum din diagrama de stare ce conectează starea inițială S_{in} şi starea finală S_{ies} reprezintă un cuvânt de cod ce diverge de la drumul zero şi se reuneşte cu acesta exact o dată. A_i este egal cu numărul drumurilor de pondere *i* din diagrama de stare modificată ce conectează starea inițală cu starea finală.

Fie *X* indeterminata asociată cu ponderea Hamming *i* a cuvântului de ieşire, *Y* indeterminata asociată cu ponderea Hamming *j* a şirului de informație şi *Z* indeterminata asociată cu fiecare ramură. Fiecare ramură din diagrama de stare modificată se etichetează cu un câştig de ramură $X^i Y^j Z$.

Diagrama de stare modificată, etichetată cu câştigurile de ramură, se numeşte *diagrama de stare augmentată*.

EXEMPLUL 9.9: Să considerăm codul convoluțional nesistematic (2,1) reprezentat în fig. 9.1, a cărui diagramă de stare este arătată în fig. 9.11. Diagrama de stare augmentată este ilustrată în fig. 9.15.

Sistemul de ecuații descriind tranzițiile de stare din diagrama de stare angmentată este următorul:

$$S_1 = XZS_2 + XZS_3 \qquad (9.64)$$

$$S_2 = YZS_1 + X^2 YZS_{in} \qquad (9.65)$$

$$S_3 = XYZS_2 + XYZS_3 \qquad (9.66)$$

$$S_{ies} = X^2 ZS_1 \qquad (9.67)$$

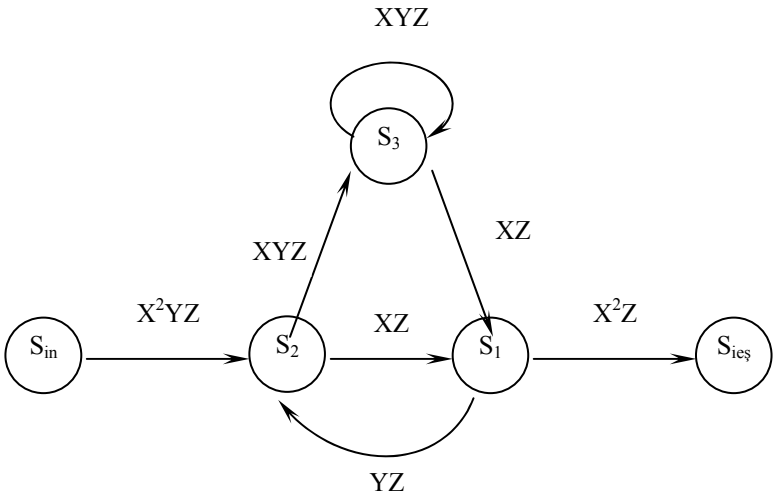

Fig. 9.15. Diagrama de stare augmentată a fig. 9.11.

Explicităm S_3 din (9.66):

$$S_3 = \frac{XYZS_2}{1-XYZ} \tag{9.68}$$

Înlocuind (9.68) în (9.64), obținem:

$$S_1 = \frac{XZS_2}{1-XYZ} \tag{9.69}$$

Înlocuind (9.68) în (9.65), avem:

$$S_2 = \frac{(1-XYZ)X^2YZS_{in}}{1-XYZ-XYZ^2} \tag{9.70}$$

Înlocuind (9.69) în (9.67), obținem

$$S_{ies} = \frac{X^3Y^2S_2}{1-XYZ} \tag{9.71}$$

În sfârşit, înlocuind (9.70) în (9.71), obținem funcția generatoare *T (X,Y,Z)*

$$\mathrm{T\ (X,Y,Z)} = \frac{S_{ies}}{S_{in}} = \frac{X^5YZ^3}{1-XYZ(1+Z)} \tag{9.72}$$

