

- fenomenul se propagă pe mai multe perioade de simbol conducând la apariția unor pachete lungi de erori, care necesită pentru corecție utilizarea unor coduri RS mari și cu putere de corecție ridicată, concatenate cu codurile convoluționale.
 - pentru reducerea probabilității de apariție a acestui fenomen, valorile d_{ci}^2 vor fi înmulțite cu un factor de scalare subunitar. Această operație schimbă valoarea distanței cumulate, dar nu afectează relația de ordine între distanțele cumulate.
- 2.** Dacă la un pas de decodare VA alege greșit supraviețuitorul, algoritmul are nevoie de un număr de pași pentru a reveni pe calea corectă.
- acest număr de pași depinde de secvența de biți transmisă și de valorile SNR pe durata acestui număr de pași. În perioadele de simbol în care VA alege o cale eronată, la decodorul furnizează un pachet de erori.
 - deci, în condițiile unui zgomot puternic de scurtă durată sau a unei desincronizări, VA *adaugă un număr suplimentar de erori* ceea ce conduce la apariția unui pachet de erori relativ lung. →
 - codurile convoluționale sunt, de obicei, concatenate cu coduri RS corectoare de pachete de erori.
 - pentru reducerea dimensiunii pachetului de erori între cele două coduri se introduce un interleaver (la emisie) și un deinterleaver (la recepție). Acesta are rolul de a „împrăștia” erorile, ducând la reducerea lungimii pachetelor de erori și implicit la creșterea ratei codului RS exterior destinat corectării pachetelor; înserează diversitate în timp
- 3.** Pentru a se reduce lungimea pachetului de erori ce poate să apară datorită alegerii greșite a căii în trellis, cât și pentru a reduce probabilitatea de depășire a capacității de stocare a R_{dcI} , mesajul transmis se „pachetizează”, prin aducerea trellisului în starea 0.
- pentru codurile nerecursive aceasta se realizează prin adăugarea, odată la L perioade de simbol, a K-1 biți informaționali de „0” (sau a combinațiilor de biți de intrare care duc trellisul în S_0);
 - pentru codurile recursive, metoda de aducere a trellis-ului în S_0 se va discuta la master.
- 4.** Pentru asigurarea unei ferestre de decodare cu lungimea w ($= 3-5K$) perioade de simbol și pentru a asigura ultimilor biți codați aceeași probabilitate de eroare la decodare, la sfârșitul pachetului este necesară transmiterea a w combinații de biți informaționali care aduc și mențin trellisul în S_0 .

Variante de implementare a algoritmului Viterbi

- pentru a reduce volumul de calcule impus de VA se pot face unele aproximări:

1. Folosirea modulului ca metrică în locul distanței euclidiene

- distanța între fazori se definește ca suma modulelor diferențelor pe cele două componente (43), în locul distanței euclidiene definită de (11)

$$d(r, f_i) = |I_r - I_i| + |Q_r - Q_i| \quad (43)$$

- folosirea acestei distanțe simplifică calculele și și scade probabilitatea de a obține valori ale distanței acumulate care să depășească capacitatea de stocare a celulelor seturilor R_{dcI}

- această aproximare conduce la scăderea performanțelor (puterii de corecție) a VA la valori mici ale SNR; pentru valori medii și mari ale SNR, puterea de corecție a VA nu este afectată semnificativ.

2. Alegerea unei aceleiași stări (de ex. S_0) ca și stare cu distanța cumulată minimă (supraviețuitor de pas)

- aceasta elimină operația de comparare a distanțelor cumulate, dar nu și celelalte operații.
- Performanțele VA nu sunt afectate semnificativ pentru valori medii și mari ale SNR; pentru valori mici ale SNR, performanțele VA scad semnificativ.

Decizia biților necodați

- în cazul modulațiilor TCM care mapează și biți necodați, decizia asupra valorilor logice ale acestor biți poate fi luată prin două metode:

- a. decizia hard
- b. decizia soft

a. Decizia hard a biților necodați

- implică estimarea prin decizie hard a fazorului recepționat, urmată de demaparea biților necodați.
- estimarea prin decizie hard a fazorului recepționat se face în fiecare perioadă de simbol prin calculul distanțelor euclidiene dintre fazorul recepționat (pe care sunt mapați și biții necodați) și fazorii constelației utilizate, urmată de alegerea fazorului constelației care se află la distanța minimă de cel recepționat.
- operația se poate face bidimensional utilizând relația (11)
- apoi se face demaparea biților necodați corespunzători fazorului decis, pe baza tabelului de demapare.
- biții necodați decizi pentru fiecare perioadă de simbol sunt stocați într-un registru care conține „istoricul”

biților necodați \mathbf{R}_{IBN} pentru a fi livrați simultan cu biți info codați transmiși în aceea perioadă de simbol.

- biți necodați decși prin această metodă *trebuie întârziati w perioade de simbol* pentru a se asigura livrarea lor simultană cu biții informaționali codați care au fos mapați pe același simbol (VA livrează biții informaționali decodați cu o întârziere de w perioade de simbol – fereastra de decodare!) → registrul \mathbf{R}_{IBN} are w celule, fiecare conținând k_0 -m biți.

