

Laborator de
Bazele Procesării
și
Transmiterii Semnalelor
II

Honorius Gâlmeanu

Brașov 2002

Laborator 1

Introducere în MATLAB

MATLAB este un pachet de programe de înaltă performanță dedicat calculului numeric și reprezentărilor grafice. El integrează analiză numerică, calcul matriceal, procesarea semnalului și reprezentările grafice într-un mediu ușor de învățat și folosit. Elementul de bază cu care operează MATLAB este matricea, calculele cu matrici fiind optimizate. Rezultatul direct este că un același calcul executat cu MATLAB va fi de 10 ori mai rapid implementat prin operații cu matrici decât iterativ, prin instrucțiuni `for`. Cu ajutorul MATLAB se pot rezolva probleme fără a fi necesară scrierea unui program într-un limbaj de nivel înalt.

1.1 Sesiunea MATLAB. Efectuarea calculelor

MATLAB este o aplicație Windows, de obicei fișierul **MATLAB.EXE**. Pornirea acesteia se face din meniul *File | Run* al *File Manager*-ului. Odată pornit, va apare un logo urmat de:

```
>>
```

unde `>>` este prompter-ul MATLAB. Lucrul se face în directorul curent; acesta este afișat de comanda:

```
>> cd  
F:\APP.WIN\MATLAB\BIN
```

Orice salvare/încărcare de variabile sau programe se face din directorul de lucru curent. Schimbarea acestuia se realizează cu aceeași comandă:

```
>> cd G:\
```

Operatorii aritmetici de bază sunt `+` `-` `*` `/` `^` iar aceștia se pot utiliza combinat cu `()`. De exemplu o comandă de calcul direct produce următorul rezultat:

```
>> 2 + 3/4*5  
ans =  
5.7500
```

Rezultatul calculelor va fi depus în variabila `ans`. Putem folosi propriile nume pentru stocarea rezultatelor:

```
>> x = 3-2^4
x =
    -13
>> ans*5
ans =
   -65
```

De obicei nu dorim să vedem rezultatul calculelor intermediare - vom încheia instrucțiunea cu caracterul `;`:

```
x = -13; y = 5*x, z = x^2+y
y =
   -65
z =
   104
```

În afară de operatori, MATLAB cunoaște funcțiile elementare `sin`, `cos`, `tan`, `asin`, `acos`, `atan`, `sqrt`, `exp`, `log`, `log10`, `log2`, `rem`, `sign`:

```
>> x = 9;
>> sqrt(x), exp(x), log(sqrt(x))
ans =
     3
ans =
 8.1031e+03
ans =
 1.0986
>> sign(x) % semnul lui x
ans =
     1
>> rem(3, 2) % restul impartirii lui 3 la 2
ans =
     1
```

1.2 Operații cu vectori și matrici

Vectorii descriși pot fi vectori linie, elemente separate de spații sau `,`, respectiv vectori coloană, unde elementele sunt separate de `;`:

```
>> v = [1 3 sqrt(5)]
v =
 1.0000  3.0000  2.2361
```

```
>> length(v)
ans =
     3
>> w = [1 ; 2 ; 3]
w =
     1
     2
     3
```

Notăția `:` produce un vector linie:

```
>> 1:4
ans =
     1     2     3     4
```

Mai general `a:b:c` produce un vector de elemente începînd cu `a`, cu incrementul `b` și terminînd cu `c`:

```
>> 0.32:0.1:0.6
ans =
    0.3200    0.4200    0.5200
```

Pentru matrice sau vector, putem afla valoarea unui anumit element sau părți din aceea matrice:

```
>> a = [1 2 3; 4 5 6]
ans =
     1     2     3
     4     5     6
>> a(2, 1)
ans =
     4
>> a(1:2, 2:3)
ans =
     2     3
     5     6
>> v = [4 1 5];
>> [a(1, 1:3) ; v]
ans =
     1     2     3
     4     1     5
```

unde am schimbat ultima linie a lui `a` cu `v`. Transpusa unei matrici `a` este `a'`. Inversa ei este `inv(a)`. Produsul a două matrici poate fi realizat cu `*` sau element cu element, cu operatorul `.*`, caz în care matricile trebuie să aibă aceeași dimensiune:

```
>> b = [3 2 1; 2 2 1];
>> a'
ans =
     1     4
     2     5
     3     6
>> a * a'
ans =
    14    32
    32    77
>> a .* b
ans =
     3     4     3
     8    10     6
>> inv([a; 3 2 3])
ans =
   -0.5000    0.0000    0.5000
   -1.0000    1.0000   -1.0000
    1.1667   -0.6667    0.5000
>> v = [2 4 6]
ans =
     2     4     6
>> sum(v)
ans =
    12
>> length(v)
ans =
     3
```

Funcția `diag(v)` întoarce o matrice pătrată ce are pe diagonala principală elementele vectorului `v`. `ones(m,n)` are ca rezultat o matrice de `m` linii și `n` coloane cu toate elementele 1. `zeros(m,n)` are același efect însă inițializează cu 0. Matricea identitate se obține cu funcția `eye(n)`:

```
>> diag(v)
ans =
     2     0     0
     0     4     0
     0     0     6
>> ones(2, 3)
ans =
     1     1     1
     1     1     1
>> zeros(2) % acelasi efect ca zeros(2, 2)
```

```
ans =
    0    0
    0    0
>> eye(3)
ans =
    1    0    0
    0    1    0
    0    0    1
```

1.3 Salvarea și restaurarea variabilelor

Variabilele din zona de lucru pot fi salvate în două formate - text și MATLAB, în cel din urmă caz fișierele au extensia .MAT. Iată un exemplu de matrice editată în Notepad:

```
A.TXT:
 1  3  2
 4  1  5
```

Și iată un exemplu de restaurare/salvare în format ASCII:

```
>> % este creata matricea A citita din fisierul cu extensia .TXT
>> load A.TXT
>> B = A;
>> % se salveaza in fisierul B.TXT variabila B, in format ASCII
>> save B.TXT B -ascii
>> % se salveaza in fisierul B.MAT variabila B, in format MATLAB
>> save B
>> clear B % elibereaza zona ocupata de variabila B
>> pack % optimizeaza zona de memorie folosita
>> clear % elibereaza toate variabilele din memorie
>> who
Your variables are:

>> % incarcare din fisierul B.MAT, in variabila B
>> load B
>> who
Your variables are:
B
>> whos % mai multe detalii despre variabilele curente
```

1.4 Structuri de control

MATLAB pune la dispozitie următoarele structuri de control:

```

if <expresie>          for i = start:stop      while <expresie>
    ...                ...                ...
elseif <expresie>     end                end
    ...
else
    ...
end

```

Expresia folosită are valoare logică. Operatorii logici folosiți de MATLAB sunt == pentru comparație, ~= diferit, & pentru AND și | pentru OR.

1.5 Fișiere de comenzi (script)

Comenzile pot fi prelucrate atât individual, la prompter-ul >>, cât și în fișiere script. Acestea pot fi de două feluri:

- fișiere script propriu-zise, pentru care comenzile se execută iar variabilele folosite sunt păstrate și după terminarea execuției fișierului;
- fișiere funcție, care se execută la fel ca fișierele script cu excepția faptului că pot fi apelate cu parametri, întorc o valoare ca rezultat iar variabilele folosite sunt dealocate, cu excepția celei ce desemnează rezultatul întors de funcție.

Un fișier script va conține comenzi MATLAB și va avea extensia .m. Odată salvat în directorul curent, lansarea lui se face simplu, de la prompter-ul MATLAB, prin invocarea numelui (fără extensie).

Un fișier funcție are o structură specială. Prima linie de cod va trebui să conțină cuvântul cheie `function` ce specifică numele și rezultatul întors de funcție. Un exemplu simplu de funcție (conținutul fișierului `putere3.m`) este următorul:

```

% putere3(a)    Calculeaza puterea a treia a lui a

% observati linia liberă de mai sus; cu comanda >> help putere3
% se vor afisa primele linii de comentariu din putere3.m pana la
% prima linie liberă
% urmeaza prima linie de cod. x este variabila ce va fi returnata
function [x] = putere3(a)
x = a * a * a;
return

```

Apelînd funcția, se vor produce următoarele rezultate:

```

>> putere3(4)
ans =
    64

```

```
>> help putere3
putere3(a)    Calculeaza puterea a treia a lui a
>> disp('text')
text
```

Într-un fișier script sau funcție, funcția `error('mesaj')` afișează mesajul și întoarce la prompter-ul MATLAB.

1.6 Funcții de existență

- `exist(x)` întoarce `true` dacă variabila `x` există
- `any(x)` întoarce `true` dacă există cel puțin un element al vectorului `x` este `true`
- `all(x)` întoarce `true` dacă toate elementele vectorului `x` sunt `true`
- `find(x)` întoarce indicii elementelor nenule din vectorul `x`
- `isempty(x)` întoarce `true` dacă cel puțin una din dimensiunile matricii `x` este 0.

1.7 Funcții pentru șiruri de caractere

- `abs('sir')` întoarce un vector conținând codurile ASCII ale caracterelor șirului
- `setstr(x)` întoarce șirul de caractere ale cărui coduri ASCII sunt date de vectorul `x`
- `isstr(x)` întoarce `true` dacă argumentul e șir de caractere
- `eval('expresie MATLAB')` evaluează acea expresie
- `strcmp(a,b)` întoarce rezultatul comparării celor două șiruri de caractere
- `findstr(a,b)` întoarce indicii aparițiilor lui `b` în `a`
- `upper(x)` convertește șirul la majuscule
- `lower(x)` convertește șirul la minuscule
- `num2str(x)` conversie de la număr la șir de caractere
- `str2num(x)` conversie inversă de string la număr

1.8 Funcții de aproximare, random și extreme

- `ceil(x)` rotunjește la cel mai apropiat întreg spre $+\infty$
- `fix(x)` rotunjește la cel mai apropiat întreg spre 0
- `floor(x)` rotunjește la cel mai apropiat întreg spre $-\infty$
- `round(x)` rotunjește la cel mai apropiat întreg

1.9 Crearea graficelor

Graficele în MATLAB se creează cu comenzile `plot`, `mesh`, `contour`. Se stabilește în prealabil domeniul pe care se reprezintă, de regulă un vector. Iată de exemplu rezolvarea grafică a ecuației:

$$\sin(x) = \frac{1-x}{1+x} \quad (1.1)$$

Secvența următoare de comenzi produce figura 1.1:

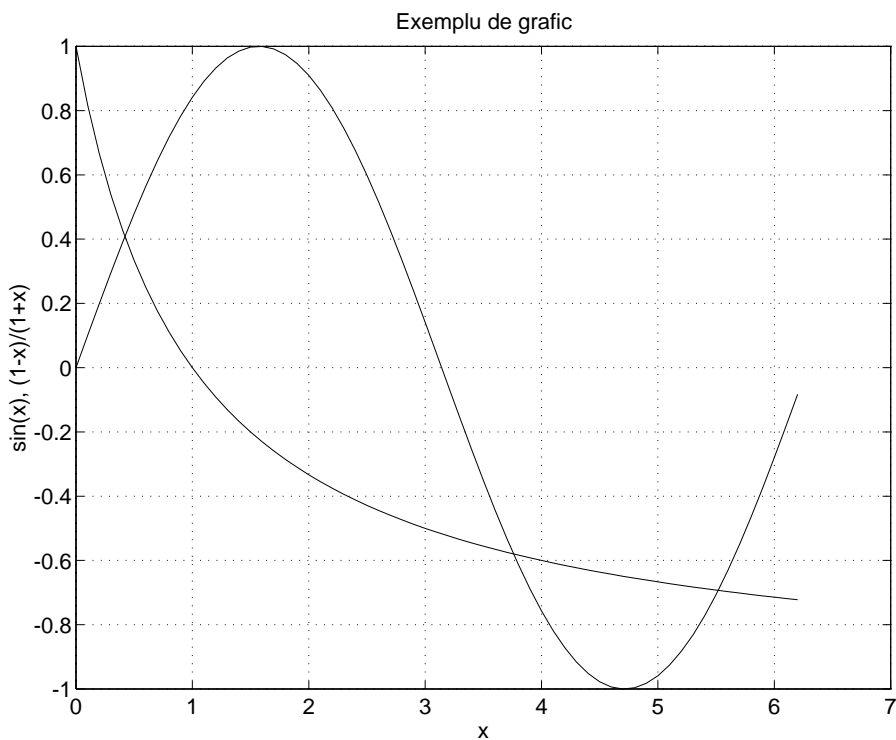


Figura 1.1: Exemplu de folosire a comenzii `plot`

```
>> x = 0:.1:2*pi;
>> plot(x,sin(x),x,(1-x)./(1+x))
>> grid
>> xlabel('x')
>> ylabel('sin(x),(1-x)/(1+x)')
>> title('Exemplu de grafic')
```

