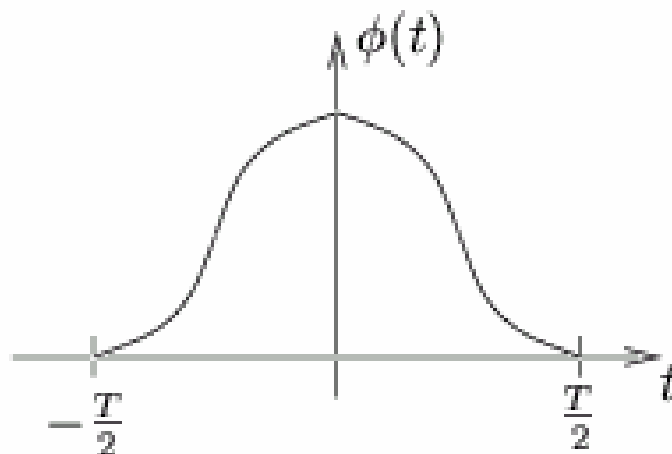


---

COMUNICAȚII DE DATE

---



ÎNDRUMĂTOR DE LABORATOR

## Cuprins

### Lucrarea nr.1

<b>Prezentarea mediului MATLAB</b> .....	3
1.1 Principalele comenzi și funcții în MATLAB.....	3
1.2 Moduri de lucru în MATLAB.....	7
1.3 Desfășurarea lucrării .....	8
1.4 Exerciții.....	10

### Lucrarea nr. 2

<b>Tehnici de codare digitală a semnalelor în banda de bază</b> .....	11
2.1 Introducere.....	11
2.2 Codarea semnalelor în banda de bază.....	11
2.3 Prezentarea unor tehnici de codare.....	11
2.4 Desfășurarea lucrării.....	15
2.4.1 Programul de simulare.....	15
2.4.2 Sarcini de îndeplinit de către studenți.....	15

### Lucrarea nr.3

<b>Modelarea matematică a sistemelor de transmisiuni în banda de bază. Trecerea unei secvențe de date printr-un filtru global tip “trece-jos ideal”</b> .....	16
3.1 Modelarea sistemelor de transmisiuni în banda de bază. Teorema lui Nyquist.....	16
3.2 Simularea pe calculator a trecerii unei secvențe de date printr-un canal echivalent de tip “trece-jos ideal”.....	18
3.2.1 Metoda de echivalare a sistemelor în timp continuu cu sisteme în timp discret prin invarianța răspunsului la impuls	18
3.2.2 Răspunsul unui sistem în timp discret la o secvență de intrare $x[n]$ .....	18
3.2.3 Generarea cu calculatorul a unei secvențe de date aleatoare.....	19
3.2.4 Răspunsul în frecvență al sistemului discret echivalent tip “filtru trece-jos ideal”.....	20
3.2.5 Secvența de la ieșirea sistemului numeric și interpretarea acesteia.....	21
3.3 Desfășurarea lucrării.....	22

### Lucrarea nr.4

<b>Transmisii de date în banda de bază fără interferență intersimbol, canalul fiind fără zgomot</b> .....	23
4.1 Filtrul global de tip “cosinus-ridicat”. Filtrul de tip “cosinus pătrat”.....	23
4.2 Procedura de simulare a lanțului de transmisiuni în banda de bază.....	25
4.3 Desfășurarea lucrării.....	27

### Lucrarea nr.5

<b>Transmisii de date în banda de bază fără interferență intersimbol, canalul fiind cu zgomot</b> .....	28
5.1 Transmisii de date prin canale cu perturbații.....	28
5.2 Procedura de simulare a lanțului de transmisiuni în banda de bază.....	31
5.3 Desfășurarea lucrării.....	33

**Lucrarea nr.6**

<b>Transmisii de date în banda de bază cu interferență controlată intersimbol</b> .....	34
6.1 Transmisii de date în banda de bază cu interferență controlată intersimbol, canalul fiind fără zgomot.....	34
6.2 Transmisii de date cu interferență controlată intersimbol prin canale cu perturbații.....	36
6.3 Procedura de simulare a lanțului de transmisiuni în banda de bază.....	38
6.4 Desfășurarea lucrării.....	41
<b>Anexe</b> .....	43
<b>Bibliografie</b> .....	55

## Lucrarea nr. 1

### Prezentarea mediului MATLAB

MATLAB (MATrix LABoratory) este un program interactiv, dezvoltat de firma Math Works Inc., fiind destinat, în special, prelucrării numerice a datelor furnizate sub formă vectorială sau matriceală.