- notând cu f_i fazorul aflat la distanță minimă de cel recepționat, probabilitatea să se recepționeze f_r dacă s-a emis f_i este dată de (13), reluată aici:

$$p(f_r / f_i) = p(I_r / I_i) \cdot p(Q_r / Q_i) = \frac{1}{2\pi \cdot \sigma^2} \cdot e^{-\frac{(I_r - I_i)^2 + (Q_r - Q_i)^2}{2\sigma^2}} = \frac{1}{2\pi \cdot \sigma^2} \cdot e^{-\frac{d_E^2(f_r - f_i)}{2\sigma^2}} \neq 1 \quad (44)$$

- prin decizia hard această probabilitate este considerată egală cu 1, ceea ce duce la apariția unor erori de simbol cu probabilitatea $1 - p(f_r/f_i)$.

- pentru valori mari ale SNR probabilitatea de eroare este scăzută, deoarece $p(f_r/f_i)$ are valori foarte apropiate de 1, dar la valori medii și mai ales la valori mici ale SNR probabilitatea de eroare a biților necodați devine considerabilă deoarece valorile $p(f_r/f_i)$ scad semnificativ.

b. Decizia soft a biților necodați

- utilizează informațiile oferite de VA prin decizia biților codați, ceea ce este echivalent cu determinarea subsetului din care provine fazorul (sau cu determinarea biților de cod, de ex. c_0 și c_1).

- probabilitatea de determinare corectă a subsetului depinde de distanța minimă între căile din trellis d_{Efree} , și este cu atât mai mare cu cât d_{Efree} are o valoare mai mare.

- probabilitatea de determinare corectă a fazorului din subset depinde de distanța minimă în subset, adică de d_p , care de obicei este mai mare sau egală cu d_{Efree} și mai mare decât distanța minimă în constelație, Δ_0 .

- calculul probabilității de eroare a biților necodați obținuți prin decizie soft este destul de complex, dar studiile arată că utilizarea deciziei soft asigură o probabilitate de eroare a biților necodați mai mică decât cea asigurată de decizia hard, la un același SNR. $P_{ebn} = p_{cbc} p_{ebn} + p_{ebc} p_{ebn2} - \text{vezi notițe pe tablă}$

Adaptarea algoritmului lui Viterbi pentru decizia soft a biților necodați

- la recepția unui fazor f_r în etapa 1, algoritmul lui Viterbi trebuia să determine distanța acestuia față de fazorul care a generat o tranziție în trellis. Dar în cazul modulațiilor cu biți necodați, există 2^{k_0-m} fazori care generează tranzițiile paralele.

- de aceea algoritmul trebuie să efectueze o etapă pregătitoare 0', înaintea etapei 1, și anume:

Etapa 0' – alegerea reprezentantului fiecărui subset și calculul d_E a subsetului în per. de simbol curentă;

- la recepția fazorului f_r din fiecare subset de fazori obținut prin MSP (și definit de biții de cod) se alege cel care are distanța euclidiană minimă față de fazorul recepționat. Acest fazor R_j se numește reprezentant al subsetului D_j la în perioada de simbol (tactul) k .

- operația se execută pentru toate cele 2^{m+1} subseturi, iar distanțele rezultatele se depun în 2^{m+1} celule de memorie, care au același index ca și cel al subsetului. Aceste celule formează memoria de distanțe a reprezentanților, \mathbf{M}_{dr} . Ea se actualizează la fiecare pas al decodării, adică la fiecare perioadă de simbol.

- la alegerea fazorului reprezentant al unui subset, biții informaționali necodați care definesc fazorul în cadrul subsetului se memorează într-un registru cu indexul subsetului în cauză - \mathbf{R}_{bnj} , $j = 0, \dots, 2^{m+1}-1$

- pentru cele 2^{m+1} subseturi există câte un registru care conține cei (k_0-m) biți necodați/perioadă de simbol, care se actualizează la fiecare perioadă de simbol, formând setul de regiștrii cu biții necodați ai reprezentanților curenți, \mathbf{R}_{bn} .

Etapa 4' – actualizarea istoricului căii în biți necodați

- o dată cu determinarea biților informaționali care au fost codați este necesară și determinarea biților informaționali necodați care a fost mapați pe fazorul decis.

- prin selectarea supraviețuitorului de nod, etapa 3, VA determină o cale în trellis care este stocată în registrul corespunzător stării din setul \mathbf{R}_{ICi} prin biții informaționali codați (și implicit prin subseturi)

- ultima tranziție a supraviețuitorului de nod este definită printr-un subset, vezi registrul T_F (f_{jl} , care în acest caz devine subset); odată subsetul cunoscut, din setul de regiștrii \mathbf{R}_{bnj} actualizați în etapa 0' se extrag biții necodați care corespund subsetului respectiv în perioada k de simbol.

- aceștia sunt introduși în ultima locație a registrului cu indexul stării din setul de regiștrii care conțin istoricul căilor în biți necodați, \mathbf{R}_{ICBni} , corespunzător perioadei curente de simbol.

- se utilizează două seturi de regiștri \mathbf{R}_{ICBni} în mod similar cu seturile \mathbf{R}_{ICi} ; ele au același mod de actualizare și de utilizare alternativă.

- regiștrii R_{ICBNi} diferă de R_{ICBi} doar prin faptul că se introduc pentru fiecare din cele w perioade de simbol un număr de $k_0 - m$ biți necodați.

Etapa 7' *Extragerea bitului (biților) informațional(i) din registrul cu istoricul în biți necodați al căii supraviețuitoare de pas (aleasă la etapa 6). Shiftarea cu pas a tuturor regiștrilor care conțin istoricul căilor în biți necodați la tactul curent.*

- din registul cu istoricul în biți necodați al căii determinată la pasul 6 se extrag ultimii $k_0 - m$ biți care reprezintă biții informaționali necodați aferenți simbolului decis în perioada de simbol k .