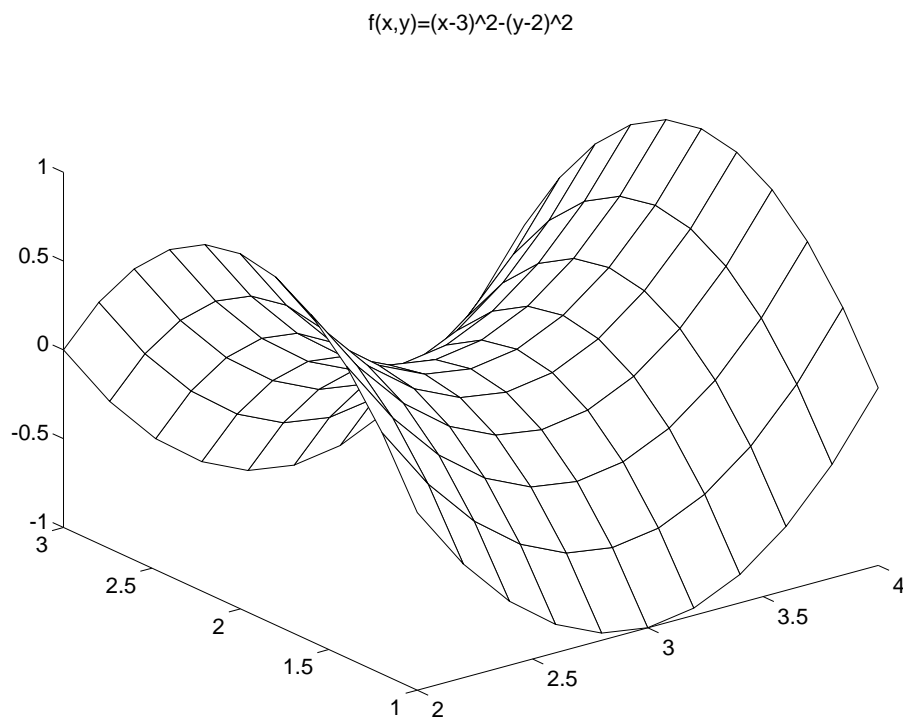


Figura 1.2: Exemplu de folosire a comenzii mesh

O suprafață este definită matematic ca înălțimea calculată pentru fiecare (x, y) , $z = f(x, y)$. De exemplu dorim să reprezentăm funcția pentru domeniile $2 \leq x \leq 4$ și $1 \leq y \leq 3$. Pentru aceasta construim valorile discrete corespunzătoare cu funcția `meshgrid`:

```
>> [X,Y] = meshgrid(2:.5:4, 1:.5:3);
>> X
X =
    2.0000    2.5000    3.0000    3.5000    4.0000
    2.0000    2.5000    3.0000    3.5000    4.0000
    2.0000    2.5000    3.0000    3.5000    4.0000
    2.0000    2.5000    3.0000    3.5000    4.0000
    2.0000    2.5000    3.0000    3.5000    4.0000
>> Y
```

```
Y =
    1.0000    1.0000    1.0000    1.0000    1.0000
    1.5000    1.5000    1.5000    1.5000    1.5000
    2.0000    2.0000    2.0000    2.0000    2.0000
    2.5000    2.5000    2.5000    2.5000    2.5000
    3.0000    3.0000    3.0000    3.0000    3.0000
```

Coordonatele punctului (i, j) sunt date de $(X(i, j), Y(i, j))$. După calculul funcției cu ajutorul lui X și Y , reprezentarea sa se face cu ajutorul funcțiilor `mesh` sau `contour`. Să presupunem că dorim să reprezentăm suprafața:

$$f(x, y) = (x - 3)^2 - (y - 2)^2 \quad (1.2)$$

pentru $2 \leq x \leq 4$ și $1 \leq y \leq 3$. Secvența de comenzi MATLAB următoare va produce figura 1.2:

```
>> [X,Y] = meshgrid(2:.2:4, 1:.2:3);
>> Z = (X-3).^2-(Y-2).^2;
>> mesh(X,Y,Z)
>> title('f(x,y)=(x-3)^2-(y-2)^2')
```

1.10 Documentație

Pentru orice comandă, o scurtă descriere este disponibilă folosind secvența:

```
>> help <comanda>
```

Demonstrații asupra capabilităților MATLAB pot fi obținute cu comanda:

```
>> demo
```

Comanda:

```
>> help
```

va produce o listă a categoriilor pentru care help-ul este disponibil.

1.11 Exerciții

Folosind cele citite până acum:

1. Realizați fișierul funcție `fib.m` care calculează șirul Fibonacci:

$$fib(n) = \begin{cases} 1 & \text{dacă } n < 2 \\ fib(n-2) + fib(n-1) & \text{dacă } n \geq 2 \end{cases} \quad (1.3)$$

2. Rezolvați, folosind calculul matricial, sistemul de ecuații:

$$\begin{cases} 2x + 3y + z = 10 \\ x + y - z = 0 \\ x + y + 5z = 18 \end{cases} \quad (1.4)$$

3. Realizați funcția care calculează recursiv C_n^k :

$$C_n^k = \begin{cases} 1, & k = 0 \\ n, & k = 1 \\ C_{n-1}^k + C_{n-1}^{k-1}, & k \leq n \end{cases} \quad (1.5)$$

4. Reprezentați grafic funcția:

$$f(x) = \frac{\sin(x)}{x} \quad (1.6)$$

5. Realizați funcția recursivă ce afișează strategia de deplasare a inelelor pentru problema turnurilor Hanoi. Considerați 8 inele așezate descrescător pe una din cele 3 vergele.

Laborator 2

Entropia unei surse binare. Surse Markov

2.1 Noțiunea și măsura informației

Fie o sursă discretă ce emite simboluri din alfabetul X , cu probabilitățile P :

$$\begin{bmatrix} X \\ P \end{bmatrix} \quad \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{bmatrix} \quad (2.1)$$

Incertitudinea de realizare a evenimentului x_i este funcție de probabilitatea sa de apariție p_i - cu cât probabilitatea de realizare a evenimentului este mai mare, cu atât incertitudinea de observare a evenimentului x_i este mai mică:

$$i(x_i) = U(x_i) = F(p_i) \quad (2.2)$$

Informația $i(x_i)$ realizată pentru un simbol x_i este egală cu incertitudinea de realizare a aceluși simbol. Deci pentru un eveniment compus $\{x_i, x_j\}$ care constă în îndeplinirea simultană a evenimentelor x_i și x_j , probabilitatea:

$$p(x_i, x_j) = p(x_i) \cdot p(x_j) \quad (2.3)$$

este mai mică decât fiecare probabilitate în parte, deci incertitudinea de realizare a evenimentului $\{x_i, x_j\}$ va fi mai mare decât fiecare incertitudine pentru realizarea evenimentelor x_i, x_j luate separat:

$$U(x_i, x_j) = U(x_i) + U(x_j) \quad (2.4)$$

Incertitudinea este funcție aditivă de probabilitate, de unde rezultă formula informației ce se obține la realizarea evenimentului x_i :

$$i(x_i) = -\log_2 p(x_i) \quad (2.5)$$

Măsura informației se exprimă în biți, iar dacă se schimbă baza logaritmului în e sau 10:

$$1[nit] = \log_2 e = 1.44[bit] \quad (2.6)$$

$$1[dit] = \log_2 10 = 3.32[bit] \quad (2.7)$$

2.2 Cazuri extreme de canale discrete

Fie cazul unui canal cu perturbații pentru care sursa emite simbolul x_i iar receptorul primește simbolul y_j . Informația realizată asupra simbolului x_i când se recepționează y_j este diferența dintre informația apriori (înainte de observare) și informația aposteriori (constatată după realizarea evenimentului):

$$i(x_i, y_j) = i(x_i) - i(x_i/y_j) = -\log_2 p(x_i) + \log_2 p(x_i/y_j) = \log_2 \frac{p(x_i/y_j)}{p(x_i)} \quad (2.8)$$

1. Cazul canalului fără perturbații

Probabilitatea de realizare a evenimentului x_i când se observă y_j este chiar probabilitatea de realizare a lui x_i (y_j observat este chiar x_i):

$$p(x_i/y_j) = p(x_i/x_i) = 1 \quad (2.9)$$

$$i(x_i, y_j) = \log_2 \frac{1}{p(x_i)} = i(x_i), \quad (2.10)$$

informația dobândită este chiar informația obținută asupra evenimentului x_i .

2. Cazul canalului puternic perturbat

Evenimentele x_i și y_j sunt independente:

$$p(x_i, y_j) = p(x_i) \cdot p(y_j) = p(x_i/y_j) \cdot p(y_j) \quad \text{deci} \quad p(x_i/y_j) = p(x_i) \quad (2.11)$$

$$i(x_i, y_j) = \log_2 \frac{p(x_i)}{p(x_i)} = 0, \quad (2.12)$$

informația dobândită asupra evenimentului x_i este nulă.

2.3 Entropia. Proprietăți

Pentru o sursă discretă, entropia sursei este informația medie statistică pe simbol:

$$H(X) = -\sum_{i=1}^n p_i \log_2 p_i \quad (2.13)$$

Proprietățile entropiei:

1. funcție continuă;
2. funcție simetrică (p_i se pot interschimba);
3. mărime aditivă;

4. valoarea maximă a entropiei (cu n numărul de stări ale sursei):

$$H_{max}(X) = \log_2 n \quad (2.14)$$

5. debitul de informație:

$$D(X) = \frac{H(X)}{\bar{\tau}} \quad (2.15)$$

unde $\bar{\tau}$ este durata medie a unui simbol, astfel:

$$\bar{\tau} = \sum_{i=1}^n p(x_i) \cdot \tau_i \quad (2.16)$$

6. redundanța sursei:

$$R(X) = H_{max}(X) - H(X) \quad (2.17)$$

7. redundanța relativă:

$$\rho(X) = 1 - \frac{H(X)}{H_{max}(X)} \quad (2.18)$$

8. eficiența sursei:

$$\eta(X) = \frac{H(X)}{H_{max}(X)} \quad (2.19)$$

2.4 Surse Markov

O **sursă discretă cu memorie** este o sursă discretă pentru care probabilitatea de emisie a simbolului la momentul curent depinde de simbolurile emise anterior:

$$p(x^{(k)}) = p(x^{(k)} | x^{(k-1)}, x^{(k-2)} \dots x^{(k-r)}) \quad (2.20)$$

O **sursă markoviană** este o sursă discretă cu memorie de ordinul I, adică probabilitatea simbolului emis la momentul curent depinde numai de simbolul emis anterior:

$$p(x^{(k)}) = p(x^{(k)} | x^{(k-1)}) \quad (2.21)$$

Notăm cu p_{ij} $p(x_i^{(k)} | x_j^{(k-1)})$, adică probabilitatea ca sursa să treacă din starea i în starea j (în fiecare stare sursa emite un anumit simbol). Dintr-o stare i sigur sursa va trece într-o altă stare, deci putem scrie:

$$\sum_{j=1}^n p_{ij} = 1 \quad (2.22)$$

Probabilitatea ca sursa să se afle în starea i la momentul k este:

$$p_{(i)}^{(k)} = p_{(1)}^{(k-1)} \cdot p_{1j} + \dots + p_{(n)}^{(k-1)} \cdot p_{nj} \quad \text{pentru} \quad i = 1 \dots n \quad (2.23)$$

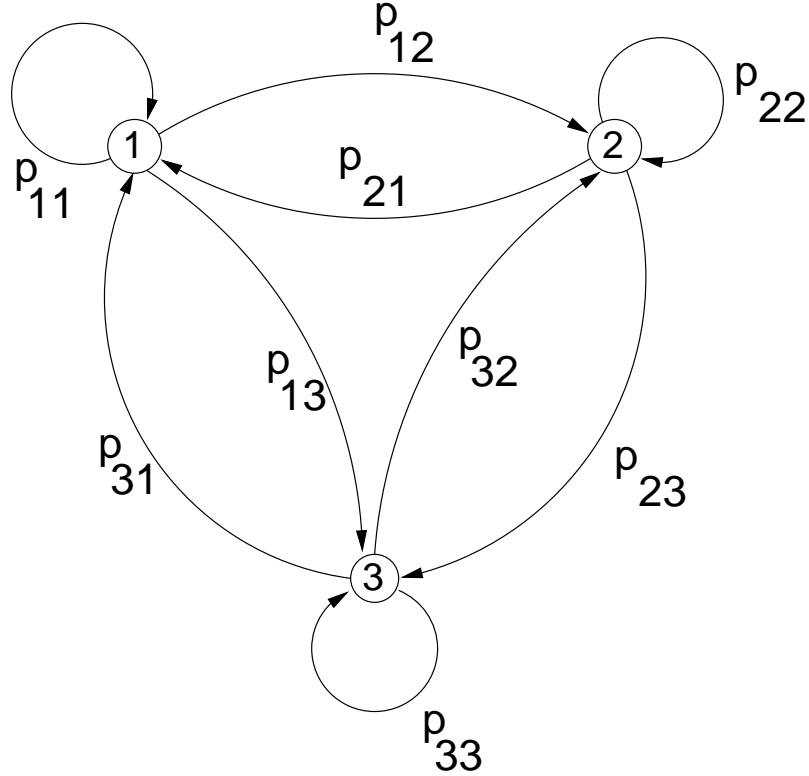


Figura 2.1: Graful de tranziție pentru o sursă Markov cu 3 stări

Corespunzător, vectorul probabilităților ca sursa să se afle într-una din cele n stări la momentul k este:

$$p^{(k)} = T^t \cdot p^{(k-1)}, \quad (2.24)$$

unde T este matricea de tranziție a stărilor, cunoscută apriori.

Să presupunem cazul particular $n = 3$ (figura 2.1). Pentru acesta putem scrie sistemul:

$$\begin{aligned} p_{(1)}^{(k)} &= p_{(1)}^{(k-1)} \cdot p_{11} + p_{(2)}^{(k-1)} \cdot p_{21} + p_{(3)}^{(k-1)} \cdot p_{31} \\ p_{(2)}^{(k)} &= p_{(1)}^{(k-1)} \cdot p_{12} + p_{(2)}^{(k-1)} \cdot p_{22} + p_{(3)}^{(k-1)} \cdot p_{32}, \\ p_{(3)}^{(k)} &= p_{(1)}^{(k-1)} \cdot p_{13} + p_{(2)}^{(k-1)} \cdot p_{23} + p_{(3)}^{(k-1)} \cdot p_{33} \end{aligned} \quad (2.25)$$

care se poate scrie matricial astfel:

$$\underbrace{\begin{bmatrix} p_{(1)}^{(k)} \\ p_{(2)}^{(k)} \\ p_{(3)}^{(k)} \end{bmatrix}}_{p^{(k)}} = \underbrace{\begin{bmatrix} p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \\ p_{13} & p_{23} & p_{33} \end{bmatrix}}_{T^t} \cdot \underbrace{\begin{bmatrix} p_{(1)}^{(k-1)} \\ p_{(2)}^{(k-1)} \\ p_{(3)}^{(k-1)} \end{bmatrix}}_{p^{(k-1)}}, \quad (2.26)$$

unde

$$\sum_{i=1}^3 p_{(i)}^{(k)} = 1, \quad (2.27)$$

iar T este matricea de tranziție a stărilor sursei. Conform celor de mai sus, suma elementelor de pe fiecare linie a lui T trebuie să fie 1.

Se numește **sursă staționară** o sursă discretă pentru care probabilitatea de emisie a simbolului în starea curentă nu depinde de originea timpului (este aceeași în timp). Condiția de staționaritate se scrie astfel:

$$p^{(k)} = T^t \cdot p^{(k-1)} = p^{(k-1)}, \quad (2.28)$$

deci probabilitatea $p^{(k)}$ la orice moment k să fie vector propriu al lui T^t , cu $\lambda = 1$.

2.5 Problemă rezolvată

Pentru o sursă Markov cu 2 stări, staționară, determinați, în funcție de matricea de tranziție a stărilor, vectorul probabilităților staționare.

