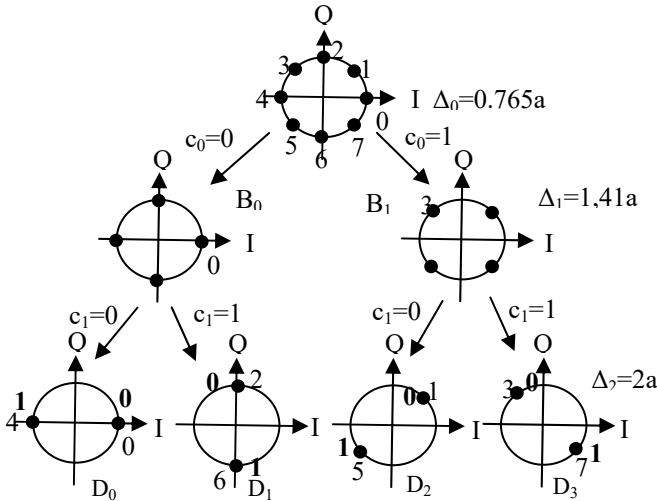


Schema bloc a codorului TCM este prezentată în fig. 10. Modulația necodată care asigură același debit binar util este 4-PSK, care are $P_{mn} = 1 \cdot a^2$ și $d_{mn} = a \cdot \sqrt{2}$, vezi fig 6.b.

Figura 10. Schema bloc a codorului TCM din Ex. 7

c. c.utilizat permite asigurarea acesteia, și apoi este codat cu codul cu $R_c = 1/2$, generând biții $(C_0C_1)^k$.
 - al doilea bit informațional i_2^k este transmis necodat, el devenind C_2^k , pentru uniformitatea notației.



- metoda de alocare multibit fazor bazată pe MSP este prezentată în fig. 11 pentru cazul mapării tribit $((C_0C_1C_2)^k$ – fazor F_k din constelația A8.

Figura 11. Alocarea MSP pentru TCM din Ex.7

- mulțimea fazorilor este împărțită în două submulțimi în funcție de valoarea logică a bitului de cod C_0 .
 - partiționarea urmărește ca distanța minimă dintre fazorii aparținând submulțimilor B_0 și B_1 să fie maxim posibilă; aceasta devine $\Delta_1 = 1,41a$, față de $\Delta_0 = 0.765a$ a constelației inițiale.
 - partiționarea continuă în funcție de valoarea celui de al doilea bit de cod, C_1 , obținându-se patru sub-seturi D_0, \dots, D_3 .

- distanța minimă între fazorii unui subset este $\Delta_2 = 2a$
- partiționarea are un număr de nivele egal cu numărul biților de cod;
- în urma partiționării rezultă un număr de 2^{m+1} ($= 4$) submulțimi având $2^{(k_0-m)}$ ($= 2^{(2-1)} = 2$) elemente
- biții de cod C_0C_1 selectează subsetul în care se află fazorul ce trebuie transmis, vezi fig. 10 și 11.
- bitul necodat C_2 selectează fazorul din subset (valorile logice boldite în ultimul rând al fig. 11), vezi fig.10
- această modalitate de mapare nu asigură invarianța la rotații de $k \cdot 90^\circ$;
- invarianța se asigură prin precodarea diferențială a biților informaționali care sunt codați pentru QAM (precodare care este realizată implicit de modulația DPSK), numai pentru anumite tipuri de coduri convoluționale, în funcție de structura diagramei trellis
- în urma partiționării distanța minimă dintre fazorii din același subset trebuie să fie maxim posibilă, pentru că bitul (biții) necodat(ți), care definește (definesc) fazorii unui subset, nu este “protejat” de codul corector.
- o tranziție în trellis este definită de biții de cod \rightarrow că o aceeași tranziție definită de biții de cod (C_0C_1) trebuie să conțină $2^{(k_0-m)}$ ($= 2$) tranziții paralele, definite de valorile posibile ale bitului (biților) necodați.
- pentru aceste modulații codate diagrama trellis în fazori va conține tranziții paralele; setul de tranziții paralele va fi reprezentat pe trellis prin submulțimea D_i , $i = 0, \dots, 2^{(m+1)}-1$, definită de cei $(m+1)$ biți de cod.
- pentru cazul particular discutat, diagrama trellis în fazori este prezentată în fig.12, vezi și fig. 8 și 11.

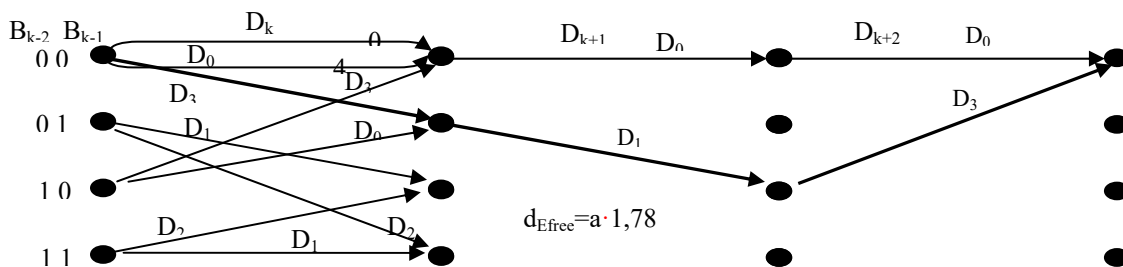


Figura 12
 Diagrama trellis în fazori a TCM din Ex. 7

- distanța minimă între căile din trellis, d_{Efree} , se obține acumulând distanțele euclidiene dintre perechile de tranziții care compun căile respective;
- deoarece fiecărei căi îi corespunde un subset, se va considera distanța minimă dintre un fazor dintr-un subset și un fazor din celălalt subset. Exemplu: distanța dintre subseturile D_0 și D_3 va fi egală cu distanța dintre fazorii F_0 și F_7 (numiți reprezentanți ai subseturilor)

- valoarea distanței d_{Efree} pentru trellisul din fig. 12 va fi:

$$d_{Efree}^2 = d^2(D_0, D_3) + d^2(D_0, D_1) + d^2(D_0, D_3) = 2 \cdot 4a^2 \sin^2 \frac{\pi}{8} + 2a^2 = 3,17a^2 \Rightarrow d_{Efree} = 1,78a \quad (27)$$

- d_{Efree} este o rezultantă între cod și modalitatea de mapare; codul alege calea minimă, iar maparea asigură mărirea distanței între reprezentanții submultimilor (tranzițiile paralele) ce fac parte din calea aleasă de cod.