- apoi biții din setul de 2^{K-1} regiștri ce conțin istoricul actualizat la tactul k sunt shiftați cu $k_0 - m$ poziții pentru a șterge cei $k_0 - m$ biți informaționali necodați care au generat tranzițiile (dintre tranzițiile paralele) căilor supraviețuitoare în urmă cu w perioade de tact.

- biții necodați decși soft vor fi livrați simultan cu biții informaționali care au fost codați nemaifiind necesară întârzierea lor.

- un exemplu de decizie soft a biților necodați folosind informațiile oferite de VA va fi discutat la laborator.

Variante ale decodurului Viterbi în funcție de metrica utilizată

- pentru decodarea cu algoritmul lui Viterbi se pot utiliza următoarele metrici:

- distanța Hamming, d_H

- distanța euclidiană, d_E

- probabilitățile a posteriori ca un fazor recepționat să fi provenit din fazorii constelației utilizate sau cu probabilitățile a posteriori ale biților mapați pe fazorul recepționat f_r .

Decodor Viterbi în distanță Hamming

- necesită decizia hard a fazorilor recepționați și apoi demaparea biților „transportați” de aceștia;

- distanțele între tranzițiile din trellis și fazorul recepționat se calculează ca distanțe Hamming dintre combinațiile de biți codați care au generat tranzițiile și combinația de biți codați demapată de pe fazorul recepționat, fiind numere naturale.

- asigură performanțe mai slabe datorită aproximării descrise de ecuația (44).

Decodor Viterbi în distanță euclidiană

- a fost descris în paragrafele anterioare

- trebuie însă menționat faptul că probabilitatea condiționată $p(r/f_j)$ ca un fazor recepționat r să fi provenit dintr-un fazor dat f_j , descrisă de relația (13), este valabilă doar pentru un canal afectat de zgomotul gaussian. Pentru canale afectate de alte tipuri de zgomote sau distorsiuni, expresia (13) se modifică, ducând la modificarea expresiei distanței (nu se poate aplica întotdeauna $\ln(p(r/f_j))$ deoarece $p(r/f_j)$ nu mai are întotdeauna formă exponențială).

- de aceea, pentru generalizarea utilizării acestui algoritm s-a introdus decodorul cu VA în probabilități.

Decodor Viterbi în probabilități a posteriori

- acesta se bazează pe calculul probabilității *a posteriori* ca o succesiune de fazorii recepționați, calea recepționată u , să provină dintr-o succesiune permisă de fazorii, calea v din trellis, probabilitate descrisă de (14), pentru fiecare cale posibilă în trellis.

- probabilitățile a posteriori ale căii recepționate sunt determinate cu funcția de plauzibilitate (likelihood):

$$p(u|v) = \prod_{i=1}^w p(r_i|f_i) \quad (45)$$

- decodorul calculează aceste probabilități $p(r_i/f_i)$ pentru toți fazorii constelației, corespunzătoare distanțelor d_E între fazorul recepționat și fazorii constelației.

- distanța cumulată a unei căi se calculează făcând produsul probabilităților de mai sus, operație corespunzătoare calculului distanței d_{cjt}^2 .

- supraviețuitorilor de nod și de pas le corespund căile cu probabilități maxime.

- pentru a elimina operațiile de înmulțire (și uneori cele de exponențiere) o variantă a acestui algoritm calculează logaritmul funcției de plauzibilitate (log-likelihood), ceea ce conduce la necesitatea calculării unor sume din logaritmi probabilităților a posteriori:

$$\log p(u|v) = \sum_{i=1}^w \log p(r_i|f_i) \quad (46)$$

- această abordare prezintă un grad de generalitate mai ridicat, ea putând fi folosită pe o varietate mai mare de canale decât decodarea în distanță euclidiană.

- pentru alte canale decât cel gaussian, funcția densității de probabilitate are alte expresii, ceea ce modifică

expresia $p(f_i/f_j)$. Probabilitățile *a posteriori* pot fi calculate relativ simplu din aceste distribuții, vezi relația (47) mai jos, dar distanțele euclidiene nu pot fi calculate ușor

- decizia soft a biților necodați cu VA ce folosește această metrică va fi descrisă în paragraful destinat decodării codurilor punctured.
- variantele descrise mai sus se folosesc doar pentru modulații codate cu coduri complete (nepuncturate).
- această metrică asigură o mai mare aplicabilitate și câștiguri ale codării mai mari algoritmului Viterbi.

Utilizarea decodurii Viterbi la decodarea TCM codată cu coduri convoluționale „punctured”

- codurile convoluționale „punctured” permit modificarea (creșterea) ratei codării, plecând de la un cod părinte cu rata de codare minimă, folosind măști de puncturare, vezi tabelul 1, pagina 2.
- pentru a se asigura o implementare cât mai simplă se urmărește ca decodarea tuturor codurilor „punctured” să se facă folosind diagrama trellis a codului părinte care le-a generat.
- pentru aceasta, la recepție se urmărește refacerea fluxului original de biți codați cu codul părinte prin trei operații:
 - extragerea biților codați, sau a unor informații referitoare la aceștia (probabilitățile *a posteriori*), din favorii recepționați
 - poziționarea biților (codați cu codul punctured) extrași pe pozițiile care le-au avut în fluxul codat cu codul părinte
 - plasarea pe pozițiile corespunzătoare biților ce nu au fost transmiși, din fluxul de biți de cod generați de codul părinte, a unor valori corespunzătoare elementelor neutre ale operației de calcul a metricii, pentru ca aceștia să nu influențeze calculul distanțelor de tranziție și al celor cumulate, deoarece receptorul nu are “informații” asupra lor, ei nefiind transmiși.