Rezolvare

Deoarece este o sursă staționară, putem scrie, pentru vectorul probabilităților simbolurilor emise de sursă p și matricea de tranziție a sursei T :

$$p = T^t \cdot p \quad (2.29)$$

unde

$$p = \begin{bmatrix} x \\ 1 - x \end{bmatrix} \quad (2.30)$$

respectiv

$$T = \begin{bmatrix} a & 1 - a \\ b & 1 - b \end{bmatrix} \quad (2.31)$$

Atunci putem scrie:

$$\begin{bmatrix} x \\ 1 - x \end{bmatrix} = \begin{bmatrix} a & b \\ 1 - a & 1 - b \end{bmatrix} \cdot \begin{bmatrix} x \\ 1 - x \end{bmatrix} \quad (2.32)$$

Adică:

$$\begin{bmatrix} x \\ 1 - x \end{bmatrix} = \begin{bmatrix} ax + b(1 - x) \\ (1 - a)x + (1 - b)(1 - x) \end{bmatrix} \quad (2.33)$$

Din prima relație avem:

$$x = ax + b - bx \quad (2.34)$$

de unde:

$$p = \begin{bmatrix} \frac{b}{1-a+b} \\ \frac{1-a}{1-a+b} \end{bmatrix} \quad (2.35)$$

Pentru matricea de tranziție:

$$T = \begin{bmatrix} 0.2 & 0.8 \\ 1 & 0 \end{bmatrix} \quad (2.36)$$

Obținem vectorul probabilităților staționare:

$$p = \begin{bmatrix} 0.55556 \\ 0.44444 \end{bmatrix} \quad (2.37)$$

2.6 Exerciții

1. Să se calculeze cantitatea de informație necesară pentru precizarea poziției unei figuri pe tabla de șah.
2. Fie 12 monede dintre care una falsă (mai ușoară sau mai grea). Determinând cantitățile de informație, să se calculeze numărul minim de cântăriri necesare găsirii monedei false și precizării dacă este mai ușoară sau mai grea. Folosim pentru cântăriri o balanță fără greutate.
3. Fie un alfabet format din 3 litere A, B, C. Să se calculeze:
 - (a) numărul maxim de mesaje de lungime 3 ce se pot forma;
 - (b) cantitatea de informație conținută de un asemenea mesaj.
4. Fie o sursă Markov cu 3 stări, staționară pentru care se dă matricea de tranziție a stărilor. Realizați programul `MATLAB` care să calculeze probabilitățile de emisie ale simbolurilor la un moment dat. Atenție la relațiile dintre elementele matricii de tranziție !
5. Pentru o sursă Markov cu 3 stări, caracterizată de matricea de tranziție:

$$\begin{bmatrix} 0.01 & 0.98 & 0.01 \\ 0.01 & 0.01 & 0.98 \\ 0.98 & 0.01 & 0.01 \end{bmatrix} \quad (2.38)$$

Realizați programul `MATLAB` ce iterează probabilitățile succesive de transmitere a celor trei simboluri. Ca probabilități inițiale folosiți $p_0 = [0 \ 1 \ 0]$. După câți pași se ajunge la staționaritate ?

6. Realizați programul `MATLAB` pentru a vizualiza variația entropiei unei surse care emite numai două simboluri. Când este nulă valoarea entropiei ? Dar maximă ?
7. O sursă discretă care generează 8 mesaje se caracterizează prin:

$$[S] = [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6 \ s_7 \ s_8] \quad (2.39)$$

$$[\tau] = [2 \ 2 \ 3 \ 3 \ 4 \ 4 \ 4 \ 4] \quad (2.40)$$

$$[P] = [1/4 \ 1/4 \ 1/8 \ 1/8 \ 1/16 \ 1/16 \ 1/16 \ 1/16] \quad (2.41)$$

Să se calculeze în `MATLAB`:

- (a) entropia sursei și valoarea ei maximă;
- (b) debitul de informație;
- (c) redundanța sursei;
- (d) redundanța relativă și eficiența sursei.

8. Cu ajutorul următoarei secvențe MATLAB:

```
% STARE(p) - genereaza simbolul emis de sursa discreta
%           caracterizata de probabilitatile simbolurilor
%           (vectorul p al acestora)

% incercati comanda "help stare"
% va afisa primele linii ce incep cu '%'
function [i] = stare(p)
% functia "stare" are un parametru si intoarce i
if sum(p) ~= 1
    error('Vectorul p nu caracterizeaza o sursa discreta !');
end
x=rand(1,1); % genereaza un numar aleator intre 0 si 1
n=length(p); % numarul de simboluri
k=0;
for i=1:n % probabilitatea sa fie emis unul din
    r(i)=p(i)+k; % simbolii de ordin <= i
    k=k+p(i);
end
for i=1:n % intoarce acel simbol (stare)
    if x<=r(i)
        return;
    end
end
end
```

Realizati programul care simulează o sursă Markov cu 3 stări. Programul construiește pe baza vectorului probabilităților inițiale $p^{(0)}$ și a matricii de tranziție a stărilor T succesiunea de simboluri care sunt emise la momente distincte în timp.

Laborator 3

Canale discrete. Transinformația

3.1 Canale discrete. Entropii condiționate

Pentru un canal discret, se notează cu $H(X)$ entropia câmpului de intrare (este chiar entropia sursei ce emite simboluri din alfabetul X și nu depinde decât de sursă), iar cu $H(Y)$ entropia câmpului de ieșire (informația medie statistică pe simbol obținută la ieșire).

De obicei se cunosc probabilitățile condiționate ale ieșirii în condițiile când s-a observat intrarea, adică matricea de zgomot:

$$P[Y/X] = \begin{bmatrix} p(y_1/x_1) & p(y_2/x_1) & \dots & p(y_m/x_1) \\ p(y_1/x_2) & p(y_2/x_2) & \dots & p(y_m/x_2) \\ \dots & \dots & \dots & \dots \\ p(y_1/x_n) & p(y_2/x_n) & \dots & p(y_m/x_n) \end{bmatrix} \quad (3.1)$$

În condițiile în care se observă intrarea, de la un anumit x_i în mod sigur vom ajunge la unul din simbolurile y_j din spațiul de ieșire, deci există relația:

$$\sum_{j=1}^m p(y_j/x_i) = 1 \quad (3.2)$$

În mod asemănător, pentru matricea de tranziție:

$$P[X/Y] = \begin{bmatrix} p(x_1/y_1) & p(x_1/y_2) & \dots & p(x_1/y_m) \\ p(x_2/y_1) & p(x_2/y_2) & \dots & p(x_2/y_m) \\ \dots & \dots & \dots & \dots \\ p(x_n/y_1) & p(x_n/y_2) & \dots & p(x_n/y_m) \end{bmatrix} \quad (3.3)$$

avem relația:

$$\sum_{i=1}^n p(x_i/y_j) = 1 \quad (3.4)$$

deoarece dacă observăm că s-a primit simbolul y_j , în mod sigur pentru aceasta s-a emis unul din simbolurile x_i .

Dacă se notează câmpul produs cu:

$$P[X, Y] = \begin{bmatrix} x_1y_1 & x_1y_2 & \dots & x_1y_m \\ x_2y_1 & x_2y_2 & \dots & x_2y_m \\ \dots & \dots & \dots & \dots \\ x_ny_1 & x_ny_2 & \dots & x_ny_m \end{bmatrix}, \quad (3.5)$$

unde cu x_iy_j am notat evenimentul simultan emisia lui x_i și recepția lui y_j , entropia câmpului produs este:

$$H(X, Y) = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(x_i, y_j) \quad (3.6)$$

Pentru cazul când se cunoaște câmpul de la ieșire Y , se face presupunerea că s-a recepționat y_j . Dacă nu ar fi perturbații, s-ar putea spune că $x_i = y_j$. În cazul real există o incertitudine asupra simbolului emis. Pornind de la expresia incertitudinii de realizare a evenimentului x_i când se realizează y_j , $U(x_i/y_j) = -\log p(x_i/y_j)$, se poate calcula incertitudinea medie asupra sursei de intrare când se cunoaște alfabetul de ieșire:

$$H(X/Y) = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(x_i/y_j), \quad (3.7)$$

și reprezintă **echivocul** ce există asupra lui X când se cunoaște Y .

Pentru cazul când se cunoaște intrarea X , se poate calcula incertitudinea despre ieșire. Există o incertitudine asupra simbolului recepționat. Pornind de la expresia incertitudinii de realizare a evenimentului y_j când s-a emis x_i , se poate calcula incertitudinea despre ieșire când se cunoaște intrarea:

$$H(Y/X) = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(y_j/x_i), \quad (3.8)$$

și reprezintă **eroarea medie** a lui Y când se cunoaște X .

$H(X/Y)$ și $H(Y/X)$ sunt referite sub denumirea de **entropii condiționate**. Se demonstrează următoarele relații între entropii:

$$H(X, Y) = H(X/Y) + H(Y) \quad (3.9)$$

$$H(X, Y) = H(Y/X) + H(X) \quad (3.10)$$

1. Pentru cazul canalului fără perturbații, dacă se trimite x_i se va recepționa x_i deci:

$$p(x_i/y_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \quad (3.11)$$

și avem:

$$\begin{aligned} H(X/Y) &= 0 & H(X, Y) &= H(Y) \\ H(Y/X) &= 0 & H(X, Y) &= H(X) \end{aligned} \quad (3.12)$$

2. Pentru cazul canalului puternic perturbat, dacă se trimite x_i se va putea recepționa absolut oricare din simbolurile y_j deci $p(x_i/y_j) = p(x_i)$ și avem:

$$p(x_i, y_j) = p(x_i) \cdot p(y_j) \quad (3.13)$$

deci:

$$H(X/Y) = H(X) \quad H(Y/X) = H(Y) \quad (3.14)$$

$$H(X, Y) = H(X) + H(Y) \quad (3.15)$$

3.2 Transinformația. Proprietăți

Pornind de la expresia informației asupra lui x_i când se cunoaște y_j ca diferență între informația proprie a lui x_i și incertitudinea asupra simbolului x_i când se cunoaște y_j , $i(x_i, y_j) = U(x_i) - U(x_i/y_j)$, se calculează valoarea medie a informației mutuale care se numește **transinformație** și reprezintă informația medie ce se trimite pe canal:

$$I(X, Y) = \sum_i \sum_j i(x_i, y_j) \cdot p(x_i, y_j) \quad (3.16)$$

Se arată că:

$$\begin{aligned} I(X, Y) &= H(X) - H(X/Y) \\ I(X, Y) &= H(Y) - H(Y/X) \end{aligned} \quad (3.17)$$

Proprietățile transinformației:

1. Capacitatea canalului este valoarea maximă a transinformației prin canal:

$$C = \max\{I(X, Y)\} = \max\{H(X) - H(X/Y)\} \quad (3.18)$$

2. Capacitatea canalului raportată la timp:

$$C_t = \frac{C}{T} \quad (3.19)$$

3. Redundanța canalului:

$$R = C - I(X, Y) \quad (3.20)$$

4. Redundanța relativă:

$$\rho = 1 - \frac{I(X, Y)}{C} = \frac{R}{C} \quad (3.21)$$

5. Eficiența canalului:

$$\eta = \frac{I(X, Y)}{C} = 1 - \rho \quad (3.22)$$

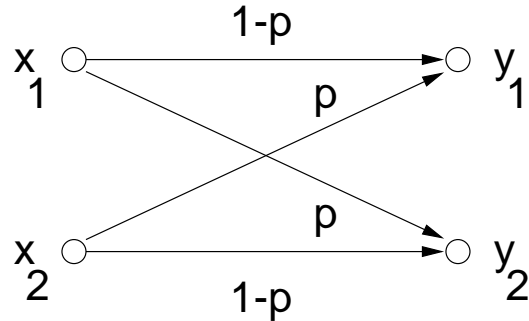


Figura 3.1: Canal binar simetric

3.3 Canalul binar simetric. Canalul binar cu anulări

Canalul binar simetric (figura 3.1) are proprietatea că probabilitatea de eronare a oricărui simbol este aceeași. Fie canalul pentru care probabilitatea de eronare este p :

Matricea de zgomot atașată (probabilitățile de eronare) este:

$$[P] = P[Y/X] = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \quad (3.23)$$

Se demonstrează că transinformația prin canal este:

$$I(X, Y) = H(Y) - H(Y/X) = H(Y) + p \cdot \log p + (1-p) \cdot \log(1-p), \quad (3.24)$$

care este maximă atunci când $H(Y)$ este maximă, capacitatea canalului fiind:

$$C = 1 + p \cdot \log p + (1-p) \cdot \log(1-p), \quad (3.25)$$

expresie care depinde numai de zgomotul de canal și nu de probabilitățile de utilizare ale simbolurilor.

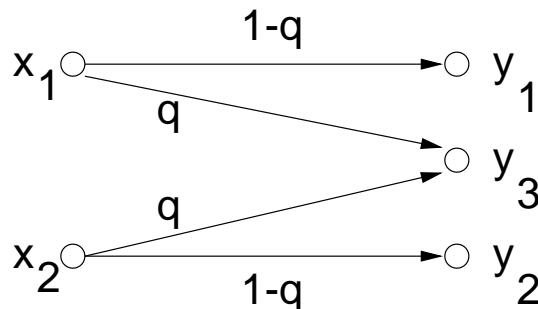


Figura 3.2: Canal binar cu anulări

Canalul binar simetric cu anulări are proprietatea că probabilitatea de anulare a oricărui simbol este aceeași. Pentru canalul din figura 3.2, matricea de zgomot este:

$$[P] = P[Y/X] = \begin{bmatrix} 1-p-q & p & q \\ p & 1-p-q & q \end{bmatrix} \quad (3.26)$$

Ținând seama de formula probabilităților condiționate:

$$p(x_i, y_j) = p(y_j/x_i) \cdot p(x_i) \quad (3.27)$$

și de relația:

$$p(y_j) = \sum_i p(x_i, y_j), \quad (3.28)$$

putem calcula probabilitățile câmpului de ieșire, apoi calculăm expresia transformărilor prin canal:

$$I(X, Y) = H(Y) - H(Y/X) \quad (3.29)$$

și în final obținem expresia capacității canalului:

$$C = 1 - q \quad (3.30)$$

3.4 Exerciții

1. Demonstrați și apoi reprezentați în MATLAB:

- (a) capacitatea canalului binar simetric;
- (b) capacitatea canalului binar cu anulări.