Fie funcția rațională $1/(1+x)$. Prin împărțirea lungă, obținem:

$$\frac{1}{1-x} = 1+x+x^2+x^3+\cdots \tag{9.73}$$

Aplicând identitatea (9.73) în (9.72), putem scrie funcția generatoare astfel:

$$T(X,Y,Z) = X^5YZ^3 + X^6Y^2\left(Z^4 + Z^5\right) + X^7Y^3\left(Z^5 + 2Z^6 + Z^7\right) + X^8Y^4\left(Z^6 + 3Z^7 + 3Z^8 + Z^9\right) + \cdots$$

Este util să ordonăm $T(X,Y,Z)$ după puterile lui Z, care arată numărul de ramuri care dau o anumită pondere:

$$T(X,Y,Z) = X^5YZ^3 + X^6Y^2Z^4 + \left(X^6Y^2 + X^7Y^3\right)Z^5 + \left(2X^7Y^3 + X^8Y^4\right)Z^6 + \left(X^7Y^3 + 3X^8Y^4 + X^9Y^5\right)Z^7 + \left(3X^8Y^4 + 4X^9Y^5 + X^{10}Y^6\right)Z^8 + \cdots$$

Funcția generatoare confirmă ce ştiam deja, şi anume, că distanța liberă pentru acest cod este 5 (puterea lui X din primul termen) şi că numărul cuvintelor de cod la această distanță este de $A_5 = 1$. Şirul de informație ce generează acest cuvânt de cod are ponderea Hamming de 1, iar cuvântul de cod conține trei ramuri de pondere diferită de zero. Un alt şir de informație de pondere 2 produce un cuvânt de cod de pondere 6 cu patru ramuri de pondere diferită de zero şi aşa mai departe.

9.9. CODURI CONVOLUȚIONALE PUNCTURATE

Puncturarea unui cod este operația prin care se măreşte rata unui cod netransmițând anumite simboluri din cuvântul de cod, de obicei, simboluri de control. Codurile puncturate sunt coduri convoluționale (n, k) derivate dintr-un cod convoluțional (n, 1) zis „mamă“.

EXEMPLUL 9.10: Fie codorul convoluțional (2, 1) reprezentat în fig. 9.1. Secțiunea de trellis este arătată în fig. 9.13. Dacă se puncturează unul din patru biți de ieşire, codorul va produce trei biți codați la fiecare doi biți de informație. În exemplul nostru, primul bit din fiecare a doua ramură (sau tranziție) a diagramei trellis pentru codul de rată 1/2 este puncturat. Noul cod este variabil în timp şi are rata egală cu 2/3. Această operație se poate descrie printr-un tablou de puncturare

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{9.74}$$

Un zero în tabloul de puncturare înseamnă că bitul respectiv nu este transmis.

Diagrama trellis a codului puncturat (3, 2) este reprezentată în fig. 9.16. Un *X* indică un bit eliminat prin puncturare.