MATLAB integrează calculul numeric cu vizualizarea rezultatelor și programarea, într-un mediu flexibil și deschis dezvoltărilor ulterioare.

MATLAB este destinat în mod particular prelucrării numerice (sau digitale) a semnalelor, modelării, simulării și proiectării sistemelor numerice complexe, precum și dezvoltării de algoritmi de prelucrare a seriilor de date, imaginilor sau câmpurilor multidimensionale de date.

Subsistemele de prelucrare (Toolbox-urile) extind funcțiile MATLAB-ului și realizează operațiuni dedicate unor domenii specifice de prelucrare numerică, cum ar fi: prelucrarea imaginilor, optimizarea, statistica, identificarea sau controlul sistemelor, studiul rețelelor neurale, a sistemelor fuzzy etc.

SIMULINK este un mediu interactiv pentru modelarea și simularea sistemelor dinamice liniare, neliniare, în timp continuu, în timp discret sau a unor structuri hibride, compuse din subsisteme analogice și digitale. SIMULINK permite modelarea sistemelor dinamice cu ajutorul unor ecuații matematice, a unor blocuri funcționale predefinite sau a unor blocuri definite de utilizator.

MATLAB este un sistem interactiv, al cărui element de bază este o matrice care nu necesită o dimensionare explicită. Aceasta ajută în rezolvarea a numeroase probleme de calcul tehnic, în special pentru cele bazate pe formalismul matricial (și vectorial), permițând scrierea programelor într-un limbaj scalar.

MATLAB-ul a evoluat în ultimul timp, mai ales prin utilizarea sa în mediul universitar, ca instrument educațional pentru cursuri de matematică, inginerie și știință. În egală măsură, MATLAB-ul se dovedește deosebit de eficient în cercetarea științifică și dezvoltarea tehnologică în mediul industrial.

### 1.1 Principalele comenzi și funcții în MATLAB

#### Comenzi generale

<i>Who</i>	listează variabilele existente în spațiul de lucru
<i>whos</i>	listează variabilele împreună cu informații despre ele
<i>what</i>	listează fișierele dintr-un director cu extensiile <i>.m</i> , <i>.mat</i> , <i>.mex</i>
<i>size</i>	precizează dimensiunile variabilelor (de tip matrice)
<i>length</i>	precizează dimensiunile vectorilor
<i>Help</i>	documentație on-line privind comenzile MATLAB
<i>exit, quit</i>	comenzi de părăsirea programului MATLAB

#### Comenzi de lucru cu discul

<i>dir, chdir, delete, load, save, type</i>	comenzi similare cu cele specifice sistemului de operare DOS
<i>pack</i>	compactează spațiul de lucru

#### Operatori aritmetici

<i>+, -, *, /, ^</i>	operatori aritmetici uzuali
<i>'</i>	operator de transpunere

**Operatori relaționali și logici**

<, <=, >, >	operatori relaționali uzuali
=	operatorul de egalitate
~ =	operatorul de inegalitate
&	operatorul ȘI logic
	operatorul SAU logic
~	operatorul NU logic

**Comenzi generale**

=	operatorul de atribuire
,	separator (de argumente de funcții)
[]	folosite pentru formarea vectorilor și matricelor
()	folosite în expresii aritmetice
:	folosite în indexarea variabilelor
;	indică sfârșit de linie sau suprimarea tipăririi
...	indică continuarea unei comenzi pe linia următoare
%	indică un comentariu

**Variabile speciale și elementare**

<i>ans</i>	numele implicit al unei variabile sau al unui rezultat
<i>eps</i>	precizia relativă în virgulă mobilă
<i>pi</i>	Valoarea
<i>i, j</i>	Valoarea
<i>Inf</i>	simbolul pentru infinit
<i>NaN</i>	Not a Number
<i>flaps</i>	contor al operațiilor în virgulă mobilă
<i>nargin</i>	numărul argumentelor de intrare a funcției
<i>nargout</i>	numărul argumentelor de ieșire a funcției