2. Un canal de transmisiune se caracterizează prin următoarea matrice de zgomot:

$$P[Y/X] = \begin{bmatrix} 0.98 & 0.01 & 0.01 \\ 0.1 & 0.75 & 0.15 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad (3.31)$$

La intrarea în canal se aplică simbolurile x_1, x_2, x_3 cu probabilitățile:

$$p(x_1) = 0.7 \quad p(x_2) = 0.2 \quad p(x_3) = 0.1 \quad (3.32)$$

Determinați:

- (a) cantitatea medie de informație furnizată de un simbol recepționat;
 - (b) cantitatea de informație ce se pierde la transmiterea unui mesaj format din 400 de simboluri;
 - (c) cantitatea de informație recepționată în condițiile de la punctul anterior.
3. Folosind programul CANALE.EXE, apoi realizând programul MATLAB, pentru canalul caracterizat de matricea de zgomot:

$$P[Y/X] = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix} \quad (3.33)$$

și probabilitățile sursei:

$$\begin{aligned} p(x_1) &= 3/4 \\ p(x_2) &= 1/4 \end{aligned} \quad (3.34)$$

determinați:

- (a) entropia câmpului de intrare;
- (b) entropia câmpului de ieșire;
- (c) entropia câmpurilor reunite;
- (d) eroarea medie și echivocația;
- (e) transinformația;
- (f) capacitatea canalului;
- (g) redundanța și eficiența canalului.

Laborator 4

Receptoare de simboluri discrete (cazul binar)

4.1 Criteriul lui Bayes

Pentru o sursă de simboluri discrete ce emite simboluri dintr-un alfabet X , cu probabilități date P :

$$\begin{bmatrix} X \\ P \end{bmatrix} \quad \begin{bmatrix} x_1 & x_2 \\ p(x_1) & p(x_2) \end{bmatrix} \quad (4.1)$$

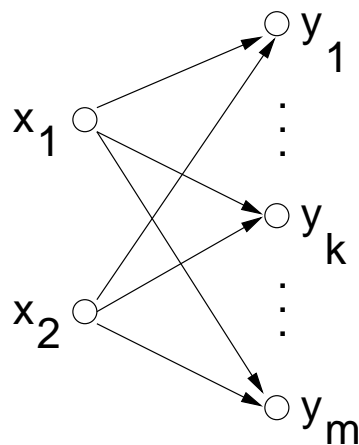


Figura 4.1: Graful de tranziție pentru un canal binar discret cu m stări în spațiul de ieșire

Se consideră un canal ce furnizează la ieșire m simboluri, $[Y] = [y_1 \dots y_m]$ (figura 4.1). Cu ajutorul criteriului lui Bayes (riscului minim) pe care îl vom da ulterior, se partitionează spațiul ieșirilor (Y) în două subspații Y_1 și Y_2 disjuncte:

$$Y = Y_1 \cup Y_2 \quad (4.2)$$

$$Y_1 \cap Y_2 = \emptyset \quad (4.3)$$

Să presupunem că la recepție se primește un simbol y_k . Acesta poate aparține fie lui Y_1 , fie lui Y_2 . Se pot construi astfel cele două decizii D_1 și D_2 :

$$D_1 = \{y_k \in Y_1\} \quad (4.4)$$

$$D_2 = \{y_k \in Y_2\} \quad (4.5)$$

Numai una din cele două decizii este corectă la un moment dat. Fiecărei decizii îi este atașat un anumit cost, în funcție și de simbolul x_j emis inițial. Partiția considerată se caracterizează cu ajutorul următoarei matrice de costuri:

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \quad (4.6)$$

Este firească presupunerea: costurile pentru o decizie corectă (D_1 când s-a emis x_1 respectiv D_2 când s-a emis x_2 , adică C_{11} și C_{22}) să fie mai mici decât costurile pentru o decizie incorectă (D_1 când s-a emis x_2 respectiv D_2 când s-a emis x_1 , adică C_{12} și C_{21}).

Facem următoarele notații:

- C_{ij} este costul deciziei D_j când s-a transmis simbolul x_i ;
- x_i este simbolul emis;
- y_k este simbolul recepționat;
- \hat{x}_j este simbolul estimat în funcție de D_j cu privire la y_k recepționat.

Criteriul de optimizare folosit în cazul receptoarelor de simboluri discrete este cel al **costului mediu (C) minim**, numit risc (R) și avînd expresia:

$$R = \sum_{i=1}^2 \sum_{j=1}^2 C_{ij} \cdot p(D_j, x_i), \quad (4.7)$$

unde $p(D_j, x_i)$ este probabilitatea luării deciziei D_j concomitent cu emiterea simbolului x_i . Presupunem că acest risc este minim pentru cazul când decizia D_1 este corectă (se emite x_1 și se observă y_k). Vom considera relațiile valabile pentru probabilității condiționate:

$$p(D_1, x_1) = p(D_1/x_1) \cdot p(x_1) \quad (4.8)$$

$$p(D_2, x_1) = p(D_2/x_1) \cdot p(x_1) \quad (4.9)$$

$$p(D_1, x_2) = p(D_1/x_2) \cdot p(x_2) \quad (4.10)$$

$$p(D_2, x_2) = p(D_2/x_2) \cdot p(x_2) \quad (4.11)$$

Expresia riscului devine atunci:

$$R = [C_{11} \cdot p(D_1/x_1) + C_{12} \cdot p(D_2/x_1)] \cdot p(x_1) + [C_{21} \cdot p(D_1/x_2) + C_{22} \cdot p(D_2/x_2)] \cdot p(x_2) \quad (4.12)$$

Deoarece evenimentele $\{D_1/x_1\}$ și $\{D_2/x_1\}$ respectiv $\{D_1/x_2\}$ și $\{D_2/x_2\}$ sunt complementare, putem scrie relațiile:

$$p(D_2/x_1) = 1 - p(D_1/x_1) \quad (4.13)$$

$$p(D_2/x_2) = 1 - p(D_1/x_2) \quad (4.14)$$

Atunci expresia riscului R se scrie:

$$R = C_{12} \cdot p(x_1) + C_{22} \cdot p(x_2) + \underbrace{p(x_2) \cdot p(D_1/x_2) \cdot (C_{21} - C_{22}) - p(x_1) \cdot p(D_1/x_1) \cdot (C_{12} - C_{11})}_{\leq 0 \text{ pentru decizia } D_1 \text{ corectă}} \quad (4.15)$$

Membrul din stânga este format din două părți, prima nu depinde de decizia luată, în schimb a doua depinde. Pentru decizie corectă, această parte trebuie să fie negativă:

$$\frac{p(D_1/x_1)}{p(D_1/x_2)} \leq \frac{p(x_2)}{p(x_1)} \cdot \frac{C_{21} - C_{22}}{C_{12} - C_{11}} \quad (4.16)$$

Revenind la notație, decizia D_1 se ia pentru y_k , deci criteriul costului (riscului) minim sau criteriul lui Bayes se scrie astfel:

$$\frac{p(y_k/x_1)}{p(y_k/x_2)} \underset{Y_2}{\overset{Y_1}{>}} \frac{p(x_2)}{p(x_1)} \cdot \frac{C_{21} - C_{22}}{C_{12} - C_{11}} \quad (4.17)$$

Vom nota partea stângă cu

$$\lambda_k = \frac{p(y_k/x_1)}{p(y_k/x_2)} \quad (4.18)$$

iar partea dreaptă, pe care o numim **pragul testului**, cu

$$K = \frac{p(x_2)}{p(x_1)} \cdot \frac{C_{21} - C_{22}}{C_{12} - C_{11}} \quad (4.19)$$

Partiționarea lui Y în Y_1 și Y_2 se va face în mod corespunzător pentru fiecare y_k :

$$\begin{aligned} \lambda_k &> K, & y_k &\in Y_1 \\ \lambda_k &< K, & y_k &\in Y_2 \end{aligned} \quad (4.20)$$

4.2 Clasificarea criteriilor

Se definesc următoarele criterii de decizie:

1. Criteriul riscului minim:

$$\frac{p(y_k/x_1)}{p(y_k/x_2)} \underset{Y_2}{\overset{Y_1}{>}} \frac{p(x_2)}{p(x_1)} \cdot \frac{C_{21} - C_{22}}{C_{12} - C_{11}} \quad (4.21)$$

2. Criteriul plauzibilității maxime:

$$\frac{p(y_k/x_1)}{p(y_k/x_2)} \underset{Y_2}{\overset{Y_1}{>}} 1 \quad (4.22)$$

3. Criteriul observatorului ideal

$$\frac{p(y_k/x_1)}{p(y_k/x_2)} \underset{Y_2}{\overset{Y_1}{>}} \frac{p(x_2)}{p(x_1)} \quad (4.23)$$

4. Criteriul probabilității a posteriori maxime:

$$\frac{p(x_1/y_k)}{p(x_2/y_k)} \underset{Y_2}{\overset{Y_1}{>}} 1 \quad (4.24)$$

4.3 Canalul echivalent

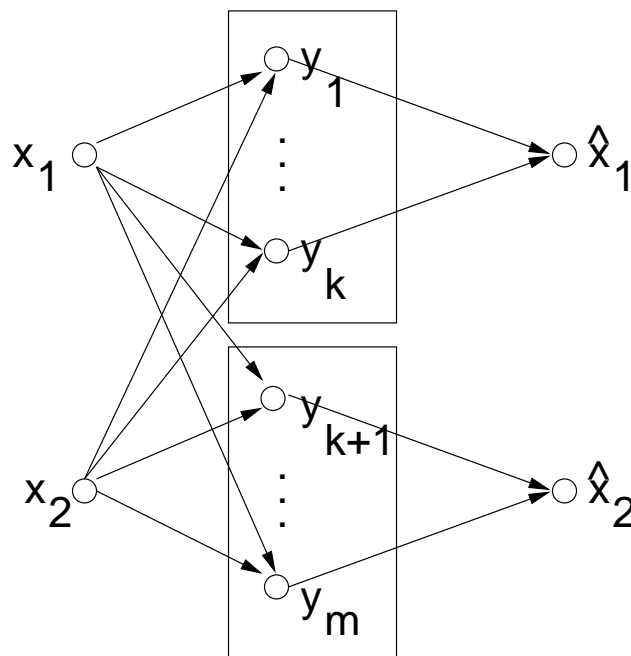


Figura 4.2: Canalul binar echivalent după realizarea partiției

Criteriile se bazează pe împărțirea spațiului observațiilor (spațiul de ieșire) în două subspații, fiecare corespunzând unei decizii. În urma partiției se estimează simbolurile \hat{x}_1 pentru y_k din Y_1 respectiv \hat{x}_2 pentru y_k din Y_2 (figura 4.2). Se obține astfel canalul binar echivalent, caracterizat de o nouă matrice de tranziție ce cuprinde caracteristicile atât ale canalului original cât și ale receptorului considerat.

Dacă se recepționează simbolurile y_k , $k = 1 \dots m$, matricea strategiei de decizie S se definește prin:

$$S = \begin{bmatrix} p(\hat{x}_1/y_1) & p(\hat{x}_2/y_1) \\ p(\hat{x}_1/y_2) & p(\hat{x}_2/y_2) \\ \dots\dots\dots & \dots\dots\dots \\ p(\hat{x}_1/y_m) & p(\hat{x}_2/y_m) \end{bmatrix} \quad (4.25)$$

Elementele lui S sunt valori booleene, 0 sau 1, corespunzând estimării corecte respectiv incorecte și reflectă în mod direct partiția făcută.

Matricea de tranziție a canalului optimizat echivalent (intrare în canalul de transmisiune - ieșire din receptorul optimal) care înglobează canalul propriu-zis și receptorul optimal este:

$$T = P[Y/X] \cdot S, \quad (4.26)$$

unde $P[Y/X]$ este matricea de zgomot a canalului inițial.

4.4 Exerciții

1. Fie un canal caracterizat de următoarea matrice de zgomot:

$$P[Y/X] = \begin{bmatrix} 1/3 & 1/3 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/3 & 1/3 \end{bmatrix} \quad (4.27)$$

La intrarea canalului se aplică simbolurile x_1 și x_2 cu probabilitățile $p(x_1) = p(x_2)$. Folosind programul `RECEPT.EXE`:

- (a) să se facă partiția simbolurilor recepționate conform criteriului plauzibilității maxime;
- (b) să se scrie matricea strategiei care caracterizează partiția;
- (c) să se scrie matricea de tranziție a canalului echivalent;
- (d) să se calculeze eroarea medie înainte și după partiție.

2. Fie un canal de transmisiune caracterizat prin următoarea matrice de zgomot:

$$P[Y/X] = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix} \quad (4.28)$$

30 LABORATOR 4. RECEPTOARE DE SIMBOLURI DISCRETE (CAZUL BINAR)

La intrarea canalului se aplică simbolurile x_1 și x_2 cu probabilitățile:

$$p(x_1) = \frac{2}{3} \quad p(x_2) = \frac{1}{3} \quad (4.29)$$

Matricea costurilor este de forma:

$$C = \begin{bmatrix} 0.2 & 0.4 \\ 0.8 & 0 \end{bmatrix} \quad (4.30)$$

Folosind **MATLAB**-ul, să se facă partiția simbolurilor și să se scrie matricea strategiei și matricea de tranziție a canalului echivalent dacă se aplică criteriul:

- (a) riscului minim;
 - (b) plauzibilității maxime;
 - (c) observatorului ideal.
3. Imaginați programul scris pentru **MATLAB** bazându-vă pe calculul matricial, pentru care, dându-se matricea de zgomot $P[Y/X]$ pentru un canal discret, probabilitățile simbolurilor de intrare $[P]$ cu un număr variabil de stări în câmpul de ieșire Y , care să calculeze:
- (a) matricea strategiei care caracterizează partiția S ,
 - (b) matricea de tranziție a canalului echivalent T ,
 - (c) eroarea medie înainte și după partiție, $H(Y/X)$ respectiv $H(\hat{X}/X)$,
- pentru fiecare dintre primele trei criterii de decizie date.
4. Care ar fi interpretarea pentru criteriul observatorului ideal ? Dar pentru criteriul plauzibilității maxime ?