- *Observație: pentru codurile nerecursive, uneori o mapare "oportunistă" dă o d_{Efree} mai mare, ex. faceți rocade între D_1 și D_3 în fig. 11 ($d_{Efree}^2 = 4.586$); pentru codurile recursive, în general MSP asigură d_{Efree} maximă.*

- valoarea absolută a SNR la care se asigură o valoare p_e impusă, p_{e0} , se obține cu relația (28) în care cu SNR_{ref} s-a notat valoarea SNR la care modulația necodată cu același debit binar util, folosită ca referință, asigură valoarea p_{e0} .

$$SNR_{TCM-pe0} = SNR_{ref-pe0} - C_G \quad (28)$$

Tipuri de erori ce apar în trellisul cu tranziții paralele

a. erori datorate alegerii unei căi incorecte – probabilitatea de apariție a acestor erori este dată de (14) și depinde de distanța cumulată minimă între căi, d_{Efree} ; aceste erori afectează biții informaționali codăți.

b. erori datorate alegerii greșite a căii paralele într-o tranziție; probabilitatea de apariție a acestor erori se calculează utilizând (13) și depinde de distanța minimă între tranzițiile paralele d_p . Acest tip de eroare afectează biții necodați.

- distanța minimă între tranzițiile paralele este distanța minimă dintre fazorii aflați în același subset; minimumul se aplică la toate subseturile. Ea este egală cu distanța corespunzătoare ultimului nivel al partiționării MSP, vezi fig. 11.

- o mapare corectă asigură o $d_p \geq d_{Efree}$; **această este echivalent cu a asigura biților informaționali necodați o probabilitate de eroare cel mult egală cu a biților informaționali codăți, vezi relația (20).**

Distanța minimă în trellis

- probabilitatea globală de eroare în trellis este dictată de probabilitatea de eroare cea mai mare dintre cele două tipuri de erori, vezi (15);

- deci în calculul câștigului codării (23), distanța minimă în trellis d_{mc} , va fi minimumul dintre d_{Efree} și d_p :

$$d_{mc} = \min(d_{Efree}, d_p); \quad (29)$$

- calculul câștigului codării pentru TCM din exemplul 7

- distanța $d_{Efree} = 1,78a$, vezi (27), iar $d_p = 2a$, vezi fig.11, deci d_{mc} va fi:

$$d_{mc} = \min(d_{Efree}, d_p) = \min(1,78a, 2a) = 1,78a \quad (30)$$

- aplicând (23) și considerând $P_{mc} = P_{mn} = a^2$ și $d_{mn} = a\sqrt{2}$, vezi sfârșitul paginii 6, avem:

$$C_G[\text{dB}] = 20 \lg \frac{d_{mc}}{d_{mn}} - 10 \lg \frac{P_{mc}}{P_{mn}} = 20 \lg \frac{1,78a}{\sqrt{2} \cdot a} - 10 \lg \frac{a^2}{a^2} \approx 1 \text{ dB} \quad (31)$$

- valoarea SNR pentru care se asigură $p_{e0} = 10^{-5}$ este $SNR_c \big|_{10^{-5}} = SNR_n \big|_{10^{-5}} - C_G = 12,5 - 1 = 11,5 \text{ dB}$

- valoarea scăzută a C_G nu justifică decât parțial complexitatea implementării TCM; acest fapt se datorează capacității scăzute de corecție a codului (doar $K = 3$), dar și mapării MSP.

- se observă că **biții necodați vor avea o probabilitate de eroare mai mică decât biții codăți, datorită faptului că $d_p > d_{Efree}$**

- pentru maparea „oportunistă”, $d_{efree} \approx 2,14 > d_p \rightarrow C_G$ asigurat este $C_G = 3 \text{ dB}$, iar $SNR_c \big|_{10^{-5}} = 9,5 \text{ dB}$, iar biții necodați vor avea o probabilitate de eroare mai mare decât cei codăți - **vezi mai jos și calcul pe tablă.**

Modalități de mărire a câștigului codării dacă rata codului este menținută constantă

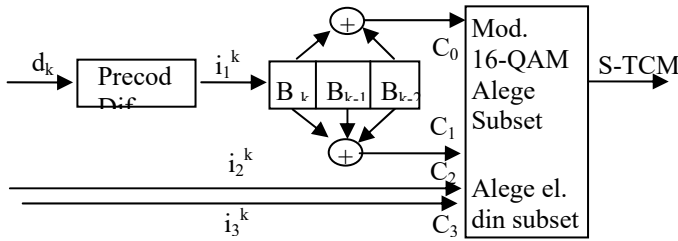
- pentru a mări d_{Efree} , și implicit capacitatea de corecție a codului, există două posibilități, dacă $R_c = \text{const.}$:

a. modificarea polinoamelor generatoare ale codului cu păstrarea constrângerii; această soluție poate duce unori la creșterea lui d_{Efree} , printr-o altă alocare subset-tranziție în trellis. Nu implică creșterea complexității implementării. – **de exemplificat cu D_1, D_3, D_1 în fig.11 și 12. Este echivalentă cu înlocuirea mapării MSP, cu o mapare „oportunistă” - vezi mai sus**

b. mărirea constrângerii codului; conduce la mărirea numărului de stări în trellis ($S = 2^v = 2^{K-1}$), fapt ce duce la creșterea lui d_{Efree} prin creșterea numărului de tranziții conținute de calea cu d_{Efree} și/sau prin creșterea distanței dintre perechile de tranziții. Implică însă creșterea complexității de implementare a TCM.

- constrângerea codului merită mărită doar până la valoarea care asigură $d_{Efree} = d_p$.