Exemplu:

- considerând un cod punctured cu $R_p = 3/4$ obținut din codul părinte cu $R = 1/2$ cu masca descrisă în coloana 3 a tabelului 1, fiecare grup de 3 biți informaționali generează 6 biți codați, linia 1 tabelul 3, din care prin puncturare sunt modulați doar 4 pe favorii f_k și f_{k+1} , linia 3 tabelul 3.
- la recepție se extrag informații privitoare la biții modulați, care apoi sunt poziționate pe pozițiile originale (folosindu-se masca de puncturare), iar pe pozițiile ce corespund biților codați „lipsă” se pun elemente notate cu X, corespunzând elementului neutru.

Tabelul 3. Refacerea fluxului de biți la decodarea codurilor punctured

- operațiile de extragere a biților info codați sau a informațiilor privitoare la aceștia și valoarea elementului neutru depind de metrica folosită în algoritmul Viterbi, (d_H , d_E sau probabilități a posteriori) și de aceea vor fi particularizate pentru aceste metrici.

R↓ perioadă de bit→	k	k+1	k+2
1/2 - emisie	c_0^k c_1^k	c_0^{k+1} (1) c_1^{k+1} (1)	c_0^{k+2} (1) c_1^{k+2} (1)
3/4 - puncturat emisie	c_0^k c_1^k	c_0^{k+1} (0) c_1^{k+1} (1)	c_0^{k+2} (1) c_1^{k+2} (0)
Modulat emisie	c_0^k (1) c_1^k f_k	c_0^{k+2} c_1^{k+1} f_{k+1}
Flux refăcut R=1/2 rec.	c_0^k c_1^k	X c_1^{k+1}	c_0^{k+2} X

Decodarea codurilor punctured în d_H

- prin decizie hard și demapare se obțin biții mapați pe favorii recepționați.
- folosind masca de puncturare se poziționează biții extrași pe pozițiile originale din fluxul de biți codați cu codul părinte
- deoarece operația de calcul a distanței Hamming nu are element neutru, pentru fiecare pereche de biți corespunzătoare unui bit informațional (unei tranziții), distanța Hamming se va calcula doar între bitul extras și bitul corespunzător din perechea ce corespunde unei tranziții din trellis; pozițiile marcate cu X indică faptul că biții aflați pe acele poziții, din trellisul de la recepție, nu sunt utilizați în calculul distanțelor de tranziție și a celor cumulate (de căi) – vezi exemplu 1 pe tablă.
- aceasta face ca distanțele Hamming cumulate ale căilor cu plauzibilități mai mici să crească mai puțin la fiecare pas al decodării și de aceea probabilitatea de eroare a decodării crește, față de cea asigurată de codurile părinte. Acest inconvenient se adaugă scăderii performanțelor cauzată de extragerea biților de cod prin decizie hard.

Decodarea codurilor punctured în d_E

- extragerea informațiilor referitoare la biții codați din favorii recepționați, informații exprimate în funcție de distanța euclidiană, este complexă și nu va fi tratată în cursul de față.

Decodarea codurilor punctured în probabilități

- în acest caz din fazorii recepționați se obțin probabilitățile ca bitul b_i , $i = 1, \dots, m+1$, al fazorului recepționat f_r să aibă valorile „0” sau „1”; aceste probabilități reprezintă informațiile privitoare la biții de cod și sau biții informațional necodați care au fost mapați pe fazorul respectiv.

- Probabilitățile *a posteriori* se obțin prin operația de demapare-soft descrisă de ecuația:

$$p(b_{ij}|1) = \frac{\sum_{l=1}^{2^n} \exp\left(-\frac{(f_r - f_l)^2}{2\sigma^2}\right) \cdot b_{lj}}{\sum_{l=1}^{2^n} \exp\left(-\frac{(f_r - f_l)^2}{2\sigma^2}\right)}; \quad p(b_{ij}|0) = \frac{\sum_{l=1}^{2^n} \exp\left(-\frac{(f_r - f_l)^2}{2\sigma^2}\right) \cdot \bar{b}_{lj}}{\sum_{l=1}^{2^n} \exp\left(-\frac{(f_r - f_l)^2}{2\sigma^2}\right)}; \quad j = 1, \dots, n; \quad (47)$$

unde f_l reprezintă cei $N = 2^n$ fazori ai constelației, f_r fazorul recepționat, b_{lj} valoarea logică a bitului cu index j din fazorul l , \bar{b}_{lj} bitul cu index j din fazorul recepționat f_r .

- calculul probabilității (distanței) de tranziție se face considerând că aceasta este egală cu produsul probabilităților *a posteriori* corespunzătoare valorilor logice ale biților de cod, c_{0C1} , care corespund tranziției respective în trellis, conform tabelului T_F , tabel care pentru această variantă va conține valorile acestor biți de cod în locul coordonatelor I_j, Q_j .