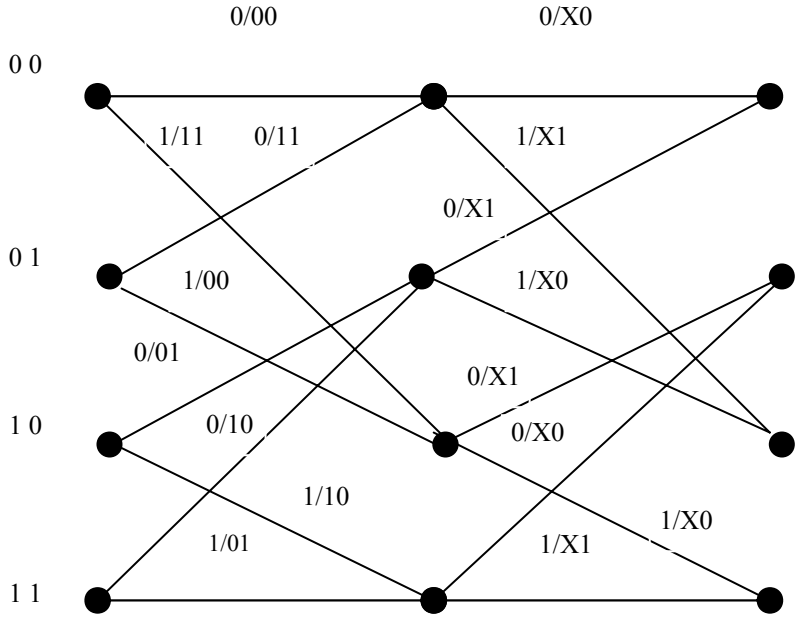

Fig. 9.16. Diagrama trellis a unui cod puncturat de rată $2/3$ produs prin eliminarea periodică a unor simboluri dintr-un cod de rată $1/2$.

În general, un cod convoluțional puncturat de rată *p*/*q* poate fi construit dintr-un cod convoluțional (*n*, 1) eliminând *np*–*q* biți codați din fiecare *np* biți codați corespunzând la *p* biți de informație aplicați la intrare. Codul (*n*, 1) se numește *codul mamă* și este specificat de matricea generatoare

$$\mathbf{G}(D) = \left[g^{(1)}(D) \quad g^{(2)}(D) \quad \cdots \quad g^{(n)}(D) \right] \tag{9.75}$$

unde

$$g^{(j)}(D) = g_0^{(j)} + g_1^{(j)} D + \cdots + g_\nu^{(j)} D^\nu, \tag{9.76}$$

$1 \le j \le n$ și $g_l^{(j)} \in \{0,1\}, l = 0, 1, \cdots, \nu.$

Tabloul de puncturare **P**, care arată biții ce se elimină periodic, este de forma

$$\mathbf{P}=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1P} \\ p_{21} & p_{22} & \cdots & p_{2P} \\ \vdots & & \ddots & \\ p_{n1} & p_{n2} & \cdots & p_{np} \end{bmatrix} \tag{9.77}$$

unde $p_{je} \in \{0,1\}$, $1 \le j \le n$ şi $1 \le l \le p$.

Întrucât puncturarea se efectuează periodic la fiecare *np* simboluri de cod, *p* se numeşte *perioada de puncturare*.

9.10. ALGORITMUL VITERBI

Se cunosc mai multe metode de decodare a codurilor convoluționale, dintre care una singură este teoretic optimă, celelalte fiind suboptimale. Metoda optimă se numeşte *decodare de secvență de maximă plauzibilitate* sau *algoritmul Viterbi*; celelalte, deşi suboptimale teoretic, au oferit soluții practice în raport cu nivelul de dezvoltare a electronicii digitale. În prezent, dispunem de procesoare digitale cu care putem implementa în timp real algoritmul Viterbi, care este descris în continuare.

Este important să ne reamintim că, spre deoasebire de un codor bloc, un codor convoluțional generează un singur cuvânt de cod într-o transmisiune dată. Lungimea acestui cuvânt de cod depinde, deci, de volumul de informație ce se transmite într-o sesiune anume. Notăm cu *L* numărul blocurilor de informație de câte *k* biți. Un alt punct important este acela că, prin inițializarea cu zero a registrelor de deplasare ale codorului convoluțional, acesta pleacă întotdeauna din starea S_0; în afară de aceasta, prin transmisia unui număr suplimentar de zerouri, fără valoare informațională, codorul convoluțional este determinat să se oprească în aceeaşi stare S_0 la sfârşitul sesiunii de transmisiune. Să nu facem confuzie între memoria *m* a codorului, definită în (9.13), şi memoria totală a codorului ν, definită în (9.14). Memoria totală ν determină numărul stărilor 2^{ν}, în vreme ce memoria *m* ne spune numărul unităților de timp în care un bloc de intrare se găseşte în registrele de deplasare ale codului. Astfel, un şir de informație $\mathbf{u}=(u_0, u_1, \cdots, u_{L-1})$ de lungime *kL* (biți) este