Temă: determinați câștigul maxim care se poate obține de modulația TCM din exemplul 7. de mai sus



Modulații TCM cu mai mulți biți necodați și cu asigurarea invarianței la rotații de $k \cdot 90^\circ$

- aceste modulații vor fi prezentate prin **Exemplul 8**
- considerăm o modulație TCM $R=3/4$ formată din c.c. cu $R_c=1/2$ cu $K=3$ și $G=[5,7]_8$ și 16-QAM.

Figura 13. Schema bloc a codorului TCM din Ex. 8.

- deoarece numărul biților informaționali codati $m =$

$1, \rightarrow k_0-m = 2$ biți info vor fi transmiși necodați.

- modulația necodată care asigură același debit binar util este 8-PSK, care are $P_{mn} = 1 \cdot a^2$ și $d_{mn} = 2a \cdot \sin(\pi/8) \approx 0,765a$, vezi fig 11.

- alocarea cuadrabit-fazor prin metoda MSP este prezentată în figura 14; aceasta va avea două nivele (avem doi biți de cod), iar cele 4 submulțimi rezultate vor avea câte $2^{k_0-m} = 2^2 = 4$ fazori.

- distanța minimă a constelației folosite pentru transmisia codată este $2a$, iar puterea medie $P_{mc} = 10a^2$.

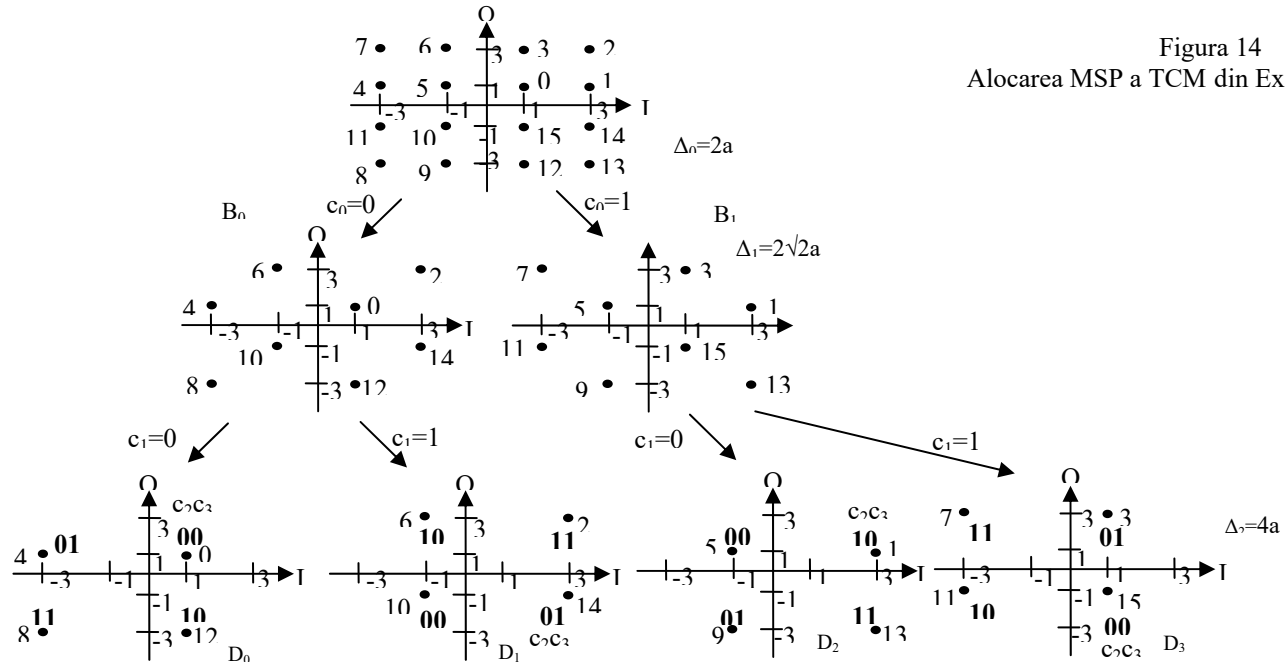


Figura 14
Alocarea MSP a TCM din Ex. 8

- pentru calcularea d_{Efree} avem nevoie de diagrama trellis în fazori, fig.15.

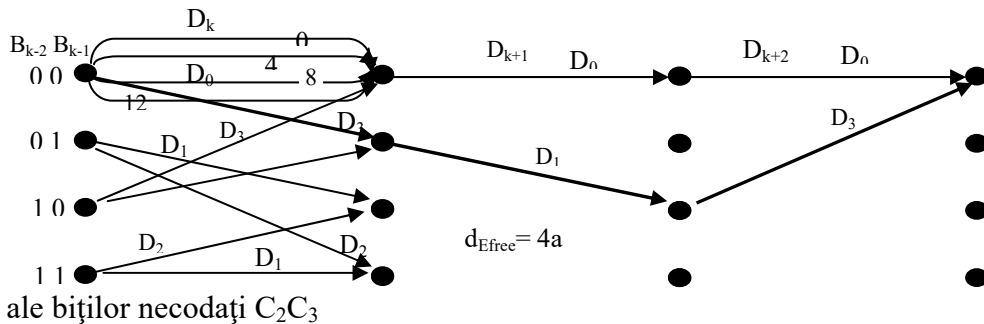


Figura 15.

Diagrama trellis în fazori a TCM din Ex. 8

- - fiecare tranziție a trellisului în fazori va fi constituită din 4 tranziții paralele, definite de cei 4 fazori care corespund combinațiilor posibile

- distanța d_{efree} se calculează folosind fig.15, și fig.14 pentru calculul distanțelor între subseturi:

$$d_{Efree}^2 = d^2(D_0, D_3) + d^2(D_0, D_1) + d^2(D_0, D_3) = 2 \cdot (2a)^2 + (2\sqrt{2} \cdot a)^2 = 16a^2 \Rightarrow d_{Efree} = 4a \quad (32)$$

- comparând cu Ex. 7, în care se folosește același cod corector, d_{Efree} este sensibil mai mare datorită constelației folosite; aceasta va conduce la un câștig mult mai mare adus de codul corector (nu de modulația codată!!)