- probabilitățile (distanțele) cumulate se calculează făcând produsul probabilităților corespunzând tranzițiilor care compun căile respective (ce reprezintă distanțele de tranziție), adică înmulțind succesiv probabilitatea cumulată a stării în care se termină tranziția la tactul anterior n , P_{Ci}^n , cu probabilitățile *a posteriori* ca biții de pe fazorul recepționat să ia valorile logice care corespund tranziției respective. **v. ex. 2 pe tablă**

- supraviețuitorii de nod și cel de pas se aleg a fi căile și respectiv starea care are *probabilitatea cumulată cea mai mare, care corespunde distanței cumulate minime, vezi relația (14)*. Aceasta deoarece această cale este cea mai plauzibilă de a fi generat calea recepționată.

- în cazul modulațiilor ce folosesc coduri punctured operația de poziționare a biților de cod extrași este identică cu cea de la decodarea în d_H , doar că în locul biților se pun probabilitățile *a posteriori* extrase prin demaparea soft. **v. exemplul 3 de pe tablă**

- în pozițiile corespunzătoare biților de cod ce nu au fost transmiși se pune valoarea 1, care este elementul neutru al operației de înmulțire prin care se calculează valoarea metricii în acest caz. O altă variantă este să se pună ambele probabilități egale cu 0,5, estimându-se că cele două valori logice sunt echiprobabile

- *din aceasta cauză probabilitățile cumulate ale căilor mai puțin plauzibile scad mai puțin la fiecare pas al decodării, față de cea a căii corecte, iar probabilitatea de eroare a decodării crește, față de cea asigurată de codul părinte.*

Decizia soft a biților necodați folosind probabilitățile *a posteriori*

- în cadrul operației de soft-demapping se determină și probabilitățile *a posteriori* ale biților necodați, în cazul recepției fazorului f_r . Aceste probabilități pot fi obținute prin mai multe metode, una dintre ele fiind prezentată mai jos

- Reprezentantul fiecărui subset este fazorul care are probabilitatea *a posteriori* cumulată maximă (obținută folosind probabilitățile tuturor biților mapați pe el) echivalentă cu d_E minimă față de f_r , iar biții săi necodați sunt memorați în registrul R_{bnj} , vezi decodarea biților necodați la algoritmul Viterbi în d_E .

- apoi, decizia asupra biților necodați se ia în mod similar cu cazul decodării având ca metrică d_E .

Performanțele codurilor convoluționale decodate cu algoritmul lui Viterbi având ca metrică probabilitățile *a posteriori*

- se vor prezenta efectele variației constrângerii codului pentru un cod complet cu rată constantă asupra BER și asupra C_G , în cazul în care se folosește VA precedat de demaparea soft, adică se lucrează cu probabilitățile *a posteriori* cumulate (înmulțite) ale biților.

- Variațiile BER ale modulației 2-PSK codată cu coduri convoluționale cu $R_c = 1/2$, a căror constrângere este modificată de la $K = 3$ până la $K = 9$, sunt prezentate în figura 25.

- câștigurile codării asigurate de codurile de mai sus se determină pentru $BER = 1 \cdot 10^{-5}$, știind că $BER(9,5 \text{ dB})_{2\text{-PSK}} = 1 \cdot 10^{-5}$, și au valorile aproximative listate în tabelul 4.

K	3	4	5	6	7	8	9	Tabelul 4 C_G ale codurilor cu $R_c = 1/2$ și $K = 3, 4, 5, 6, 7, 8, 9$ – obținute prin simulări pe calculator
C_G [dB]	3	3,5	4,5	5	6	6,5	7	

- codurile cu K mic asigură câștiguri reduse, dar sunt ușor de implementat, iar cele cu K mare asigură C_G mai mari dar sunt mai dificil de implementat datorită numărului mai mare de stări în trellis.

- codurile convoluționale asigură câștiguri mult mai scăzute decât turbocodurile sau codurile LDPC, care pot să asigure la această rată câștiguri de până la 10-12 dB, în anumite condiții.

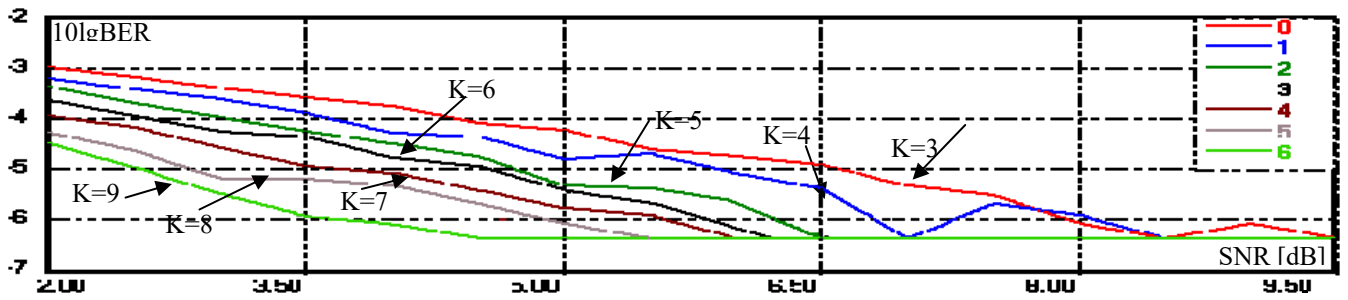


Figura 25 BER vs. SNR ale codurilor convoluționale cu $R_c = 1/2$ având $K = 3, 4, 5, 6, 7, 8, 9$

- variațiile BER vs. SNR ale modulației 64-QAM codată cu coduri convoluționale cu $K = 9$, având ratele $R_c = 1/2, 3/4, 4/5$ și $6/7$, obținute prin puncturarea codului părinte cu $R_c = 1/2, K=9$, sunt prezentate în fig. 26, alături de cea a modulației 64-QAM necodate.