Laborator 5

Coduri pentru canale fără perturbații

5.1 Noțiuni introductive

Pentru o sursă de simboluri discrete ce emite simboluri dintr-un alfabet S , cu probabilități date P :

$$\begin{bmatrix} S \\ P(S) \end{bmatrix} \quad \begin{bmatrix} s_1 & \dots & s_n \\ p(s_1) & \dots & p(s_n) \end{bmatrix}, \quad (5.1)$$

se realizează operația de codare folosind alfabetul:

$$\begin{bmatrix} X \\ P(X) \end{bmatrix} \quad \begin{bmatrix} x_1 & \dots & x_D \\ p(x_1) & \dots & p(x_D) \end{bmatrix}, \quad (5.2)$$

unde $D \leq n$. Pentru fiecare simbol emis de sursa S operația de codare presupune asocierea pentru fiecare s_i a unui cod $c_i = x_{i1} \dots x_{il_i}$, unde $l_i = |c_i|$, format dintr-o succesiune de simboluri ai alfabetului de intrare al canalului X , astfel încât să se maximizeze transinformația:

$$I(X, Y) = H(X) - H(X/Y) = H_{max}(X) - 0 = \log D \quad (5.3)$$

Prin **codare** se înțelege operația de stabilire a unei corespondențe biunivoce între simbolurile sursei primare S și șirurile c_i denumite cuvinte de cod (sau cuvinte cu sens).

Se numește **cod unic decodabil** un cod pentru care oricărei succesiuni de cuvinte de cod îi corespunde o unică succesiune de simboluri din S .

Se numește **cod separabil** un cod pentru care nu există semne de demarcare între cuvintele de cod.

Se numește **prefix** al unui cuvânt de cod $c_i = x_{i1} \dots x_{il_i}$ orice succesiune de simboluri $x_{i1} \dots x_{ir}$ pentru care $r \leq l_i$.

Se numește **cod instantaneu** un cod pentru care se poate decoda succesiunea de simboluri atașată fără a determina referințele la codurile următoare transmise. Condiția necesară și suficientă pentru ca un cod să fie instantaneu este ca nici un alt cuvânt de cod să nu fie prefix al altui cuvânt de cod (vorbim în acest caz de coduri prefix).

5.2 Lungimea medie a unui cuvânt de cod

Pentru fiecare simbol emis de sursa S se atașează prin codare un cuvânt de cod c_i astfel încât pentru un transfer de date prin canal să se minimizeze timpul total. Timpul pentru transmiterea unui cuvânt de cod c_i este:

$$\tau_i = \tau \cdot l_i, \quad (5.4)$$

unde τ este timpul transmiterii prin canal al unui simbol x_i . Se încearcă minimizarea costului mediu:

$$R(C) = \sum_{i=1}^n \tau_i \cdot p(c_i) = \tau \sum_{i=1}^n l_i \cdot p(c_i) = \tau \cdot \bar{l}, \quad (5.5)$$

unde prin \bar{l} am notat lungimea medie a unui cuvânt de cod.

Informația medie pe cuvânt de cod s_i (c_i) este egală cu produsul dintre lungimea medie a unui cuvânt de cod și informația medie pe simbol x_i din cuvântul de cod:

$$H(S) = H(C) = \bar{l} \cdot H(X) \leq \bar{l} \cdot \log D, \quad (5.6)$$

unde am ținut cont de valoarea maximă a entropiei alfabetului canalului $H(X)$. Se poate scrie atunci:

$$\bar{l} \geq \frac{H(S)}{\log D} = \bar{l}_{min} \quad (5.7)$$

Informația medie pe o literă din alfabetul codului (X) $H(S)/\bar{l}$ nu poate fi mai mare decât valoarea maximă a entropiei alfabetului codului ($\log D$).

5.3 Caracteristicile codului

Următoarele mărimi caracterizează un cod adaptat pentru un canal fără perturbații:

1. **Capacitatea codului** este valoarea maximă a entropiei alfabetului codului:

$$C = \max H(X) = \log D \quad (5.8)$$

2. **Eficiența codului** este raportul dintre lungimea medie minimă și lungimea medie a unui cuvânt de cod:

$$\eta = \frac{\bar{l}_{min}}{\bar{l}} = \frac{H(S)}{\bar{l} \cdot \log D} = \frac{H(X)}{\log D} \leq 1 \quad (5.9)$$

3. **Redundanța codului:**

$$\rho = 1 - \eta = \frac{\log D - H(X)}{\log D} \quad (5.10)$$

Se numește **cod absolut optimal** un cod pentru care eficiența codului este 1. S-a văzut că acest lucru se obține pentru $H(X)$ având valoarea maximă:

$$p(x_1) = \dots p(x_D) = \frac{1}{D} \quad (5.11)$$

Pentru acest caz, un cuvânt de cod c_i va fi emis cu probabilitatea:

$$p(s_i) = p(c_i) = p(x_{i1}) \dots p(x_{il_i}) = D^{-l_i} \quad (5.12)$$

Aceasta este condiția limită pentru teorema McMillan care precizează condiția ca un cod să fie ireductibil:

$$\sum_{i=1}^k D^{-l_i} \leq 1 \quad (5.13)$$

5.4 Codarea

Asocierea simbol emis de sursă (s_i) - cuvânt de cod (c_i) se face prin partiționarea succesivă a setului inițial de simboluri S , construind un arbore, până se ajunge ca nodurile terminale să fie constituite din mulțimi conținând câte un singur simbol (figura 5.1).

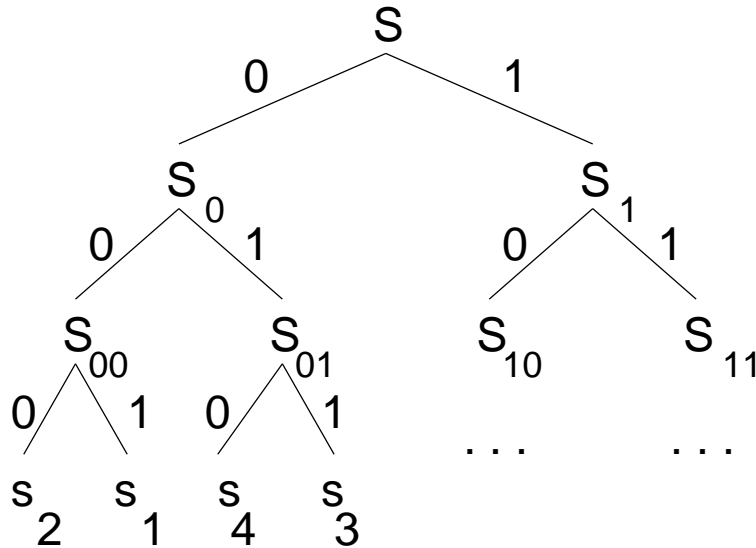


Figura 5.1: Arborele de codificare pentru un canal binar

Pentru fiecare strategie, criteriul de alcătuire al arborelui diferă:

1. **algoritmul Shannon-Fano** împarte setul inițial S în două subseturi astfel ca:

$$\begin{aligned} S_0 &= \{s_i \mid \sum p(s_i) = 2^{-1}\} \\ S_1 &= \{s_i \mid \sum p(s_i) = 2^{-1}\} \end{aligned} \quad (5.14)$$

La pasul următor fiecare subset se repartizionează astfel încât suma probabilităților pentru fiecare subset să fie 2^{-2} . Se continuă astfel până când partițiile nu mai conțin decât un singur simbol. Se etichetează fiecare ramură cu codul asociat. Pentru frunzele astfel obținute, codul unei frunze se află parcurgând arborele de la rădăcină către frunza respectivă și citind etichetele ramurilor.

2. **algoritmul Huffman** sortează simbolurile în ordinea descrescătoare a probabilităților, apoi grupează ultimele două simboluri cu probabilitățile cele mai reduse, pentru care asociază un nou simbol. S-a format astfel un subarbore. Se continuă astfel până când se construiește tot arborele de codificare.

Algoritmul Shannon-Fano construiește arborele pornind de la rădăcină spre frunze (top-down), față de algoritmul Huffman care construiește pornind de la frunze (abordare bottom-up).

5.5 Exerciții

1. Se consideră o sursă discretă descrisă de alfabetul:

$$[S] = [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6] \quad (5.15)$$

și probabilitățile:

$$[P] = [1/2 \ 1/4 \ 1/8 \ 1/16 \ 1/32 \ 1/32] \quad (5.16)$$

Să se determine:

- (a) un cod compact folosind algoritmul de codare Shannon-Fano, dacă alfabetul codului este $[X] = [0 \ 1]$;
 - (b) lungimea medie \bar{l} a unui cuvânt de cod;
 - (c) eficiența și redundanța codului.
2. Se consideră o sursă cu alfabetul $[S] = [s_1 \dots s_6]$ și probabilitățile:

$$[P] = [0,05 \ 0,1 \ 0,3 \ 0,25 \ 0,1 \ 0,2] \quad (5.17)$$

- (a) să se determine un cod compact folosind codul Huffman dacă alfabetul codului este $[X] = [0 \ 1]$ și dacă alfabetul codului este $[X] = [0 \ 1 \ 2]$;
- (b) pentru cele două cazuri să se calculeze lungimea medie a unui cuvânt de cod și eficiența codului.

Folosiți pentru aceasta programul HUFFMAN.EXE.

3. Pentru șirul "STAREA ACEASTA CERE STARI" să se realizeze codificarea Huffman folosind un canal binar, folosind programul HUFFMAN.EXE. Construiți arborele iar apoi șirul ce reprezintă codificarea sa.

Laborator 6

Coduri pentru canale cu perturbații I (coduri grup)

6.1 Noțiuni introductive

Simbolurile emise de către o sursă discretă S sunt transmise pe un canal cu perturbații. Pentru detecția și corecția la recepție a erorilor sunt necesare, pe lângă simbolurile de informație, și simboluri de control. Se transmite astfel informație redundantă dar care realizează un plus de siguranță.

Codurile pentru canalele cu perturbații se clasifică în:

- **coduri bloc**, la care prelucrarea la recepție în scopul detecției / corecției erorilor se face pe blocuri de n simboluri, și ele pot fi:
 - **coduri grup**, la care cuvintele de n simboluri sunt considerate elemente ale unui spațiu vectorial;
 - **coduri ciclice**, la care cuvintele de n simboluri sunt considerate elemente ale unei algebre.
- **coduri convoluționale**, pentru care prelucrarea la recepție se face în flux continuu.

Fie un spațiu vectorial V presupus a conține numai cuvintele cu sens (coduri pentru simbolurile emise de sursa S) și fie un alt spațiu vectorial W care cuprinde toate cuvintele de cod, atât cele cu sens cât și cele introduse redundant pentru a asigura corecția / detecția erorilor. Bineînțeles avem $V \subset W$.

Se definește, pentru două cuvinte de cod α și β de lungime n , **distanța de cod Hamming** între cele două cuvinte de cod, astfel:

$$d(\alpha, \beta) = \sum_{i=1}^n \alpha_i \oplus \beta_i, \quad (6.1)$$

ce are ca semnificație numărul de simboluri de 0 sau 1 prin care cele două cuvinte de cod diferă. De exemplu cuvintele de cod $[0 \ 1 \ 1 \ 0]$ și $[1 \ 0 \ 1 \ 0]$ diferă prin 2 biți, deci distanța Hamming dintre ele este 2.

Presupunem că la transmitere canalul binar simetric cu probabilitatea de eronare p eronează un cuvânt cu sens și se recepționează cuvântul de cod v_i' . Nu știm care cuvânt a fost transmis inițial, v_i sau v_j . Probabilitatea de a se fi emis v_i când s-a recepționat v_i' se scrie pe baza distanței Hamming astfel:

$$p(v_i|v_i') = p^{d(v_i, v_i')} \cdot p^{n-d(v_i, v_i')} \simeq p^{d(v_i, v_i')} \quad (6.2)$$

pentru o probabilitate de eronare scăzută (tipică pentru canalele obișnuite). Atunci dacă se emite v_i avem succesiv:

$$p(v_j|v_i') < p(v_i|v_i') \quad (6.3)$$

$$p^{d(v_j, v_i')} < p^{d(v_i, v_i')} \quad (6.4)$$

$$d(v_j, v_i') > d(v_i, v_i'), \quad (6.5)$$

adică distanța Hamming dintre v_i' și v_i este mai mică decât cea dintre v_i' și v_j .

Pe baza definiției distanței Hamming se fac observațiile următoare:

- pentru detecția a e erori, distanța între două cuvinte de cod trebuie să fie cel puțin $e + 1$;
- pentru corecția a e erori, distanța între două cuvinte de cod trebuie să fie cel puțin $2e + 1$.

Pentru fiecare din cazurile de mai sus, dacă se eronează cel mult e biți, va rezulta un cuvânt de cod care nu este cuvânt cu sens, care ne indică sigur eronarea și pentru al doilea caz permite și aprecierea cuvântului cu sens inițial deoarece distanța dintre cuvântul eronat și cel inițial este cel mult e .

6.2 Mecanismul de detecție și corecție a erorilor

Pe mulțimea cuvintelor de cod W definim o funcție de corecție $f : W \rightarrow Z$, unde Z se numește **spațiul corectorilor**. Un corector z indică poziția dintr-un cuvânt de cod unde s-au produs erori. Pentru un cuvânt cu sens $v \in V$ corectorul asociat este $H\{v\} = 0$. Deci dacă pentru un cuvânt de cod v' avem $H\{v'\} = 0$, înseamnă că nu s-au produs erori.

Definim aplicația $D : Z \rightarrow E$, unde $\varepsilon \in E$, ε se numește vector eroare și indică biții care s-au eronat dintr-un cuvânt de cod, iar E este spațiul corectorilor. La recepție perturbațiile peste cuvântul de cod pot fi percepute ca zgomot aditiv:

$$v' = v + \varepsilon \quad v = v' + \varepsilon \quad (6.6)$$

Se definește **matricea de control** H , iar corectorul pentru un cuvânt de cod primit v' va fi calculat cu relația:

$$z = H \cdot v'^T \quad (6.7)$$

Matricea de control va avea m linii și n coloane, unde $n = m + k$, 2^m este numărul corectorilor iar 2^k numărul simbolurilor de informație.

Un cuvânt de cod $v = [c \ i]$ este format din biții de control c și biții de informație i . Dacă matricea de control se scrie sub formă canonică, $H = [I_m \ Q]$, unde I_m este matricea identitate, relația de codare pentru un cuvânt de cod se deduce din condiția ca pentru un cuvânt cu sens corectorul asociat să fie nul:

$$H \cdot v^T = 0 \quad (6.8)$$

$$[I_m \ Q] \cdot [c \ i]^T = 0 \quad (6.9)$$

$$I_m \cdot c^T + Q \cdot i^T = 0 \quad (6.10)$$

$$c^T = Q \cdot i^T \quad (6.11)$$

$$c = i \cdot Q^T, \quad (6.12)$$

care se numește relație de codare.