- distanța între tranzițiile paralele este $d_p = 4a$, vezi fig. 14 și deci distanța minimă în TCM este:

$$d_{mc} = \min(d_{free}, d_p) = \min(4a, 4a) = 4a \quad (33)$$

- în acest caz biții de cod și cei necodați vor avea aceeași probabilitate de eroare.

- câștigul codării se obține folosind (23):

$$C_G[\text{dB}] = 20 \lg \frac{d_{mc}}{d_{mn}} - 10 \lg \frac{P_{mc}}{P_{mn}} = 20 \lg \frac{4a}{0,765a} - 10 \lg \frac{10a^2}{a^2} \approx 14,36 \text{ dB} - 10 \text{ dB} = 4,36 \text{ dB} \quad (34)$$

- distanța d_{mc} mai mare adusă de cod asigură un câștig foarte mare, 14,36 dB, dar ea este obținută și datorită utilizării unei constelații codate în care distanțele sunt mult mai mari, 16-QAM fig. 14, față de 8-PSK, fig. 11, pentru varianta necodată.

- creșterea distanțelor în constelație se traduce prin creșterea puterii medii, și de aceea efectul acestei creșteri este micșorat (al doilea termen în (34) – 10 dB), rămânând un $C_G \approx 4,36$ dB.

- valoarea SNR_c pentru care se asigură o valoare p_{et} impusă este $SNR_c|_{pet} = SNR_n|_{pet} - C_G$ care pentru $p_{et} = 10^{-5}$ dă $SNR_c = 17.8 - 4.36 = 13.44$ dB; $p_{e-8-PSK}(17.8 \text{ dB}) = 1 \cdot 10^{-5}$; debitul **util** asigurat este $D = 3f_s$

Temă: - determinați câștigul codării, dacă folosiți pentru transmisia necodată constelația 8-QAM formată din fazorii cu etichete pare ai constelației 16-QAM din fig. 14. Faceți legătura cu relația (28)

- Determinați C_G maxim ce poate fi asigurat de configurația din Ex.8. Justificați prin calcule și raționament.

Asigurarea invarianței la rotații de $k \cdot 90^\circ$

- invarianța la rotații de $k \cdot 90^\circ$ se asigură în mod diferit pentru biții informaționali codați și cei necodați:

- biții informaționali codați sunt precodați diferențial înainte de a fi introduși în codorul convoluțional – vezi cursul de QAM – aceasta doar în cazul în care trellisul codului utilizat păstrează invarianța asigurată de precodarea diferențială, proprietate pe care o au numai anumite tipuri de coduri

- biții informaționali necodați sunt mapați Gray în fiecare subset, vezi fig. 14, pentru a reduce probabilitatea de eroare de bit. Pentru a asigura invarianța, combinațiile de biți alocate unui fazor se rotesc cu $k \cdot 90^\circ$ de la un subset la altul astfel încât fazorilor care au același modul și diferă doar prin defazaje de $k \cdot 90^\circ$, să le corespundă același grup de biți necodați. Spre exemplu, fazorului 4, subset D_0 (fig.14) îi corespunde dibitul 00. Același dibit va corespunde și fazorilor 14 (subset D_1), 9 (subset D_2) și 3 (subset D_3).

- modalitățile de asigurare a invarianței biților codați și necodați la rotații de $k \cdot 90^\circ$ sunt similare cu cele utilizate în cazul modulațiilor necodate, dacă codul convoluțional are această proprietate. Dacă nu, pentru biții informaționali codați nu se poate asigura invarianța la rotații de $k \cdot 90^\circ$.

Reguli euristice pentru MSP

- G.Ungerboeck a dat următoarele reguli euristice (nedemonstrate riguros, ci rezultate din analiza unui număr relativ ridicat de exemple):

- Fazorii constelației trebuie utilizați cu probabilități egale (necesitatea scramblării pentru a asigura aceeași probabilitate de apariție a multibiților)

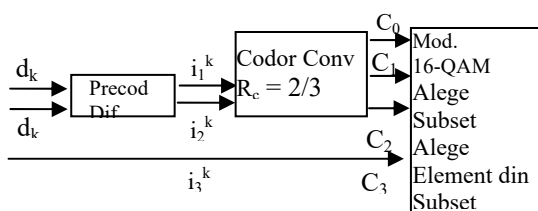
- Tranzițiilor care pleacă din (converg în) același nod al trellisului li se alocă fazori din submulțimile B_0 sau B_1 . De ex. în fig. 14, celor două tranziții care pleacă din nodul 00 li se alocă fazori din submulțimea B_0 , subsetul D_0 sau din B_1 , subsetul D_3 .

- Tranzițiilor paralele (dacă apar) trebuie să li se aloce fazori separați prin distanțele maxim posibile în constelație pentru că aceste tranziții sunt generate de biți necodați, neprotejați direct de codul corector; aceasta are ca scop asigurarea condiției $d_p \geq d_{Efree}$.

- această metodă de mapare asigură d_{Efree} maximă dacă se utilizează coduri convoluționale recursive

Mărirea câștigului codării prin schimbarea ratei codului corector

- dacă rata codului corector este păstrată $R_c = 1/2$, creșterea (cu efect) a d_{Efree} este limitată de valoarea d_p , care este impusă de constelație (la valoarea $4a$ în cazul particular studiat).



- pentru a mări valoarea d_p trebuie mărit numărul nivelelor partiționării, ceea ce implică mărirea R_c .

- dacă în Ex.8, se va folosi un cod cu $R_c = 2/3$ și aceeași constelație 16-QAM, în urma codării vom avea 3 biți de cod și numai 1 bit necodat, vezi schema bloc din fig. 16.

Figura 16. Schema bloc a TCM $R = 3/4$ cu cod corector $R_c = 2/3$

- rezultă că partiționarea va avea 3 nivele, obținându-se 8 subseturi a câte 2 fazori, vezi fig. 17(tablă).