codat drept cuvântul de cod $\mathbf{v}=(v_0,v_1,\cdots,v_L,\cdots,v_{L+m-1})$ de lungime $N=n(L+m)$ (biți). Utilizând o modulație binară (de exemplu, *BPSK* sau *FSK*), cuvântul de cod este transmis printr-un canal de comunicație. Modelăm acest canal de comunicație drept *canal discret fără memorie* cu intrare binară şi ieşire *Q*-ară, aşa cum se arată în fig. 9.17.

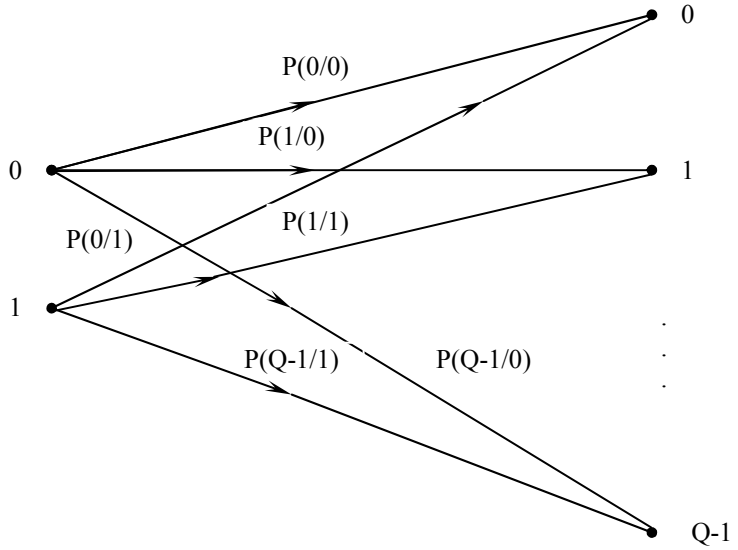

Fig. 9.17. Model de canal fără memorie cu intrare binară şi ieşire *Q*-ară.

Trebuie să înțelegem acest model în sensul următor: emițătorul aplică la intrarea în canal, unul câte unu, simboluri binare (0 şi 1), dar la recepție, semnalul demodulat, care ia valori continue într-un interval bine definit, este transformat de un convertor analogic / digital în *Q* nivele discrete. Am notat cu $P(j/0)$ probabilitatea de a avea la ieşire nivelul *j* condiționată de emisia unui bit de 0 şi cu $P(j/1)$ probabilitatea de a avea nivelul *j* condiționată de emisia unui bit de 1, unde $0\le j\le Q-1$. La recepție, după încheierea transmisiei, dispunem de şirul *Q*-ar $\mathbf{r}=(r_0,r_1,\cdots,r_{L+m-1})$. Vom presupune că avem posibilitatea tehnică de a memora toate aceste blocuri. Având în vedere că blocurile de intrare $\mathbf{u}_i$ au câte *k* biți, iar un cadru de ieşire $\mathbf{v}_i$ are *n* biți, cele trei şiruri se mai pot scrie şi altfel:

$$\mathbf{u}=(u_0,u_1,\cdots,u_{kL-1})$$
$$\mathbf{v}=(v_0,v_1,\cdots,v_{N-1})$$
$$\mathbf{r}=(r_0,r_1,\cdots,r_{N-1})$$

unele indicii arată ordinea simbolurilor în timp. Pe baza şirului recepționat **r**, decodorul poate produce o estimație $\hat{\mathbf{v}}$ a cuvântului de cod **v**. Un decodor de plauzibilitate maximă alege $\hat{\mathbf{v}}$ drept acel cuvânt de cod **v** care maximează funcția logaritm de probabilitate $\log P(\mathbf{r} \mid \mathbf{v})$. Pentru un canal discret fără memorie, avem

$$P(\mathbf{r} \mid \mathbf{v}) = \prod_{i=0}^{L+m-1} P(\mathbf{r}_i \mid \mathbf{v}_i) = \prod_{i=0}^{N-1} P(r_i \mid v_i). \tag{9.78}$$