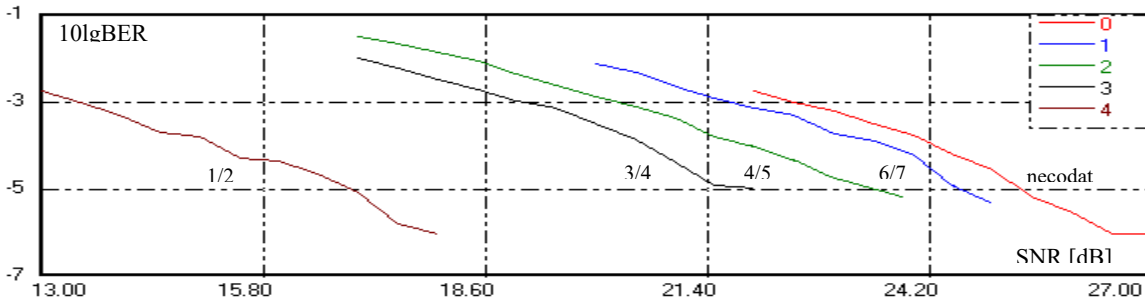


Figura 26 BER vs. SNR a modulației 64-QAM necodată și codată cu coduri convoluționale puncturate cu $R_c = 3/4, 4/5, 6/7$ obținute din codul părinte $R_c = 1/2, K = 9$

- câștigurile aproximative ale codării, față de transmisia necodată ($R_c = 1$) sunt rezumate în tabelul 5

R_c	1	6/7	5/6	4/5	3/4	2/3	1/2
C_G [dB]	-	0,5	1	3	4	6	8

Tabelul 5 C_G ale codurilor cu $R_c = 1/2, 2/3, 3/4, 4/5, 5/6, 6/7, K = 9$ – obținute prin simulări pe calculator. Notați că $p_c(25.5 \text{ dB})|_{64\text{-QAM}} \approx 1 \cdot 10^{-5}$

- graficele din figurile 25 și 26, determinate pe 10^6 biți, au o marjă de eroare de ± 0.5 dB datorită pasului de variație de 0.5 dB al SNR folosit în simulări și datorită numărului insuficient de biți ai estimării.
- codurile cu rată mare aduc câștiguri extrem de scăzute care, de cele mai multe ori, nu justifică creșterea complexității de implementare
- trebuie reținut că în cazul modulației TCM, din câștigul codării trebuie scăzută creșterea SNR datorată dublării constelației, circa 3 dB în cazul constelațiilor QAM. De aceea, configurațiile TCM cu coduri de rată mare nu merită utilizate – sunt preferabile configurații cu biți necodați și coduri cu rată mică
- pentru asigurarea unor câștiguri mai mari decât cei 3 dB necesari pentru compensarea dublării constelației, se pot utiliza modulații TCM cu rată mare care sunt generate cu coduri de rată scăzută, care aduc un câștig mare și asigură BER redus pentru biții informaționali codati, și mapând biți necodați, cărora printr-o mapare corespunzătoare li se poate asigura o probabilitate de eroare comparabilă cu a biților codati. Această mapare poate fi maparea de tip MSP sau o mapare de tip Gray bidimensională, vezi paragraful „Maparea dublu Gray”, alegerea făcându-se și în funcție de codul utilizat.
- dacă la n biți/simbol codăm $m < n$ biți informaționali cu un cod cu rată R_c , rezultă n_c biți de cod, iar rata configurației codate este:

$$R_{c_{fg}} = [n_c \cdot R_c + (n - n_c)]/n \quad (48)$$

- pentru codurile cu $R_c = m/(m+1)$, $n_c = m+1$.
- prin alegerea corespunzătoare a numerelor de biți informaționali codati și necodați și a ratei de codare (scăzute) a codului utilizat, se pot obține configurații cu rate $R_{c_{fg}}$ ridicate care asigură performanțe cel puțin egale cu cele ale configurațiilor ce utilizează numai biți codati cu coduri cu rată mare, dacă se asigură o mapare optimă celor două tipuri de biți, fie ea de tip MSP sau de tip Gray.
- metoda prezintă avantajul că elimină utilizarea mai multor coduri, fie ele și „punctured”, simplificând codarea/decodarea cu prețul mapării și deciziei biților necodați. **se va exemplifica la laborator și la master.**

Calculul debitului binar efectiv asigurat de modulația TCM

- debitul binar nominal D_n asigurat de o modulație (configurație) TCM care are rata definită de (48) este egal cu debitul transmisiei necodate care asigură același debit util, chiar dacă debitul binar efectiv transmis este mai mare de $1/R_{c_{fg}}$ ori. D_n este exprimat de:

$$D_n = f_s \cdot n \cdot R_{c_{fg}} \quad (49)$$

- pentru modulațiile codate care mapează numai biți codați, în relația (48), se va pune $n = m+1=n_c$.
- debitul binar efectiv (throughput) Θ se va obține prin înmulțirea debitului nominal cu probabilitatea ca biții să fie decodați (și deciși) corect, care se poate exprima ca $(1-p_b)$, unde p_b este probabilitatea de eroare de bit a transmisiei TCM. Dacă se lucrează la nivelul MAC se va utiliza probabilitatea de eroare de bloc codat BLER, pentru că un bloc eronat poate fi detectat folosind CRC

$$\Theta(\text{SNR}) = D_n(1-p_b(\text{SNR})) \text{ sau } \Theta(\text{SNR}) = D_n(1-\text{BLER}(\text{SNR})) \quad (50)$$

- probabilitatea de eroare de bit se poate calcula determinând separat probabilitățile de eroare a biților informaționali codați, vezi relația (15) pentru decodorul Viterbi, și a biților necodați, vezi relația $p_{ebn} = p_{cbe} p_{ebn} + p_{ebc} p_{ebn2} - \text{pag (16)}$ și explicațiile date în curs pe marginea ei.