- distanța între fazorii aceluiași subset, d_p va crește de $\sqrt{2}$ ori, ajungând pentru Ex. 8 la $d_p = 4\sqrt{2} \cdot a$

- dacă codul cu $R_c = 2/3$ utilizat va avea o constrângere suficient de mare ($K > 4$) pentru a asigura $d_{Efree} \geq d_p$ atunci d_{mc} va fi limitată de d_p .

Temă: - calculați care e câștigul maxim al codării ce se poate obține la o TCM $R = 3/4$ și 16-QAM, față de 8-PSK, dacă se folosește un cod cu $R_c = 2/3$. De ce nu se folosește un cod cu $R_c = 3/4$?

- în multe aplicații se folosesc coduri sistematice; dacă codurile sistematice nerekursive au performanțe slabe, adică asigură un d_{Efree} mic, atunci se folosesc codurile sistematice recursive care asigură d_{Efree} mare.

- utilizarea unuia sau altuia din cele două tipuri de coduri, trebuie conjugată cu tipul de mapare (MSP sau Gray) astfel încât să se asigure o d_{mc} maxim posibilă.

Performanțe ale TCM

- tabelul 2 prezintă în rezumat câștigurile maxime ale codării pentru modulații TCM cu $R = k_0/(k_0+1)$ care folosesc coduri convoluționale cu $R_c = m/(m+1)$. Se indică modulațiile codate și necodate folosite precum și raportul între d_{Efree} obținut și distanța minimă Δ_0 a constelației folosită de varianta codată.

Constr. K	m	R= 3/4; 16Q/8P	R= 4/5; 32Q/16Q	R= 5/6; 64Q/32Q	d_{Efree}^2 / Δ_0^2
3	1	4,36 dB $d_p \geq d_{Efree}$	3,01 dB $d_p < d_{Efree}$	2,81 dB $d_p < d_{Efree}$	4
4	2	5,33 dB	3,98 dB	3,77 dB	5
10	2	7,37 dB	5,81 dB	5,81 dB	8

Tabelul 2. Câștigurile codării unor TCM – cf. G.Ungerboeck

- C_G crește odată cu creșterea constrângerii codului, dar odată cu creșterea lui K peste 4, creșterea câștigurilor codării se obține cu o creștere semnificativă a complexității implementării

- creșterea ratei TCM conduce la scăderea C_G pentru o aceeași K, datorită a două cauze:

- mărirea constelației; aceasta conduce la scăderea lui Δ_0 între fazorii constelației, adică la scăderea d_{Efree} , dacă se păstrează aceeași putere medie a constelației codate P_{mc} .
- creșterea numărului de biți necodați $k_0-m \rightarrow$ scăderea $d_p \rightarrow$ scăderea d_{min} .

Decodarea semnalelor TCM folosind algoritmul lui Viterbi în d_E

Decizie hard vs. decizie soft

- deoarece demodolatoarele QAM furnizează coordonatele I_r și Q_r ale fazorului recepționat f_r , există două variante posibile de utilizare a algoritmului lui Viterbi (VA) pentru decodarea codului convoluțional:

a. Decizia hard

- circuitul de decizie alege fazorul, $f_i (I_i, Q_i)$, din constelație aflat la distanța euclidiană cea mai mică față de fazorul recepționat; această operație se efectuează în fiecare perioadă de simbol.
- apoi circuitul de demapare extrage multibitul codat mapat pe acel fazor.
- decodorul Viterbi extrage biții informaționali care au fost codați, folosind multibiții furnizați în fiecare perioadă de simbol de circuitele de decizie hard și demapare. *Metrica folosită de VA cu decizie hard este distanța Hamming, d_H .*

b. Decizia soft

- Coordonatele fazorului demodulat în fiecare perioadă de simbol, $f_i (I_i, Q_i)$, sunt introduse direct în decodorul cu VA, care va extrage direct biții informaționali codați. *Metrica folosită de VA cu decizie soft este distanța Euclidiană, d_E . O altă metrică ce poate fi folosită o constituie setul de probabilități a posteriori.*

- probabilitatea ca să se fi emis fazorul f_i , dacă s-a recepționat fazorul f_r , este, vezi (13):

$$p(f_r / f_i) = p(I_r / I_i) \cdot p(Q_r / Q_i) = \frac{1}{2\pi \cdot \sigma^2} \cdot e^{-\frac{(I_r - I_i)^2 + (Q_r - Q_i)^2}{2\sigma^2}} = \frac{1}{2\pi \cdot \sigma^2} \cdot e^{-\frac{d_E^2(f_r - f_i)}{2\sigma^2}} \neq 1 \quad (35)$$

- decizia hard alege fazorul cel mai apropiat, în sensul d_E , ceea ce este echivalent cu:

$$p(f_r / f_i) = 1 \text{ si } p(f_r / f_j) = 0 \text{ pt } j \in 0, \dots, N-1 \text{ } j \neq i \quad (36)$$

- prin neglijarea valorilor probabilităților ca fazorul recepționat să fi provenit din alți fazori decât fazorul f_i aflat la cea mai mică d_E , decizia hard „distruge (neglijează)” informații existente în semnalul recepționat. Aceasta conduce la o probabilitate de eroare mai mare, după decodarea cu VA în d_H , în special la valori mici ale SNR unde aproximațiile din (36) sunt destul de mari.

- decizia soft utilizează nivele demodulate I_r și Q_r , nu cele decise I_i și Q_i ; ea nu face aproximarea de mai sus, ci folosește informațiile din semnalul recepționat (coordonatele acestuia), ceea ce conduce la valori mai mici ale BER în special la valori mici ale SNR.