Urmează că

$$\log P(\mathbf{r} \mid \mathbf{v}) = \sum_{i=0}^{L+m-1} \log P(\mathbf{r}_i \mid \mathbf{v}_i) = \sum_{i=0}^{N-1} \log P(r_i \mid v_i) \tag{9.79}$$

În (9.78) şi (9.79), $P(r_i \mid v_i)$ este probabilitatea de tranziție a canalului, adică probabilitatea de a recepționa eronat un simbol r_i atunci când simbolul emis a fost v_i.

Dacă toate cuvintele de cod sunt egal probabile, aceasta este regula de decodare cu probabilitate minimă de eroare. Funcția logaritm de probabilitate $\log P(\mathbf{r} \mid \mathbf{v})$ se numeşte *metrica* asociată cu drumul **v** şi se notează cu $M(\mathbf{r} \mid \mathbf{v})$. Termenii $\log P(\mathbf{r}_i \mid \mathbf{v}_i)$ din suma (9.79) se numesc *metrici de ramură* şi se notează cu $M(\mathbf{r}_i \mid \mathbf{v}_i)$, iar termenii $\log P(r_i \mid v_i)$ se numesc *metrici de bit* şi se notează cu $M(r_i \mid v_i)$. Cu aceste notații, metrica drumului $M(\mathbf{r} \mid \mathbf{v})$ se poate scrie

$$M(\mathbf{r} \mid \mathbf{v}) = \sum_{i=0}^{L+m-1} M(\mathbf{r}_i \mid \mathbf{v}_i) = \sum_{i=0}^{N-1} M(r_i \mid v_i). \tag{9.80}$$

Metrica unui drum parțial pentru primele *j* ramuri ale unui drum se poate exprima astfel:

$$M\left([\mathbf{r} \mid \mathbf{v}]_j\right) = \sum_{i=0}^{j-1} M(\mathbf{r}_i \mid \mathbf{v}_i). \tag{9.81}$$

Algoritmul Viterbi, aplicat la şirul **r** recepționat dintr-un canal discret fără memorie, găseşte drumul prin graful orientat trellis cu metrica cea mai mare, adică, drumul de probabilitate maximă care este, deci, cel mai plauzibil. Algoritmul prelucrează **r** iterativ. La fiecare pas, algoritmul compară metricile tuturor drumurilor ce intră în fiecare stare şi memorează drumul cu metrica cea mai mare, drum numit *supraviețuitor*, împreună cu metrica sa.

Algoritmul Viterbi

Pasul 1. Începând de la unitatea de timp $j = m$, se calculează metrica parțială pentru fiecare drum care intră în fiecare stare. Se memorează supraviețuitorul şi metrica sa pentru fiecare stare.

Pasul 2. Se incrementează j cu 1. Se calculează metrica parțială pentru toate cele 2^k drumuri ce intră într-o anumită stare (la timpul $j + 1$) adunând metrica de ramură ce intră în acea stare la metrica supraviețuitorului din unitatea de timp precedentă j ce duce la respectiva stare. Pentru fiecare stare (la timpul $j + 1$), se memorează drumul cu metrica cea mai mare – supraviețuitorul, împreună cu metrica sa şi se elimină toate celălalte drumuri.

Pasul 3. Dacă $j < L + m$, se repetă pasul 2. În caz contrar, stop.

În fiecare unitate de timp cuprinsă între m şi L, există 2^{ν} supraviețuitori, câte unu pentru fiecare din cele 2^{ν} stări. După unitatea de timp L, există mai puțin supraviețuitori, căci stările sunt mai puține, deoarece codorul revine spre starea zero (S_0). În sfârşit, la unitatea de timp $L + m$, nu mai există decât o stare posibilă, starea zero, şi deci un singur supraviețuitor, aşa încât algoritmul se opreşte. Se demonstrează că acest supraviețuitor este drumulde probabilitate maximă. De aceea, algoritmul Viterbi este optim.