- apoi probabilitatea globală de eroare de bit se calculează făcând media ponderată a celor două probabilități

$$p_b = \frac{n_c \cdot p_{ebc} + (n - n_c) p_{ebn}}{n} \quad (51)$$

- o altă metodă de calcul a probabilității de eroare de bit. mai simplă dar mai puțin precisă, constă în calculul probabilității de eroare de simbol a constelației necodate care asigură același debit binar util, folosind relațiile determinate în cursul de TM, în care se introduce un raport semnal/zgomot mărit cu câștigul codării C_G asigurat de configurația TCM utilizată:

$$\text{SNR}_{\text{echiv}} = \text{SNR} + C_G \leftrightarrow \rho_{\text{echiv}} = \rho \cdot 10^{C_G/10} \quad (52)$$

- apoi această probabilitate de eroare de simbol se împarte la numărul de biți/simbol al transmisiei necodate.
- eficiența spectrală (efectivă) a transmisiei TCM se obține împărțind debitul nominal (throughputul) la lățimea de bandă a transmisiei (care este aceeași atât pentru transmisia codată cât și pentru cea necodată):

$$\beta_n = D_n/f_s(1+\alpha) \text{ bps/Hz; sau } \beta_{\text{ef}} = \Theta / f_s(1+\alpha) \text{ bps/Hz; } \quad (53)$$

Schema bloc a receptorului TCM

- figura 27 prezintă schema bloc a receptorului TCM cu demodulare cu transformată Hilbert, recuperare a purtătorului asistată de decizie (DDCR), decodare soft în d_E a biților codați și decizie soft a biților necodați. În schemă nu sunt figurate blocurile de intrare (egalizare, FTB) și descramblerul la ieșire.

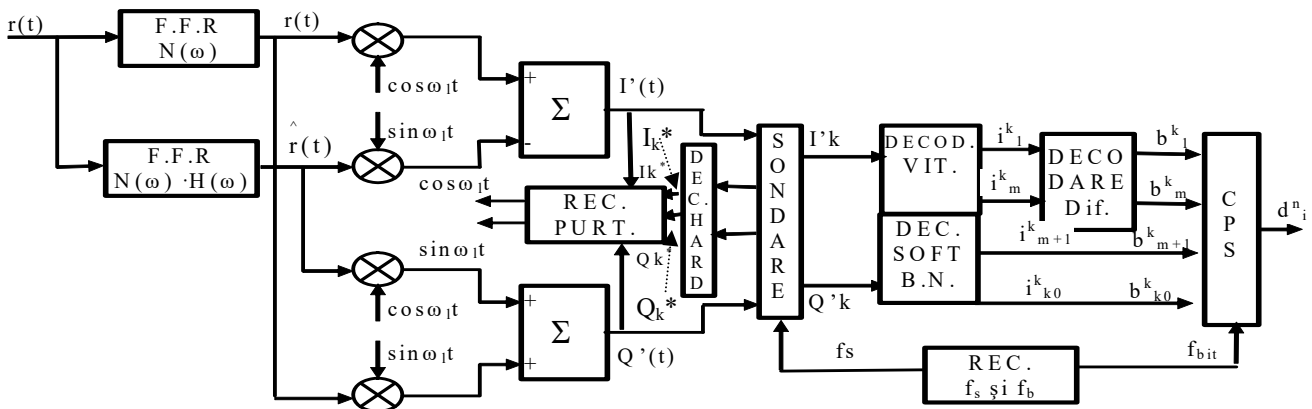


Figura 27 Schema bloc a receptorului TCM cu demodulator cu transformata H, recuperare DDCR a purtătorului, decodare VA în d_E și decizie soft a biților necodați

- receptoarele semnalelor TCM trebuie să efectueze următoarele operații:
 - egalizarea canalului și filtrarea TB și eventual AGC
 - demodularea semnalelor QAM (incluzând aici recuperarea purtătorului local și a tactului de simbol) pentru a obține semnalele I'_k și Q'_k – aici există variantele demodulatoarelor cu FTJ sau cu transformată Hilbert, iar la recuperarea purtătorului, recuperare asistată de decizie DDCR (necesită decizia hard a fazorilor demodulați) sau prin ridicare la puterea a patra.
 - decodarea biților codați – aceasta se se poate face utilizând VA cu decizie soft în d_E sau cu decizie soft în probabilități, caz în care trebuie introdus blocul de demapare soft, sau cu decizie hard în d_H , caz în care trebuie introduse blocurile de decizie hard și demapare a fazorilor
 - decodarea diferențială a biților informaționali obținuți din decodare VA, pentru a asigura invarianța acestora la rotații de $k \cdot 90^\circ$ - aceasta dacă și structura trellisului codului convoluțional utilizat o permite
 - decizia biților necodați (dacă aceștia există) – se poate face prin decizie soft, ținând cont de „informațiile” oferite de VA, sau prin decizie hard, caz în care, după decizia hard și demaparea fazorilor, biții obținuți trebuie întârziați în perioade de simbol.