**Funcții matematice elementare**

<i>abs</i>	valoarea absolută
<i>sqrt</i>	rădăcină pătrată
<i>Real</i>	partea reală a unei variabile complexe
<i>imag</i>	partea imaginară a unei variabile complexe
<i>angle</i>	unghiul (faza) unei variabile complexe
<i>Conj</i>	conjugatul unei variabile complexe
<i>Sign</i>	semnul unei variabile
<i>rem</i>	restul unei împărțiri întregi
<i>exp</i>	funcția exponențială
<i>log</i>	funcția logaritm natural
<i>log10</i>	funcția logaritm zecimal

**Funcții trigonometrice**

<i>sin, cos, tan, cot, sec</i>	funcții trigonometrice uzuale
<i>asin, acos, atan, acot, asec</i>	funcții trigonometrice inverse
<i>sinh, cosh, tanh, coth, sech</i>	funcții hiperbolice
<i>asinh, acosh, atanh, acoth, asech</i>	funcții hiperbolice inverse

**Funcții destinate analizei datelor**

<i>Max</i>	valoarea maximă a unei variabile tablou
<i>min</i>	valoarea minimă
<i>mean</i>	valoarea medie
<i>std</i>	deviația standard
<i>Sum</i>	suma elementelor unei variabile tablou
<i>cumsum</i>	suma cumulată a elementelor unei variabile
<i>cumsqr</i>	suma pătratelor elementelor unei variabile
<i>prod</i>	produsul elementelor unei variabile tablou
<i>cumprod</i>	produsul cumulată al elementelor unei variabile

**Funcții referitoare la polinoame**

<i>conv</i>	multiplică două polinoame
<i>deconv</i>	împarte două polinoame
<i>roots</i>	determină rădăcinile unui polinom
<i>Poly</i>	construiește un polinom din rădăcinile sale
<i>polyval</i>	evaluează un polinom
<i>residue</i>	dezvoltă în fracții parțiale (o fracție rațională)

**Funcții referitoare la (vectori și) matrice**

<i>zeros</i>	crează o matrice cu elemente nule
<i>ones</i>	crează o matrice cu elemente unitare
<i>rand</i>	crează o matrice cu numere aleatoare distribuite uniform
<i>randn</i>	crează o matrice cu numere aleatoare distribuite normal
<i>linspace</i>	crează un vector cu elemente distribuite liniar
<i>logspace</i>	crează un vector cu elemente distribuite logaritmice
<i>det</i>	calculează valoarea unui determinant
<i>norm</i>	calculează norma unui vector sau matrice
<i>inv</i>	calculează matricea inversă

**Funcții de intrare**

<i>input</i>	permite introducerea datelor de la tastatură
<i>ginput</i>	permite preluarea datelor în mod grafic, de la <i>mouse</i> sau cursor

**Comenzi grafice 1D și 2D**

<i>plot</i>	desenează grafic (în planul <i>xy</i> )
<i>subplot</i>	desenează grafic doar într o porțiune a ecranului
<i>bar</i>	reprezintă un grafic cu bare
<i>hist</i>	reprezintă un grafic sub forma unei histograme
<i>polar</i>	reprezintă un grafic în coordonate polare
<i>stairs</i>	reprezintă un grafic în trepte (scara)
<i>Stem</i>	reprezintă grafic o secvență de date (discrete)
<i>semilogx, semilogy</i>	reprezintă grafic în scara semilogaritmică (pe <i>x</i> , și respectiv pe <i>y</i> )
<i>loglog</i>	reprezintă grafic în scara logaritmică pe <i>x</i> și pe <i>y</i>
<i>xlabel, ylabel</i>	marchează axele de coordonate
<i>title</i>	scrie un text (explicativ) pe grafic
<i>Grid</i>	trasează un caroiaj pe un grafic
<i>figure</i>	crează o fereastră grafică
<i>close</i>	închide o figură
<i>clg</i>	șterge graficul de pe ecran
<i>close all</i>	închide toate ferestrele grafice
<i>cla</i>	șterge axele curente
<i>clr</i>	șterge ecranul
<i>Hold on/off</i>	permite/omite afișarea suprapusă a graficelor
<i>axis</i>	crează (și indică scara pe) axe