- mai există o variantă de decizie soft care utilizează ca metrică *probabilitatea a posteriori*- se va discuta atât în continuare cât și la laborator

Etapele decodării cu VA cu decizie soft (în d_E)

- etapele de mai jos se referă la decodarea TCM fără biți necodați

0. Generarea în receptor a diagramei trellis, în fazori, a codului utilizat.

1. Calculul d_E dintre fazorul recepționat și fazorii din fiecare subset

2. Calculul distanței cumulate al fiecărei căi care se termină în starea S_0

3. Alegerea supraviețuitorului de stare al stării S_0

4. Actualizarea registrului ce conține istoricul căii ce se termină în starea S_0 la tactul k

5. Se efectuează etapele 2, 3, 4 pentru toate stările $S_1, \dots, S_2^{(K-1)}-1$

6. Alegerea supraviețuitorului de pas (tact)

7. Extragerea bitului (biților) informațional(i) din registrul cu istoricul căii care a fost aleasă la etapa 6.

8. Shiftarea cu un pas a tuturor regiștrilor care conțin istoricul căilor la tactul curent.

0. Generarea în receptor a diagramei trellis, în fazori, a codului utilizat.

- poate fi generată tabelar sau recursiv. Pentru codul utilizat mai sus, fig. 8 și 9, dacă se cunoaște starea finală a unei tranziții S_j , $j = 0, \dots, 3$, cele 2^m stări inițiale S_l din care există tranziții spre S_j , sunt date de (37), în funcție de valoarea multibitului informațional codat (în acest caz $d \leftrightarrow$ valoarea bitului 0 sau 1).

$$S_j = (2 \cdot S_l)_{\text{mod } 2^{K-1}} + d \quad (37)$$

- generarea trellisului mai trebuie completată cu tabelul fazorilor (I și Q) mapați pe fiecare tranziție în trellis
- **tabelul fazorilor T_F** va conține S_j, S_l, I_{jl}, Q_{jl} (eventual și eticheta fazorului), și cei m biți info care au generat tranziția $l \rightarrow j$.

- pentru codurile „punctured” se va genera digrama trellis a codului „părinte”.

1. Calculul d_E dintre fazorul recepționat și fazorii din fiecare subset

- se face utilizând relația (38) pentru toți fazorii din constelație

$$d_{Ei}^2 = (x_r - x_i)^2 + (y_r - y_i)^2; i \in \{0, \dots, N-1\} \quad (38)$$

- nu este necesară determinarea distanței euclidiene (extragerea radicalului), deoarece algoritmul VA folosește relații de ordine între distanțele cumulate, iar pentru valori pozitive ale argumentului funcția x^2 păstrează relația de ordine, fiind strict crescătoare pentru valori pozitive ale argumentului.

- **distanțele sunt stocate într-un tabel T_D** , în funcție de eticheta fazorului, și sunt actualizate la fiecare perioadă de simbol.

2. Calculul distanței cumulate al fiecărei căi care se termină în starea S_0

- se calculează distanța cumulată a fiecărei căi, din cele 2^m care se termină în starea $S_j (S_0)$, folosind (39);

$$d_{cjt}^2 = d_{cl}^2 + d_{Ejlt}^2 \quad (39)$$

- în (39) s-au notat:

- cu d_{cjt}^2 distanța cumulată a fiecărei căi care se termină în starea $S_j (S_0)$, - distanța „totală curentă”
- cu d_{cl}^2 distanța cumulată a căilor „supraviețuitor” care s-au terminat în perioada de simbol anterioară în stările S_l din care pornesc tranziții spre starea $S_j (S_0)$. - distanța „totală anterioară”
- cu d_{Ejlt}^2 distanța euclidiană între fazorul recepționat și cel corespunzător tranziției de la starea S_l la starea $S_j (S_0)$ – citită din tabelul T_D actualizat la pasul 1 folosind tabelul T_F de la pasul 0. - „distanța de tranziție

3. Alegerea supraviețuitorului de stare (de nod) al stării S_0

- se păstrează doar calea care are distanța cumulată minimă dintre cele care se termină în starea $S_j (S_0)$

- se face conform relației:

$$d_{cj}^2 \text{ min} = \min(d_{cj-t}^2); t = 1, \dots, 2^m \quad (40)$$

- calculul supraviețuitorului de stare se face combinând pașii 2 și 3, pentru fiecare tranziție (valoare a lui t); distanța (39) odată calculată este comparată cu cea anterioară și se păstrează doar cea mai mică.

- trebuie de asemenea păstrat indexul stării S_l din care provine tranziția care încheie calea (terminată în S_j) cu distanța cumulată minimă – **în registrul R_T**

- mai trebuie păstrat bitul (biții) informaționali care au generat tranziția ce încheie calea cu distanță cumulată minimă – **citiți din T_F**

4. Actualizarea registrului ce conține istoricul căii care se termină în starea S_0 la tactul k

- cei m biți informaționali care au generat ultima tranziție a căii ce a fost aleasă supraviețuitor de nod sunt introduși într-un registru ce conține „istoricul” căii. Este un registru cu indexul egal cu cel al stării (nodului) care conține **w grupuri de m biți informaționali** care au generat tranzițiile ce compun calea.

- w este lungimea ferestrei de decodare și are o valoare cuprinsă între $3-5 \cdot K$.

5. Se efectuează etapele 2, 3, 4 pentru toate stările $S_1, \dots, S_2^{(K-1)}-1$

- se calculează distanțele euclidiene cumulate, supraviețuitorul de nod și se actualizează regiștrii „istoric” pentru toate celelalte $2^{K-1}-1$ stări ale trellisului.

- de menționat că distanțele cumulate ale supraviețuitorilor de stare trebuie stocate în două seturi de 2^{K-1} re-

giștri, $R_{dc}i$. Unul, $R_{dc}0$, va păstra valorile distanțelor cumulate al căilor supraviețuitor la sfârșitul perioadei de simbol $k-1$, iar celălalt, $R_{dc}1$, va conține distanțele cumulate obținute după pasul k .

- în mod similar istoricul căilor supraviețuitor de nod trebuie păstrat în două seturi, $R_{IC}0$ și $R_{IC}1$ de câte 2^{K-1} regiștri.

- utilizarea a două seturi de regiștri este necesară, deoarece după actualizările distanței cumulate și ale istoricului căii ce se termină în S_0 la tactul k , datele aflate în regiștrii ce conțineau informațiile la tactul $k-1$ sunt necesare pentru calculele efectuate pentru starea S_2 (fiecare stare trimite/primește tranziții spre/de la alte 2^m stări).