Din perspectiva implementării, este mai convenabil să se utilizeze numere naturale în locul metricilor de bit efective. Metrica de bit $M(r_i \mid v_i) = \log P(r_i \mid v_i)$ se poate înlocui cu $c_2[\log P(r_i \mid v_i) + c_1]$, unde c_1 poate fi orice număr real iar c_2 poate fi orice număr real pozitiv. Se poate demonstra că un drum **v** care maximizează

$$M(\mathbf{r} \mid \mathbf{v}) = \sum_{i=0}^{N-1} M(r_i \mid v_i) = \sum_{i=0}^{N-1} \log P(r_i \mid v_i)$$

maximizează de asemenea

$$\sum_{i=0}^{N-1} c_2[\log P(r_i \mid v_i) + c_1],$$

astfel încât se poate utiliza metrica modificată fără a afecta performanța algoritmului Viterbi. Dacă se alege c_1 astfel încât metrica cea mai mică să fie 0, se poate alege c_2 astfel încât toate metricile să poată fi aproximate prin numere naturale.

EXEMPLUL 9.11: Să considerăm canalul discret fără memorie cu intrare binară şi ieşire cuaternară ($Q = 4$) reprezentat în fig. 9.18.

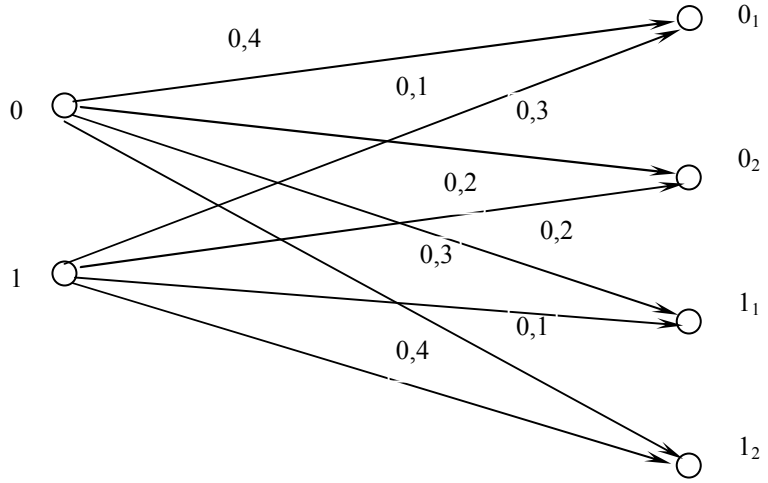

Fig. 9.18. Model de canal discret fără memorie cu intrări binare şi ieşiri cuaternare pentru care, la recepție, sunt două valori pentru zero $(0_1$ şi $0_2)$ şi două valori pentru unu $(1_1$ şi $1_2)$.

Utilizând logaritmi zecimali, metricile de bit pentru acest canal sunt prezentate în Tabelul 9.2. Alegând $c_1 = 1$ şi $c_2 = 17{,}3$, obținem metricile exprimate prin numere naturale din Tabelul 9.3.

Tabelul 9.2

v_i/r_i	0_1	0_2	1_1	1_2
0	-0,4	-0,52	-0,7	-1
1	-1	-0,7	-0,52	-0,4

Tabelul 9.3

v_i/r_i	0_1	0_2	1_1	1_2
0	10	8	5	0
1	0	5	8	10

Să presupunem că se transmite un cuvânt de cod din codul (2,1) generat de codorul reprezentat în fig. 9.1. şi că se recepționează şirul

$$\mathbf{r} = (1_20_1,\ 1_10_2,\ 1_10_1,\ 1_11_1,\ 1_20_1,\ 0_21_1,\ 0_11_1).$$

În fig. 9.19, se arată diagrama trellis, cu metrica de bit indicată deasupra fiecărei stări. Supraviețuitorul final este arătat cu linie groasă $\hat{\mathbf{v}}$ = (00, 11, 10, 01, 10, 11, 00), iar şirul de informație decodat este $\hat{\mathbf{u}}$ = (0 1 1 1 0). Ramurile eliminate sunt marcate cu „\".