Pentru codarea a N simbolii sunt necesari k biți, unde $2^k \geq N$. Pentru ca un cod cu o anumită matrice de control să corecteze e erori, este necesar să obținem corectori distincți pentru toate erorile ce pot apărea. Corectorii îi vom coda cu m biți. Atunci numărul de corectori este:

$$2^m \geq 1 + \sum_{i=1}^e C_n^i \quad (6.13)$$

Pentru un cuvânt eronat corectorul este $z = H \cdot v'^T = H \cdot \varepsilon^T$. Alegînd două cuvinte de eroare cu e erori, corectorii asociați trebuie să fie diferiți. Se demonstrează că:

- pentru corecția a e erori, suma a oricăror $2e$ coloane trebuie să fie nenulă, precum și suma a oricăror mai puțin de $2e$ coloane;
- pentru detecția a e erori, suma a oricăror e coloane trebuie să fie nenulă, precum și suma a oricăror mai puțin de e coloane.

Pentru detecția unui număr de $2p + 1$ erori, putem lua matricea de control:

$$H = \begin{bmatrix} h_1 & h_2 & \dots & h_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (6.14)$$

Cuvântul de eroare $\varepsilon = [\dots \ \alpha_{i1} \ \dots \ \alpha_{i2p+1} \ \dots]$ are asociat corectorul:

$$z = H \cdot \varepsilon^T = \begin{bmatrix} h_{i1} \\ 1 \end{bmatrix} + \dots + \begin{bmatrix} h_{i2p+1} \\ 1 \end{bmatrix}, \quad (6.15)$$

unde $h_{i1} \dots h_{i2p+1}$ sunt $2p + 1$ din coloanele matricii de control.

Dacă dorim un cod care să detecteze și erorile duble, alegem matricea de control astfel:

$$H = \begin{bmatrix} 0 & h_1 & h_2 & \dots & h_n \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \quad (6.16)$$

Atunci expresia corectorului devine:

$$z = H \cdot v'^T = \begin{bmatrix} 0 & h_1 & h_2 & \dots & h_n \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} a_0' \\ a_1' \\ \dots \\ a_n' \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \quad (6.17)$$

unde c_1 este corectorul propriu-zis pentru matricea de control ale cărei coloane sunt $h_1 \cdots h_n$, iar c_2 este bitul de paritate. Atunci:

- dacă $c_1 = 0$ și $c_2 = 0$, nu există erori;
- dacă $c_1 \neq 0$ și $c_2 \neq 0$, avem o eroare corectabilă prin c_1 sau 3 erori;
- dacă $c_1 = 0$ și $c_2 \neq 0$, s-a eronat bitul de paritate a_0 ;
- dacă $c_1 \neq 0$ și $c_2 = 0$ avem o eroare dublă.

6.3 Codarea cu ajutorul matricii generatoare

În cazul codării cu ajutorul unei matrici generatoare G dorim să găsim o matrice G de forma:

$$v = i \cdot G \quad (6.18)$$

Deoarece matricea de control H poate fi pusă în forma canonică $H = [I_m \quad Q]$, se demonstrează că matricea G are expresia:

$$G = [Q^T \quad I_k] \quad (6.19)$$

6.4 Codul Hamming grup corector de o eroare

Matricea de control asociată codului Hamming are fiecare coloană formată din reprezentarea binară a numărului acelei coloane. De exemplu pentru $n = 3$, aceasta este de forma:

$$H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad (6.20)$$

Acest cod codează 2 simboluri ($k = 1$) și are 4 corectori ($m = 2$). Cuvântul de cod va fi de forma (nu este un cod perfect):

$$v = [c_1 \quad c_2 \quad i_3] \quad (6.21)$$

Se numește **cod perfect** un cod pentru care biții de control c sunt situați pe poziții consecutive în cuvântul de cod. Pentru un astfel de cod matricea de control este în formă canonică.

Pentru a alcătui schema codorului, cu reprezentarea lui v de mai sus se pune condiția ca pentru un cuvânt de cod corectorul asociat să fie nul:

$$H \cdot v^T = 0, \quad (6.22)$$

de unde se deduc expresiile biților de control c_i .

Pentru schema decodorului, se construiește mai întâi corectorul pentru un cuvânt de cod recepționat:

$$z = H \cdot v'^T \quad (6.23)$$

Acesta va indica poziția bitului eronat, adică este coloana matricii de control cu același indice ca și bitul eronat. În particular pentru codul Hamming, va fi reprezentarea binară a indicelui bitului eronat. Acesta este apoi introdus într-un decodificator pentru a forma cuvântul de eroare.

6.5 Exerciții

1. Un număr de 16 simboluri se transmit pe un canal binar cu perturbații folosind un cod Hamming grup corector de o eroare. Se cer:
 - (a) numărul de simboluri de informație și numărul de simboluri de control;
 - (b) formați în **MATLAB** matricea de control H și cuvintele de cod;
 - (c) determinați matricea generatoare G ;
 - (d) realizați schema codorului și a decodorului, apoi în **MATLAB** realizați procedurile corespunzătoare codării și decodării;
 - (e) stabiliți apoi cu procedurile realizate expresia corectorului când se eronează al patrulea simbol. Dar dacă se eronează simbolurile 2 și 7 ?
2. Pentru același număr de 16 simboluri se utilizează acum un cod Hamming care are și proprietatea că detectează erorile duble.
 - (a) deduceți forma matricii de control și a cuvântului de cod;
 - (b) experimentați și explicați apoi funcționarea codorului și a decodorului prezentate în programul **HAMMING.EXE**;
 - (c) realizați procedurile **MATLAB** pentru codor și decodor.
3. Un număr de 8 simboluri se transmit cu ajutorul unui cod Hamming grup corector de o eroare și detector de erori duble. Se cere:
 - (a) să se determine numărul simbolurilor de informație k , al celor de control m și lungimea cuvântului de cod n ;
 - (b) să se scrie matricea de control H a codului;
 - (c) să se stabilească expresia corectorului corespunzător eronării simbolului c_2 ;

40LABORATOR 6. CODURI PENTRU CANALE CU PERTURBAȚII I (CODURI GRUP)

- (d) să se determine corectorul corespunzător eronării simbolurilor c_2 și c_1 ;
- (e) să se scrie cuvintele de cod;
- (f) să se precizeze dacă $v = [1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0]$ este un cuvânt al acestui cod.

Laborator 7

Coduri pentru canale cu perturbații II (coduri ciclice)

7.1 Noțiuni introductive

Codurile ciclice sunt coduri bloc în care cele n simboluri care formează un cuvânt de cod sunt considerate ca fiind coeficienți binari ai unui polinom $v(x)$ de grad $n - 1$. Ele au proprietatea că orice permutare ciclică a unui cuvânt cu sens este tot un cuvânt cu sens.

Cuvintele de cod sunt elemente ale unei algebre liniare modulo $p(x) = x^n + 1$. Cuvintele cu sens (în număr de 2^k) sunt elemente ale idealului generat de polinomul ireductibil și primitiv $g(x)$ numit polinom generator al polinomului $p(x)$, de grad m . Orice cuvânt cu sens este multiplu al polinomului generator:

$$v(x) = q(x) \cdot g(x) \quad (7.1)$$

Polinomul generator este de forma:

$$g(x) = g_0 + g_1x + \dots + g_{m-1}x^{m-1} + x^m, \quad (7.2)$$

unde $g_m = 1$.

Cuvintele fără sens (în număr de $2^n - 2^k$) alcătuiesc cele 2^m clase de resturi modulo $p(x)$.

Codurile ciclice fac parte din categoria codurilor grup deci relațiile matriciale demonstrate pentru codurile grup rămân valabile.

Un cuvânt de cod $v(x)$ este format dintr-o parte de control și o parte de informație:

$$v(x) = \underbrace{a_0 + a_1x + \dots + a_{m-1}x^{m-1}}_{c(x)} + \underbrace{a_mx^m + \dots + a_{m+k-1}x^{m+k-1}}_{i(x)} \quad (7.3)$$

respectiv

$$v(x) = c(x) + x^m \cdot i(x) \quad (7.4)$$

dar

$$v(x) = q(x) \cdot g(x), \quad (7.5)$$

deci

$$x^m \cdot i(x) = q(x) \cdot g(x) + c(x) \quad (7.6)$$

adică polinomul de control se socotește ca rest al împărțirii polinomului de informație multiplicat cu x^m și polinomul generator:

$$c(x) = \text{rest} \left\{ \frac{x^m \cdot i(x)}{g(x)} \right\} \quad (7.7)$$

Codul Hamming ciclic este un cod perfect (numărul cuvintelor de cod este $n = 2^m - 1$), corector de o eroare (distanța minimă între oricare două cuvinte cu sens este 3) și sistematic, adică simbolurile de informație sunt delimitate de simbolurile de control și plasate la sfârșitul cuvântului de cod.

Decodarea constă în găsirea unei corespondențe între cuvântul recepționat $v'(x)$ și erorile introduse de canal $e(x)$:

$$v'(x) = v(x) + e(x) \quad (7.8)$$

Expresia corectorului asociat este:

$$s'(x) = \text{rest} \left\{ \frac{v'(x)}{g(x)} \right\} = \text{rest} \left\{ \frac{e(x)}{g(x)} \right\} \quad (7.9)$$

Codurile ciclice cu distanța minimă 3 pot fi utilizate pentru corecția unei erori sau detecția erorilor duble prelucrând valoarea corectorului.

7.2 Codarea cu circuit de divizare

Codorul este format dintr-un circuit de divizare cu celule binare de memorare pe 1 bit și sumatoare modulo 2 interioare. Circuitul de divizare are conexiunile buclei de reacție realizate conform coeficienților polinomului generator $g(x)$. Schema bloc este prezentată în figura 7.1.

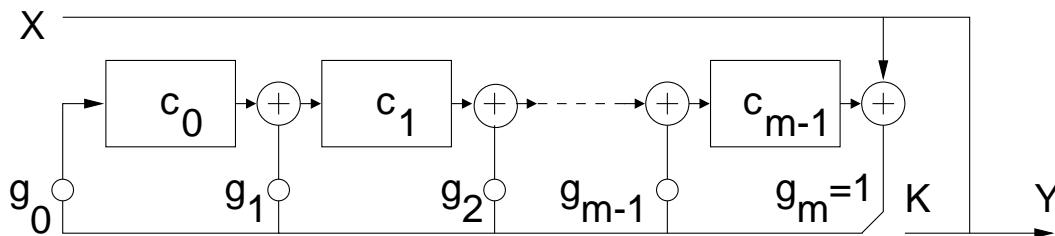


Figura 7.1: Codor cu circuit de divizare prin $g(x)$

Pe durata primelor k tacte comutatorul K se permite introducerea celor k biți de informație atât în celulele divizorului cât și copierea lor nealterată la ieșire. După $m < k$ tacte în celulele divizorului începe calculul restului împărțirii la polinomul generator.

După cele k tacte, poziția comutatorului se inversează astfel încât la ieșire se vor copia biții restului calculați în celulele divizorului, adică cei m biți de control.

Schema codorului este o variantă a celei prezentate la curs. Justificarea calculului restului se poate face prin considerarea circuitului ca un circuit cu celule de întârziere ca în figura 7.2: Aplicând transformata Z, putem scrie că circuitul calculează succesiv:

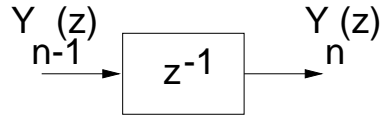


Figura 7.2: Celulă de întârziere (memorare de 1 bit)

$$Y(z) = X(z) + z^{-1}[g_{m-1}Y(z) + Y_{n-1}(z)] \quad (7.10)$$

sau

$$Y(z) = X(z) + z^{-1}g_{m-1}Y(z) + z^{-1}Y_{n-1}(z) \quad (7.11)$$

sau, recursiv:

$$Y(z) = X(z) + z^{-1}g_{m-1}Y(z) + z^{-2}g_{m-2}Y(z) + z^{-2}Y_{n-2}(z) \quad (7.12)$$

...

$$Y(z) = X(z) + z^{-1}g_{m-1}Y(z) + \dots + z^{-m}g_0Y(z) \quad (7.13)$$

$$Y(z) = X(z) + [z^{-1}g_{m-1} + \dots + z^{-m}g_0] \cdot Y(z) \quad (7.14)$$

$$Y(z) = \frac{X(z)}{g_m + z^{-1}g_{m-1} + \dots + z^{-m}g_0} = \frac{X(z)}{z^{-m} \cdot g(z)} = \frac{z^m \cdot X(z)}{g(z)} \quad (7.15)$$

7.3 Detecția erorilor cu circuite de divizare

Decodorul pentru detecția erorilor este eficient pentru detecția pachetelor de erori de lungime maximă m .

Se introduc toți biții cuvântului recepționat. După n tacte, în circuitul de divizare se va afla restul împărțirii polinomului cuvântului la polinomul generator $g(x)$, din considerentele arătate mai sus. Dacă acest rest este 0, ieșirea porții SAU-NU este 1, deci cuvântul recepționat este corect. Ieșirea porții se poziționează pe 0 când în cuvântul recepționat s-au introdus erori, indiferent de poziția acestora.

7.4 Corecția erorilor cu circuite de divizare

Decodorul cu corecția erorilor este format dintr-un registru de deplasare în inel, de lungime n , 3 porți logice ȘI, un circuit de divizare cu m celule și un multiplexor.

Registrul de deplasare are rolul de a stoca cei n biți recepționați, necesari pentru efectuarea corecției. MUX-ul este deschis pe perioada primelor n tacte și în registrul de divizare se vor stoca biții cuvântului recepționat.

Pe perioada următoarelor n tacte se obține cuvântul corectat. Bitul de ieșire se va calcula ca suma modulo 2 dintre bitul corespunzător din cuvântul recepționat (stocat în poziția cea mai puțin semnificativă din registru) și un bit care are valoarea 1 numai când acest bit din cuvântul recepționat este eronat. Deci momentul când bitul eronat a ajuns în celula cea mai puțin semnificativă din registru este identificat cu ajutorul unui detector al stării circuitului de divizare.

În ultimele n tacte MUX-ul permite înscrierea în celula cea mai semnificativă din registru a ieșirii, deci conținutul registrului este rotit de la dreapta la stânga în inel.