- conversia paralel-serie
- descramblarea
- în funcție de variantele adoptate pentru realizarea blocurilor componente, există mai multe variante de realizare a receptorului.
- **TEMĂ:**
- *Construiți schema bloc a receptorului TCM cu demodulare FTJ, recuperare DDCR, decodare VA în d_E a biților codați și decizie hard a biților necodați*
- *Construiți schema bloc a receptorului TCM cu demodulare Hilbert, recuperare prin ridicare la puterea a patra, decodare VA în probabilități a biților codați și decizie soft a biților necodați*
- *Faceți o scurtă comparație între aceste variante și cea din figura 27 în ceea ce privește performanțele asigurate și complexitatea implementării*

Studiu de caz: modemul ITU-T V.32 bis

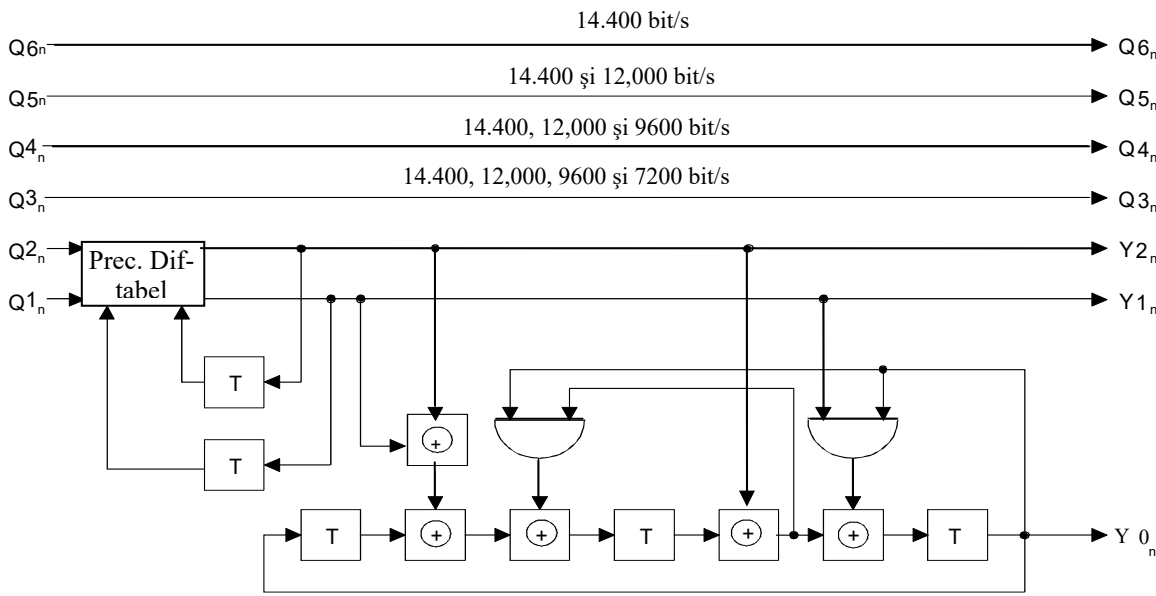
- este destinat transmisiilor adaptive multidebit pe canalul telefonic vocal analogic

Parametrii schemei de transmisie utilizate:

- $f_c = 1800$ Hz, $f_s = 2400$ Hz, $\alpha = 0,125 \rightarrow LB = f_s(1+\alpha) = 2700$ Hz; $BF = (450$ Hz, 3150 Hz)
- folosește TCM cu rata $k_0/(k_0+1)$ compusă din cod convoluțional cu $R_c = 2/3$ și $K = 4$, și constelațiile 16, 32, 64, 128-QAM
- numărul $k_0 - m$ de biți necodați, constelațiile utilizate, debitele utile și cele transmise ale configurațiilor utilizate sunt prezentate în tabelul 6 și ilustrate în figura 28.

$k_0 - m$	R_{TCM}	Const.	D_{util} [bps]	D_{trs} [bps]
0 – nu e în standard	$2/3$	8-QAM	$2 \cdot 2400 = 4800$	$3 \cdot 2400 = 7200$
1	$3/4$	16-QAM	$3 \cdot 2400 = 7200$	$4 \cdot 2400 = 9600$
2	$4/5$	32-QAM	$4 \cdot 2400 = 9600$	$5 \cdot 2400 = 12000$
3	$5/6$	64-QAM	$5 \cdot 2400 = 12000$	$6 \cdot 2400 = 14400$
4	$6/7$	128-QAM	$6 \cdot 2400 = 14400$	$7 \cdot 2400 = 16800$

Tabelul 6 Parametrii configurațiilor TCM utilizate în modemul ITU-T V.32bis



TCM folosite în modemul ITU-T V.32bis

Figura 28 Structura configurațiilor

- **Chestiuni de studiat:**

- *de ce se utilizează $R_c = 2/3$?*
- *care e diagrama trellis ?*
- *care e partiționarea MSP?*
- *cât e d_{Efree} și d_p ?*
- *cât e câștigul codării ?*
- *care sunt domeniile de valori ale SNR pentru care se utilizează fiecare configurație? Vezi laborator*

Notă: acesta este primul caz de utilizare a modulației și codării adaptive (AMC), folosind un set de configurații stocate în fiecare echipament, în aplicații comerciale cu largă utilizare.