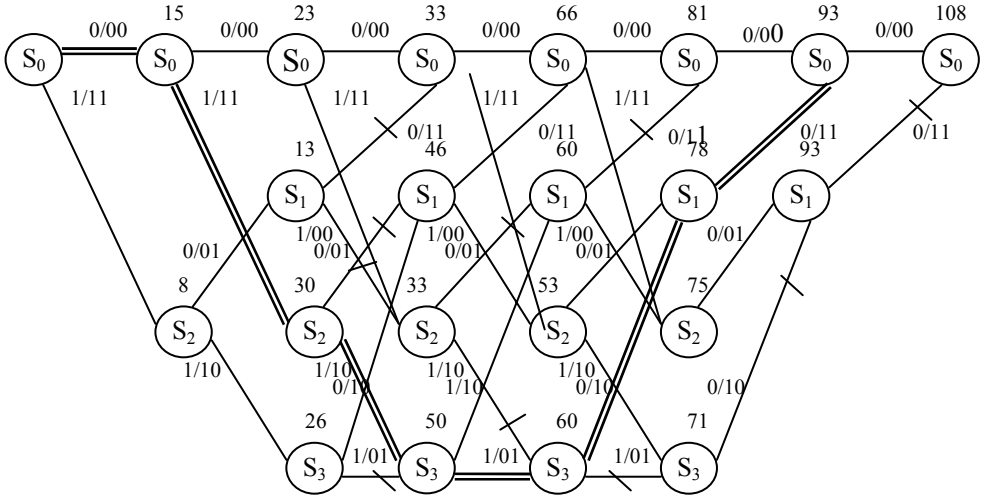

Fig. 9.19. Diagrama trellis pentru Exemplul 9.11.

Observație. Ultimele *m* ramuri din orice drum prin graful trellis corespund unor intrări 0 (*biți albi*) introduse pentru a forța codorul să revină în starea zero (S_0) şi, deci, nu sunt considerate ca parte a şirului de informație.

În cazul particular al utilizării unui model de *canal binar simetric* cu probabilitate de tranziție (de eroare) $p < \frac{1}{2}$, receptorul ia o decizie *fermă* cu privire la valoarea furnizată de demodulator, astfel încât şirul de recepție **r** este binar (Q = 2). Fie $N = n(L+m)$ lungimea cuvântului de cod. Din cauza zgomotului din canalul de comunicație, şirul de recepție **r** poate să difere de cuvântul de cod transmis **v** în anumite poziții binare. Dacă $r_i \neq v_i$, avem $P(r_i \mid v_i) = p$, iar dacă $r_i = v_i$, $P(r_i \mid v_i) = 1 - p$. Fie $d(\mathbf{r}, \mathbf{v})$ distanța Hamming dintre **r** şi **v**. Pentru un cod convoluțional, avem

$$\begin{aligned} \log P(\mathbf{r} \mid \mathbf{v}) &= d(\mathbf{r}, \mathbf{v}) \log p + \left[N - d(\mathbf{r}, \mathbf{v}) \right] \log(1-p) \\ &= d(\mathbf{r}, \mathbf{v}) \log \frac{p}{1-p} + N \log(1-p). \end{aligned} \tag{9.82}$$