- cele două seturi de regiștri ale supraviețuitorului de nod, pentru distanța cumulată și respectiv pentru istoric, sunt utilizate alternativ ca „memorie de informații noi” și „memorie de informații vechi”.

6. Alegerea supraviețuitorului de pas (tact)

- se alege calea care are distanța cumulată minimă, dintre căile supraviețuitor de stare. Indexul stării care are calea supraviețuitoare cu distanța cumulată minimă, **stocat în R_{SP}** , va fi folosit la pasul 7.

7. Extragerea bitului (biților) informațional(i) din registrul cu istoricul căii supraviețuitoare de pas (aleasă la etapa 6). Shiftarea cu un pas a tuturor regiștrilor care conțin istoricul căilor la tactul curent.

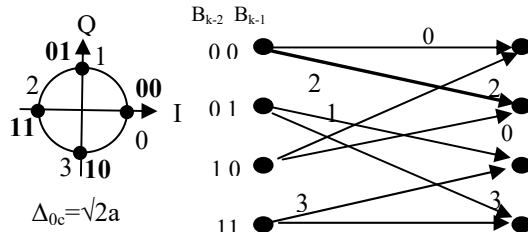
- din istoricul căii determinată la pasul 6 se extrag ultimii m biți care reprezintă biții informaționali decodați în perioada de simbol k .

- apoi biții din setul de 2^{K-1} regiștri ce conțin istoricul actualizat la tactul k sunt shiftați cu m poziții pentru a șterge cei m biți info care au generat tranzițiile căilor supraviețuitoare în urmă cu w perioade de tact.

Exemplul 9

- se consideră modulația TCM din exemplul 6 ($R = 1/2$, $R_c = 1/2$, $K = 3$, $G = [5,7]_8$, 4-PSK)

- maparea fazorilor și diagrama trellis în fazori sunt prezentate în fig. 18 și 19.



Figurile 18 și 19

Maparea dibitului pe constelație și diagrama trellis în fazori

- se consideră că distanțele cumulate după 4 perioade de tact sunt $d_{c0} = 3,12$; $d_{c1} = 4,21$; $d_{c2} = 3,11$; $d_{c3} = 5,17$;

- fazorul recepționat în perioada $k+4$ este $f_r (-0,8; 0,2)$

- diagrama trellis a evoluției decodurului VA este dată în fig. 20

- fereastra de decodare are lungimea $w = 5$, pentru simplitate.

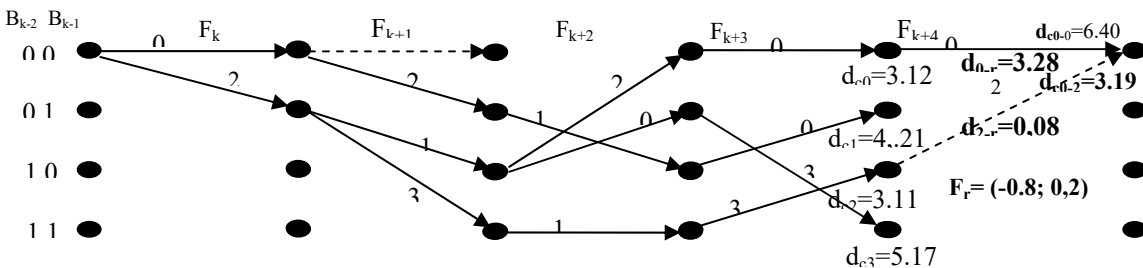


Figura 20

Diagrama trellis a evoluției decodurului VA pentru Exemplul 9

- **etapa 1:** - dist. euclid. între fazorii constelației și f_r se calculează cu (11) și sunt (se pun în tabelul T_D):

$$d_{E0-r}^2 = (I_r - I_0)^2 + (Q_r - Q_0)^2 = (-0,8 - 1)^2 + (0,2 - 0)^2 = 3,28; \quad (41)$$

$$d_{E1-r}^2 = 1,28; \quad d_{E2-r}^2 = 0,08; \quad d_{E3-r}^2 = 2,08$$

- **etapa 2** - în starea S_0 se poate ajunge din stările care satisfac ecuația (37), rescrisă pentru $j = 0$

$$S_0 = (2 \cdot S_1)_{\text{mod } 2^{3-1}} + d \Rightarrow S_1 = S_0 \text{ sau } S_2 \text{ - bit info } d = 0 \quad (42)$$

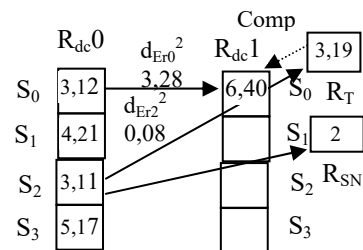


Figura 21 Stocarea distanțelor cumulate ale căii și alegerea supraviețuitorului de stare

- stările din care se poate ajunge în starea S_0 mai pot fi citite din tabelul T_F .

- distanțele cumulate ale căilor care ajung în starea S_0 se calculează cu (39) și sunt date în (43), vezi fig. 21:

$$d_{c00}^2 = d_{c0}^2 + d_{E00t}^2 = 3,12 + 3,28 = 6,40; \quad d_{c02}^2 = d_{c2}^2 + d_{E02t}^2 = 3,11 + 0,08 = 3,19; \quad (43)$$

- distanțele cumulate din perioada $k+3$ sunt stocate în setul de regiștri $R_{dc}0$, (păstrează valorile anterioare).

- distanțele cumulate din perioada $k+4$ vor fi stocate în setul de regiștri $R_{dc}1$, (păstrează valorile actualizate).