Decodorul poate corecta o eroare indiferent de poziția acesteia.

7.5 Exerciții

1. Fie polinomul de informație $i(x) = i_0 + i_1x + i_2x^2 + i_3x^3 + i_4x^4$ și polinomul generator $g(x) = 1 + x + x^3$. Construiți schema codorului cu circuite de divizare particularizat pentru acest polinom generator. Calculați apoi teoretic restul împărțirii și iterați apoi schema realizată pentru a verifica corectitudinea ei, introducând succesiv simbolurile de informație.
2. Folosind programul `HAMM.EXE`, realizați:
 - (a) pentru polinomul generator $g(x) = 1 + x + x^3$ stabiliți simbolurile de control pentru două polinoame de informație, notându-se tabelul evoluției stărilor celulelor circuitului de divizare.
 - (b) comparați rezultatele obținute cu valorile teoretice.
 - (c) folosind decodorul cu circuit de divizare pentru detecția erorilor, decodați cuvintele obținute la punctul anterior. Notați tabelul evoluției stărilor celulelor circuitului de divizare.
 - (d) folosiți acum decodorul cu circuit de divizare pentru corecția erorilor. Notați tabelul evoluției stărilor celulelor circuitului de divizare.
 - (e) realizați subpunctele anterioare pentru aceleași polinoame de informație, pentru una, două și trei erori.
 - (f) realizați acum subpunctele anterioare pentru polinomul generator $g(x) = 1 + x + x^4$.
3. Ce conțin celulele codorului cu circuit de divizare după primele k tacte ?
4. Care este numărul maxim de erori detectate de decodorul cu circuit de divizare cu detecția erorilor ?

Laborator 8

Coduri pentru canale cu perturbații III (coduri ciclice)

8.1 Noțiuni introductive

Restul împărțirii polinomului de informație la polinomul generator $g(x)$ se poate calcula cu ajutorul unui registru de deplasare cu reacție. Codul astfel obținut este, ca și în cazul codorului cu celule de divizare, un cod sistematic.

8.2 Codor cu registru de deplasare cu reacție

Registrul de deplasare este format din $m = n - k$ celule cu conexiunile conform polinomului generator:

$$g(x) = g_0 + g_1x + g_2x^2 + \dots + g_mx^m \quad (8.1)$$

Starea inițială a celulelor este nulă. Conform figurii 8.1, notînd cu $'$ starea următoare, putem exprima starea următoare a celulelor registrului astfel:

$$\begin{aligned} c_0' &= c_1 \\ c_1' &= \quad \quad \quad c_2 \\ \dots & \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ c_{m-2}' &= \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad c_{m-1} \\ c_{m-1}' &= g_0c_0 + g_1c_1 + \dots + g_{m-1}c_{m-1} \end{aligned} \quad (8.2)$$

Dacă notăm starea curentă a registrului cu:

$$S = \begin{bmatrix} c_0 \\ c_1 \\ \dots \\ c_{m-1} \end{bmatrix}, \quad (8.3)$$

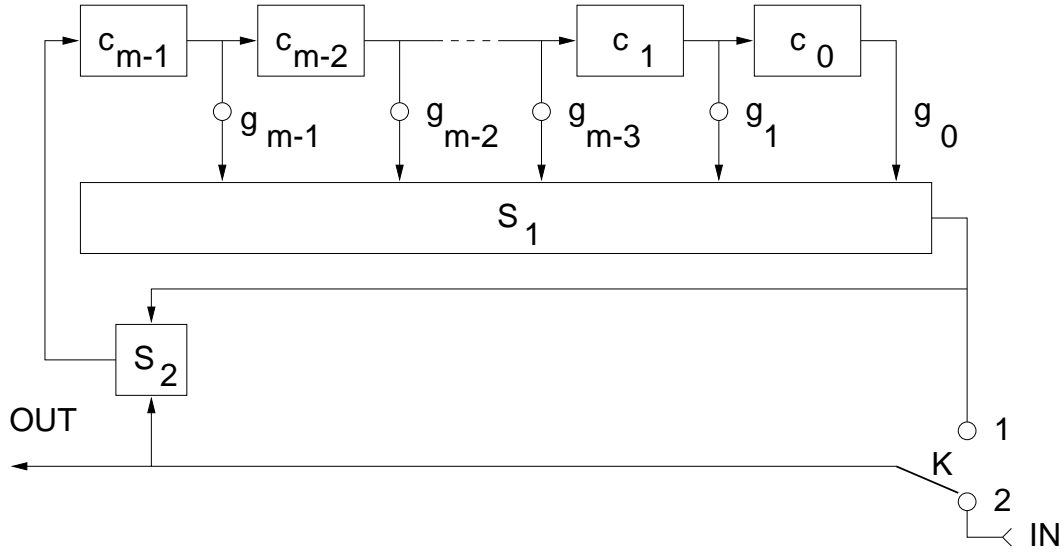


Figura 8.1: Codor cu registru de deplasare cu reacție

starea următoare a registrului cu:

$$S' = \begin{bmatrix} c_0' \\ c_1' \\ \dots \\ c_{m-1}' \end{bmatrix} \quad (8.4)$$

și cu T matricea:

$$T = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ g_0 & g_1 & g_2 & \dots & g_{m-2} & g_{m-1} \end{bmatrix} \quad (8.5)$$

atunci starea următoare se exprimă funcție de starea curentă astfel:

$$S' = T \cdot S \quad (8.6)$$

Fie notația:

$$U = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix} \quad (8.7)$$

cu m elemente. Iată cum evoluează circuitul:

- comutatorul se găsește pe poziția 1 și se introduc cele k simboluri de informație a_{n-1}, \dots, a_{n-k} care apar și la ieșire:

tact	starea registrului
1	$a_{n-1}U$
2	$a_{n-1}TU + a_{n-2}U$
...	...
k	$a_{n-1}T^kU + a_{n-2}T^{k-1}U + \dots + a_{n-k}U$

- comutatorul trece în poziția 2, la intrarea în registru se obține după fiecare deplasare simbolul 0. Celulele registrului vor conține coeficienții restului (biții de control) care vor fi transmiși la ieșire:

tact	starea registrului
k+1	$a_{n-1}T^{k+1}U + a_{n-2}T^kU + \dots + a_{n-k}TU + a_{n-k-1}U$
...	...
n-1	$a_{n-1}T^{n-1}U + a_{n-2}T^{n-2}U + \dots + a_1TU + a_0U$

unde a_{n-k-1}, \dots, a_0 sunt simbolurile de control. Circuitul revine după n tacte în starea inițială adică 0.

Dacă scriem relația matricial, avem:

$$[U \ TU \ T^2U \ \dots \ T^{n-2}U \ T^{n-1}U] \cdot [a_0 \ a_1 \ \dots \ a_{n-1}]^T = 0 \quad (8.8)$$

adică

$$H \cdot v^T = 0, \quad (8.9)$$

unde $H = [U \ TU \ T^2U \ \dots \ T^{n-2}U \ T^{n-1}U]$ este matricea de control și $v = [a_0 \ a_1 \ \dots \ a_{n-1}]$ este vectorul de cod.

8.3 Decodor cu registru de deplasare cu reacție

Decodorul este format dintr-un registru de memorare RM , două decodoare identice $DC1$ și $DC2$ cu registru de deplasare cu reacție care funcționează în tandem și o poartă SAU. Schema decodorului apare în figura 8.2.

Funcționarea se desfășoară în două etape:

- comutatorul se află pe poziția 1 și în primele n tacte cuvântul recepționat $v' = [a_0' \ \dots \ a_{n-1}']$ este pe de o parte înmagazinat în registrul de memorare RM iar pe de altă parte este încărcat în decodorul $DC1$. După cele n tacte $DC1$ va conține corectorul corespunzător cuvântului recepționat:

$$z' = H \cdot v'^T = a_0'U + a_1'TU + \dots + a_{n-1}'T^{n-1}U \quad (8.10)$$

- comutatorul se pune în poziția 2 și ieșirea se obține însumând modulo 2 bitul din celula cea mai puțin semnificativă a registrului de memorare RM cu un bit (calculat pe baza corectorului) care are valoarea 1 dacă bitul corespunzător ca poziție din cuvântul recepționat este eronat.

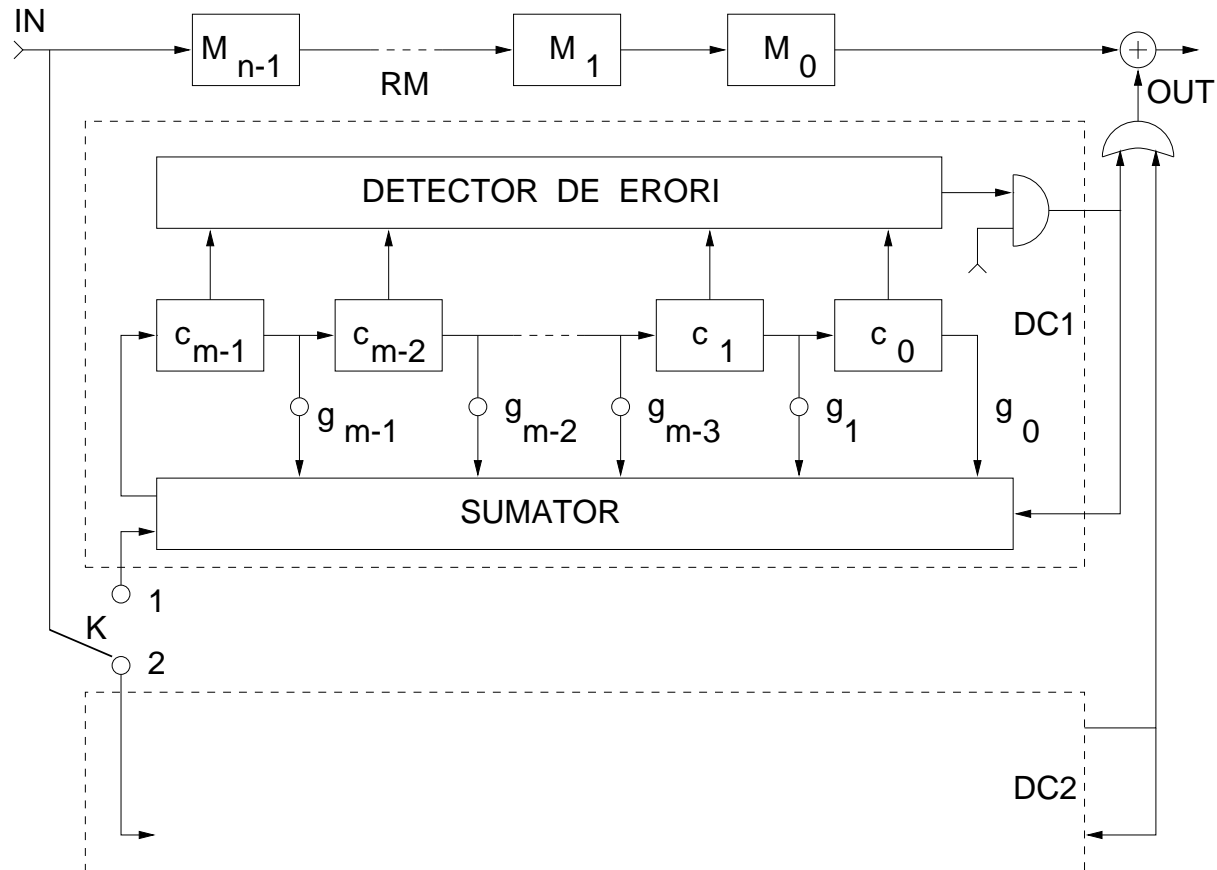


Figura 8.2: Decodor cu registru de deplasare cu reacție

Decodorul $DC2$ execută același ciclu de $n + n$ tacte dar în opoziție față de $DC1$: când $DC1$ calculează corectorul cuvântului recepționat care tocmai intră în RM , $DC2$ efectuează operația de corecție asupra cuvântului recepționat anterior care iese din RM , respectiv invers. Operația de decodare decurge astfel în flux continuu.

Decodorul poate corecta o eroare indiferent de poziția acesteia.

8.4 Exerciții

1. Folosind programul `HAMM.EXE`, realizați:
 - (a) pentru polinomul generator $g(x) = 1 + x + x^3$ se stabilesc simbolurile de control pentru două polinoame de informație, notându-se tabelul evoluției stărilor celulelor registrului de deplasare cu reacție.
 - (b) se compară valorile obținute cu rezultatele teoretice.
 - (c) se decodează cuvântul obținut. Se notează tabelul evoluției stărilor celulelor registrului de deplasare cu reacție.

- (d) se repetă punctul anterior pentru o eroare, două și trei erori, indiferent de poziția acestora.
 - (e) se repetă punctele anterioare pentru polinomul generator $g(x) = 1 + x + x^4$.
2. De ce, atunci când comutatorul K este în poziția 2 la codorul cu registru de deplasare cu reacție, intrarea în registru este 0 ?
 3. Când este 1 bitul de corecție generat de detectorul de erori al decodorului cu registru de deplasare cu reacție pentru corecția erorilor ?
 4. Cum acționează circuitele de corecție atunci când în cuvântul recepționat sunt două sau mai multe erori ?

Laborator 9

Transformata Fourier Rapidă

9.1 Noțiuni introductive

Transformata Fourier discretă (FFT) determină spectrul eșantionat al transformatei Fourier integrale.

Fie un semnal eșantionat $\{x(n) \mid n = \overline{0, N-1}\}$ de lungime N . Transformata Fourier discretă realizează trecerea de la secvența $x(n)$ la secvența $\{X(k) \mid k = \overline{0, N-1}\}$, de aceeași lungime:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N}kn}, \quad k = \overline{0, N-1} \quad (9.1)$$

Conform ecuației 9.1, ordinul timpului este în $O(N^2)$. Este necesară efectuarea unui număr de N^2 înmulțiri complexe, pe un set de date de lungime N .

Algoritmul FFT (Fast Fourier Transform) are două variante, cu decimare în timp și cu decimare în frecvență. Acest algoritim, realizat de Cooley și Tokey (1965), are ordinul timpului în $O(N \cdot \log_2 N)$, pentru N putere a lui 2.