Întrucât $\log\left[p/(1-p)\right] < 0$ iar $N\log(1-p)$ este o constantă pentru toți **v**, un decodor de plauzibilitate maximă pentru un canal binar simetric alege **v** drept acel cuvânt de cod care minimizează distanța Hamming

$$d(\mathbf{r},\mathbf{v}) = \sum_{i=0}^{L+m-1} d(\mathbf{r}_i,\mathbf{v}_i) = \sum_{i=0}^{N-1} d(r_i,v_i). \tag{9.83}$$

Prin urmare, dacă se aplică algoritmul Viterbi utilizând modelul unui canal binar simetric, $d(\mathbf{r}_i,\mathbf{v}_i)$ devine metrica de ramură, $d(r_i,v_i)$ devine metrica de bit iar algoritmul trebuie să găsească drumul prin graful trellis cu cea mai mică metrică, adică, drumul cel mai apropiat de **r** folosind ca măsură distanța Hamming.

EXEMPLUL 9.12: Transformăm canalul discret fără memorie cu intrări binare şi ieşiri cuaternare reprezentat în fig. 9.18 într-un canal binar simetric contopind ieşirile 0_1 şi 0_2 într-o singură ieşire 0 şi ieşirile 1_1 şi 1_2 într-o singură ieşire 1. Probabilitatea de tranziție (de eroare) devine p = 0,3. Ca şi în exemplul precedent, presupunem că se transmite un cuvânt de cod generat de codorul din fig. 9.1, dar de această dată canalul este considerat binar simetric, adică, receptorul ia decizii ferme cu privire la valorile furnizate de demodulator, producând şirul binar **r** = (10, 10, 10, 11, 10, 01, 01*)*.

Pentru decodare, utilizăm algoritmul Viterbi luând drept metrică distanța Hamming. Diagrama trellis este reprezentată în fig. 9.20.

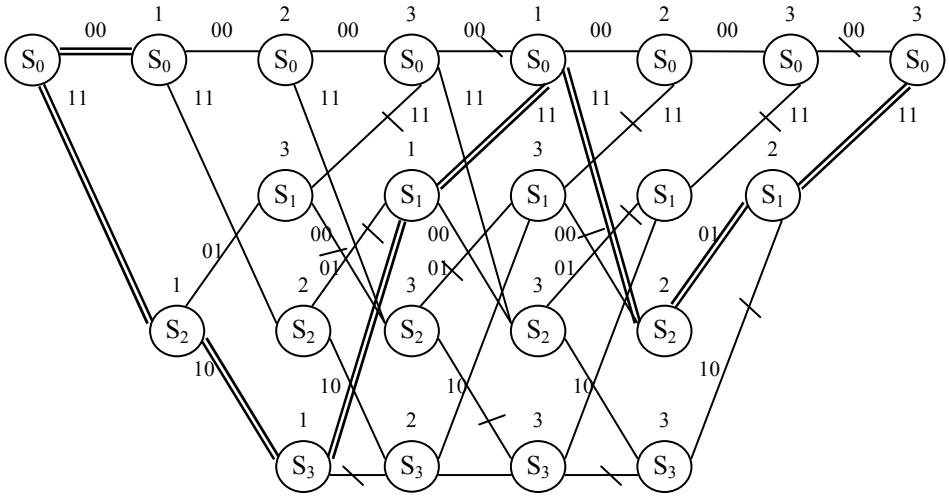

Fig. 9.20. Diagrama trellis pentru Exemplul 9.12.

Şirul decodat este $\hat{u} = (1\ 1\ 0\ 0\ 1)$.

Observăm că într-o stare (S_2) nu se elimină unul din drumuri, căci ambele au aceeaşi metrică. Dacă supraviețuitorul final trece printr-o astfel de stare, există mai multe drumuri de probabilitate maximă. Ori de câte ori se iveşte o astfel de situație, se alege arbitrar unul din drumuri drept supraviețuitor, căci este nepractic pentru implementare să se memoreze un număr variabil de drumuri. Aceasta nu are nici un efect asupra probabilității erorii de decodare.