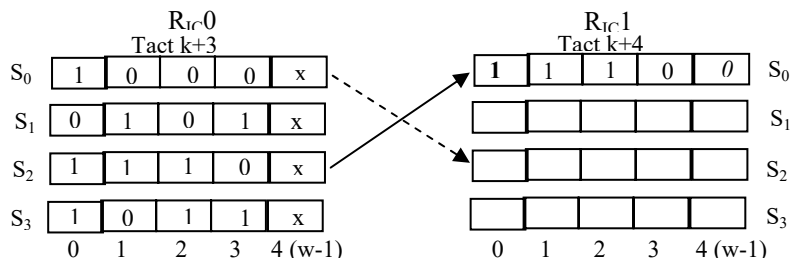
- prima distanță calculată d_{c00}^2 se pune în celula 0 a lui $R_{dc}1$, iar a doua d_{c02}^2 se pune într-o celulă buffer R_T .

- **etapa 3:** se execută comparația dintre ele iar valoarea cea mai mică, noua d_{c0}^2 , este plasată în celula 0 a lui R_{dc1} . Cealaltă valoare se va pierde, aceasta echivalând cu alegerea supraviețuitorului de nod

- se va păstra într-un registru indexul stării de unde provine calea supraviețuitoare de nod (2) - R_{SN} .

- **etapa 4 :** actualizarea istoricului căii supraviețuitor în starea S_0

- istoricul căii 2 aflat în celulele registrului 2 al setului R_{IC0} (care conține istoricul actualizat la perioada de simbol $k+3$), este shiftat în față cu o celulă (deci bitul cu x va fi „liber”) și este transferat în primele $w-1$ celule ale registrului 0 al setului R_{IC1} , care va conține istoricul căii supraviețuitoare a stării S_0 la tactul $k+4$.



- din tabelul inițial T_F se extrage bitul informațional care a generat tranziția $S_2 \rightarrow S_0$, care face parte din calea supraviețuitoare a stării S_0 , adică bitul **0** și se pune pe ultima poziție a registrului **0** din setul R_{IC1} , vezi fig. 22, caracter italic.

Figura 22 Actualizarea istoricului căii supraviețuitoare a stării S_0

- istoricul „vechi” (la tactul $k+3$) al căilor care trec prin S_0 și S_2 trebuie păstrat pentru că ar mai putea fi necesar în calculele efectuate pentru stările S_1 și S_2 .

- **etapa 5:** implică efectuarea etapelor 2, 3 și 4 pentru stările S_1 , S_2 și S_3 .

- calculele distanțelor cumulate ale supraviețuitorilor de stare și favorii generatori ai tranzițiilor prin care aceste căi au ajuns în stările S_1 , S_2 și S_3 sunt rezumate în fig.23.

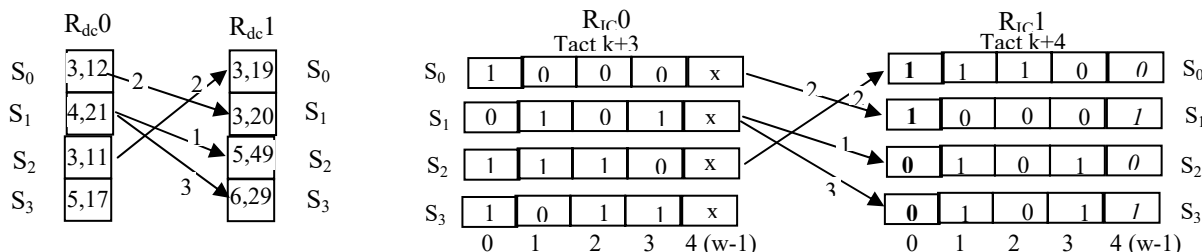


Figura 24 Actualizarea istoricului căilor supraviețuitoare ale stărilor S_0 , S_1 , S_2 , S_3

Figura 23 Actualizarea d_{Ec}^2 ale stărilor S_0 , S_1 , S_2 și S_3

- actualizarea istoricului căilor supraviețuitoare din stările S_0 , S_1 , S_2 și S_3 este rezumată în fig. 23.

- în figură sunt prezentați și favorii care au generat ultimele tranziții la tactul $k+4$, biții informaționali care au generat aceste tranziții (ultima poziție în fiecare registru – cu *italic*) și biții informaționali ce au generat tranzițiile aflate cu w ($= 5$) pași în urmă (cu **bold**).

etapa 6 - alegerea supraviețuitorului de pas (tact)

- din registrul R_{dc1} se alege starea care are distanța cumulată minimă (care corespunde unei plauzibilități maxime), adică starea S_0

- indexul ei se memorează pentru a fi folosit în etapa următoare - R_{SP}

- operația de determinare a stării cu d_{Ec} minimă poate fi realizată în cursul etapelor 2-5 prin utilizarea unui registru auxiliar.

etapa 7 - extragerea bitului (biților) informațional(i) din registrul cu istoricul căii supraviețuitoare de pas

- din setul de regiștrii R_{ICi} care conține istoricul actualizat la perioada de tact curentă, R_{IC1} , se extrag cei m (unu) biți aflați pe prima poziție (cu o întârziere de w perioade de simbol - **bold**) a registrului S_0 , calea determinată ca supraviețuitoare de pas, determinată în etapa 6, **valoarea lui R_{SP}** . Valoarea bitului decis cu decizie soft este 1 – **vezi reîntoacerea în trellis cu w perioade de simbol, fig. 20**

- apoi toți regiștrii setului R_{IC1} se shiftează la stânga cu o poziție, spre a elimina seturile de m biți corespunzătoare pasului $k+4$.

- în pasul următor, $k+5$, regiștrul R_{dc1} și setul R_{IC1} vor conține valorile „vechi”, iar în registrul R_{dc0} și în setul R_{IC0} se vor valorile actualizate corespunzătoare acestui pas.

- biții decodați vor fi livrați cu o întârziere de w perioade de tact de simbol (fereastra de decodare).

Observații:

1. folosirea VA pe un număr foarte mare de perioade de simbol poate conduce la obținerea unor valori ale distanțelor acumulate d_{ci}^2 care pot depăși capacitatea de stocare a celulelor seturilor de regiștrii R_{dc1} .

• rezultă schimbarea ordinii distanțelor prin depășirea capacității de stocare (overflow), celula cu distanța cumulată maximă va conține distanța cea mai mică, și implicit la alegerea greșită a supraviețuitorului de pas. – *de exemplificat*