9.2 Algoritmul FFT cu decimare în timp

Se exprimă spectrul eșantionat separat în funcție de secvența pară respectiv cea impară a semnalului:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)w_N^{2kn} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)w_N^{k(2n+1)} \quad (9.2)$$

unde avem:

$$w_N^{2kn} = e^{-j\frac{2\pi}{N}2kn} = w_{\frac{N}{2}}^{kn} \quad (9.3)$$

și atunci relația se scrie:

$$X(k) = \underbrace{\sum_{n=0}^{\frac{N}{2}-1} x(2n)w_N^{kn}}_{X_{par}(k)} + \underbrace{\left[\sum_{n=0}^{\frac{N}{2}-1} x(2n+1)w_N^{kn} \right]}_{X_{impar}(k)} \cdot w_N^k = X_{par}(k) + w_N^k \cdot X_{impar}(k) \quad (9.4)$$

Cu $X_{par}(k)$ am notat transformata Fourier discretă a eșantioanelor pare ale secvenței $x(n)$ iar cu $X_{impar}(k)$ transformata Fourier discretă a eșantioanelor impare ale secvenței $x(n)$.

Secvența FFT are proprietatea că este simetrică față de mijlocul secvenței, astfel încât pentru $k = 0, \frac{N}{2} - 1$ avem:

$$\begin{aligned} X_{par}(k + \frac{N}{2}) &= X_{par}(k) \\ w_N^{k+\frac{N}{2}} \cdot X_{impar}(k + \frac{N}{2}) &= -w_N^k \cdot X_{impar}(k) \end{aligned} \quad (9.5)$$

Atunci pentru secvența de eșantioane $x(n)$, cu $n = \overline{0, N-1}$ secvența FFT asociată pentru $k = 0, \frac{N}{2} - 1$ este:

$$\begin{cases} X(k) &= X_{par}(k) + w_N^k \cdot X_{impar}(k) \\ X(k + \frac{N}{2}) &= X_{par}(k) - w_N^k \cdot X_{impar}(k) \end{cases} \quad (9.6)$$

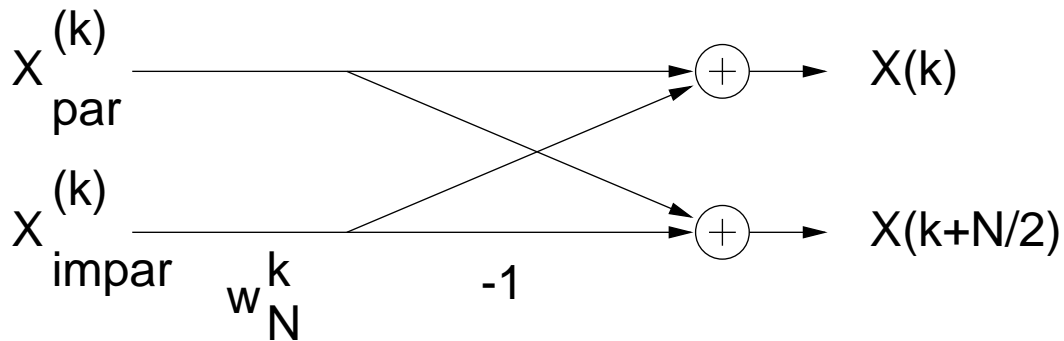


Figura 9.1: Celula fluture (butterfly) de calcul FFT

Structura unei celule pentru calculul coeficienților secvenței FFT este prezentată în figura 9.1.

Configurația pentru calculul transformatei Fourier rapide este prezentată în figura 9.2. Cu linie îngroșată s-au reprezentat arcele caracterizate de ponderi diferite de 1.

9.3 Algoritmul FFT cu decimare în frecvență

Se dezvoltă termenii pari respectiv cei impari ai transformatei Fourier discrete:

$$X(2k) = \sum_{n=0}^{N-1} x(n)w_N^{2kn} = \sum_{n=0}^{\frac{N}{2}-1} x(n)w_N^{2kn} + \sum_{n=0}^{\frac{N}{2}-1} x(n + \frac{N}{2})w_N^{2k(n+\frac{N}{2})} \quad (9.7)$$

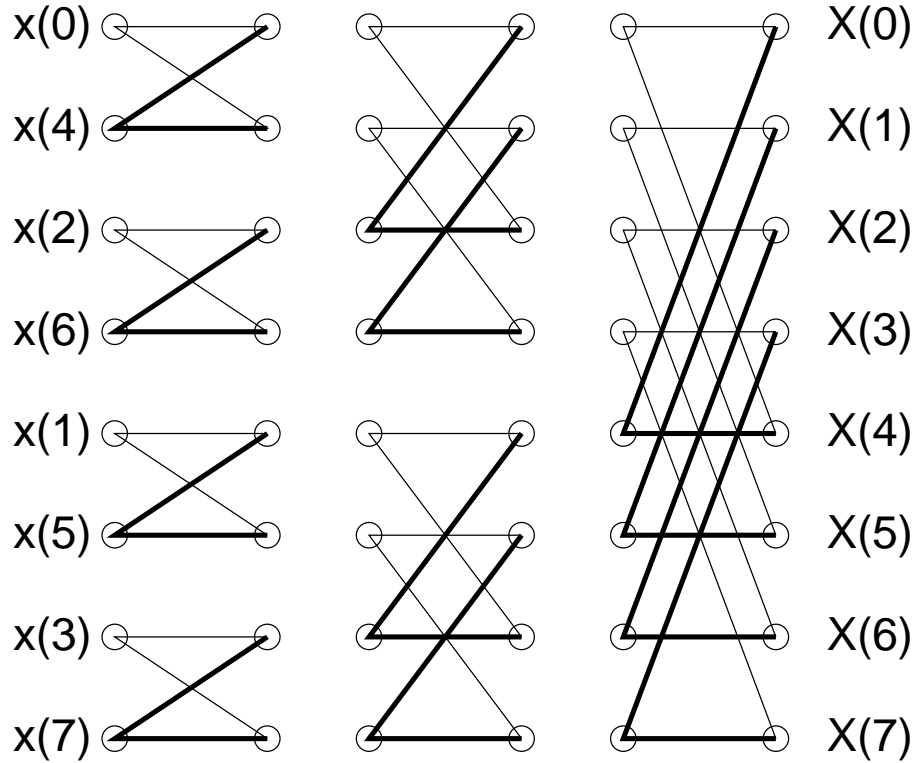


Figura 9.2: Structura pentru calculul FFT cu decimare în timp la o secvență de 8 eșantioane

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} x(n)w_{\frac{N}{2}}^{kn} + \sum_{n=0}^{\frac{N}{2}-1} x(n + \frac{N}{2})w_{\frac{N}{2}}^{kn} \quad (9.8)$$

unde, dacă notăm cu $\text{inf } x(n)$ secvența primelor $\frac{N}{2}$ eșantioane și cu $\text{sup } x(n)$ secvența ultimelor $\frac{N}{2}$ eșantioane ale secvenței inițiale, obținem:

$$X(2k) = X\{\text{inf } x(n)\} + X\{\text{sup } x(n)\} \quad (9.9)$$

Procedînd analog pentru $X(2k + 1)$, obținem:

$$X(2k + 1) = X\{\text{inf } x(n) \cdot w_N^n\} - X\{\text{sup } x(n) \cdot w_N^n\} \quad (9.10)$$

pentru $k = 0, \overline{\frac{N}{2} - 1}$.

9.4 Exerciții

1. Realizați programul MATLAB pentru calculul transformatei Fourier rapide cu decimare în timp. Considerați o secvență de date. Completați cu zerouri, dacă este cazul, pentru ca lungimea să fie putere a lui 2. Comparați rezultatul obținut cu rezultatul dat de funcția MATLAB `fft(x)`. Ce observați ?

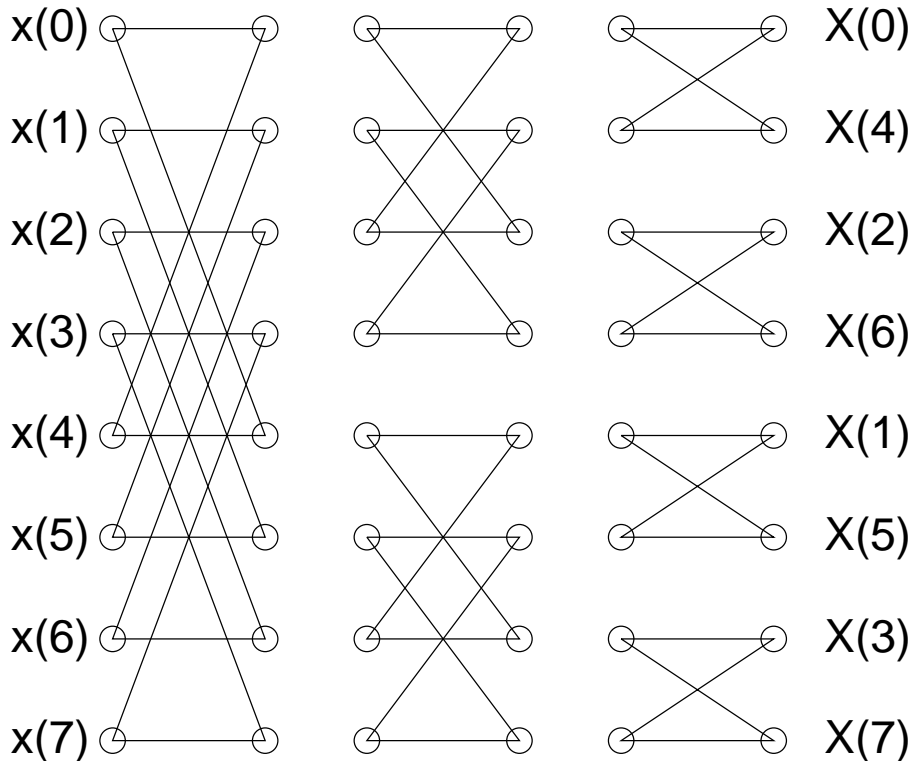


Figura 9.3: Structura pentru calculul FFT cu decimare în frecvență la o secvență de 8 eșantioane

2. Realizați algoritmul pentru calculul transformatei Fourier discrete cu decimare în frecvență. Comparați cu rezultatele obținute cu cele anterioare.
3. Pentru semnalul:

$$s(t) = 4 \cdot \sin(2\pi \cdot 50 \cdot t) + 2 \cdot \sin(2\pi \cdot 100 \cdot t) + 6 \cdot \sin(2\pi \cdot 350 \cdot t), \quad (9.11)$$

reprezentați în același grafic (comanda `subplot`) semnalul și spectrul de frecvență asociat. Vectorul frecvențelor va fi construit astfel:

$$f = [0 : N/2] * f_e / N$$

unde f_e este frecvența de eșantionare și N lungimea secvenței. Justificați alegerea considerată pentru vectorul frecvențelor. Amplitudinea semnalului va fi calculată astfel:

$$\begin{aligned} X_t &= \text{fft}(x); \\ X_m &= \text{abs}(X_t); \\ X &= X_m(1, 1 : N/2 + 1) / (N/2); \end{aligned}$$

4. Cum este spectrul de frecvență al unui semnal discret ? Dar al unui semnal discret și periodic ?

Cuprins

1	Introducere în MATLAB	1
1.1	Sesiunea MATLAB. Efectuarea calculelor	1
1.2	Operații cu vectori și matrici	2
1.3	Salvarea și restaurarea variabilelor	5
1.4	Structuri de control	5
1.5	Fișiere de comenzi (script)	6
1.6	Funcții de existență	7
1.7	Funcții pentru șiruri de caractere	7
1.8	Funcții de aproximare, random și extreme	8
1.9	Crearea graficelor	8
1.10	Documentație	10
1.11	Exerciții	10
2	Entropia unei surse binare. Surse Markov	12
2.1	Noțiunea și măsura informației	12
2.2	Cazuri extreme de canale discrete	13
2.3	Entropia. Proprietăți	13
2.4	Surse Markov	14
2.5	Problemă rezolvată	16
2.6	Exerciții	17
3	Canale discrete. Transinformația	19
3.1	Canale discrete. Entropii condiționate	19
3.2	Transinformația. Proprietăți	21
3.3	Canalul binar simetric. Canalul binar cu anulări	22
3.4	Exerciții	23
4	Receptoare de simboluri discrete (cazul binar)	25
4.1	Criteriul lui Bayes	25
4.2	Clasificarea criteriilor	27
4.3	Canalul echivalent	28
4.4	Exerciții	29

5	Coduri pentru canale fără perturbații	31
5.1	Noțiuni introductive	31
5.2	Lungimea medie a unui cuvânt de cod	32
5.3	Caracteristicile codului	32
5.4	Codarea	33
5.5	Exerciții	34
6	Coduri pentru canale cu perturbații I (coduri grup)	35
6.1	Noțiuni introductive	35
6.2	Mecanismul de detecție și corecție a erorilor	36
6.3	Codarea cu ajutorul matricii generatoare	38
6.4	Codul Hamming grup corector de o eroare	38
6.5	Exerciții	39
7	Coduri pentru canale cu perturbații II (coduri ciclice)	41
7.1	Noțiuni introductive	41
7.2	Codarea cu circuit de divizare	42
7.3	Detecția erorilor cu circuite de divizare	43
7.4	Corecția erorilor cu circuite de divizare	43
7.5	Exerciții	44
8	Coduri pentru canale cu perturbații III (coduri ciclice)	45
8.1	Noțiuni introductive	45
8.2	Codor cu registru de deplasare cu reacție	45
8.3	Decodor cu registru de deplasare cu reacție	47
8.4	Exerciții	48
9	Transformata Fourier Rapidă	50
9.1	Noțiuni introductive	50
9.2	Algoritmul FFT cu decimare în timp	50
9.3	Algoritmul FFT cu decimare în frecvență	51
9.4	Exerciții	52

Bibliografie

- [1] Mateescu, A., Dumitru, N., Stanciu, L., *Semnale și sisteme - Aplicații în filtrarea semnalelor*, Editura Teora, 2001
- [2] Murgan, A.T., Spânu, I., Gavat, I., Sztojanov, I., Neagoe, V.I., Vlad, A., *Teoria transmisiunii informației - probleme*, Editura Didactică și Pedagogică, București, 1983
- [3] Rădescu, R., Stoian, R., *Teoria Informației și a Codurilor (Teoria Transmisiunii Informației 1) - îndrumar de laborator*, Editura PRINTECH, București, 1998
- [4] Spătaru, A., *Teoria transmisiunii informației*, Editura Didactică și Pedagogică, București, 1982
- [5] * * *, *The Student Edition of MATLAB - User's guide*, Prentice Hall, 1996