**Comenzi grafice 3D**

<i>mesh</i>	desenează o suprafață 3D
<i>contour</i>	reprezintă o suprafață prin curbe de nivel

**Comenzi de control al programului**

<i>if</i>	comanda de execuție condiționată
<i>else, elseif</i>	utilizate cu comanda <i>if</i>
<i>end</i>	termină execuția comenzilor <i>for</i> , <i>while</i> și <i>if</i>
<i>for</i>	comanda de repetarea execuției de un număr de ori
<i>while</i>	comanda de repetare condiționată
<i>break</i>	termină execuția unei bucle
<i>return</i>	comanda de întoarcere după apelul unei funcții
<i>pause</i>	comanda de așteptare

## 1.2 Moduri de lucru în MATLAB

Lansarea în execuție a programului MATLAB se face din WINDOWS.

După lansarea în execuție, programul MATLAB intră în “modul de comandă”, afișând prompterul „ >> ” și așteptând introducerea unei comenzi de către utilizator. Executarea unei comenzi este urmată, de obicei, fie de crearea unei variabile în spațiul de lucru, fie de afișarea unui mesaj sau desenarea unui grafic. De exemplu, comanda:

```
>>v=0:10
```

va crea variabila  $v$  și va afișa elementele acesteia pe ecran. Dacă nu se dorește afișarea rezultatelor unei comenzi pe ecran, comanda trebuie urmată de caracterul „ ; ” care nu inhibă însă și desenarea graficelor.

În afara modului de lucru “în linie de comandă”, în MATLAB se pot crea fișiere ce conțin instrucțiuni MATLAB, numite fișiere-M (deoarece au extensia “.m”). Un program MATLAB poate fi scris sub forma fișierelor “*script*” sau a fișierelor “*function*”. Ambele tipuri de fișiere sunt scrise în format ASCII, iar algoritmul care a fost implementat poate fi urmărit cu foarte mare ușurință, dacă se cunosc convențiile și sintaxa MATLAB. Aceste tipuri de fișiere, obligatoriu cu extensia “.m”, permit crearea unor funcții noi care le pot completa pe cele deja existente. Prin această facilitate, MATLAB-ul poate fi extins la aplicații specifice utilizatorului, care are posibilitatea să scrie noi proceduri.

### *Fișiere “script”*

Un fișier “*script*” este un fișier care conține o secvență de comenzi MATLAB. Prin apelarea numelui fișierului, se execută secvența MATLAB conținută în acesta. După execuția completă a unui fișier *script*, variabilele cu care acesta a operat rămân în zona de memorie a aplicației. Aceste fișiere nu permit integrarea în programe mari, realizate pe principiul modularizării. Fișierele *script* sunt folosite pentru rezolvarea unor probleme care cer comenzi succesive atât de lungi, încât ar putea deveni greoaie pentru lucrul în mod iterativ, adică în modul de lucru linie de comandă.

### *Fișiere “funcție”*

Dacă prima linie a fișierului-M conține cuvântul “*function*”, fișierul respectiv este declarat ca fișier funcție. O funcție diferă de un fișier “*script*” prin faptul că poate lucra cu argumente. Variabilele definite și manipulate în interiorul fișierului funcție sunt localizate la nivelul acestuia. Prin urmare, la terminarea execuției unei funcții, în memoria calculatorului nu rămân decât variabilele de ieșire ale acesteia.

Fișierele funcție sunt utilizate pentru extinderea MATLAB-ului, adică pentru crearea unor noi funcții MATLAB. Forma generală a primei linii a unui fișier funcție este:

```
function [param_ieșire] = nume_funcție (param_intrare)
```

unde:

*function* - este cuvânt cheie care declară fișierul ca fișier funcție (obligatoriu);

nume\_funcție - numele funcției, adică numele sub care se salvează fișierul, fără extensie. Nu poate fi identic cu cel al unui fișier-M preexistent.

param\_ieșire - parametri de ieșire trebuie separați cu virgulă și cuprinși între paranteze drepte. Dacă funcția nu are parametri de ieșire, parantezele drepte și semnul egal nu mai au sens.

param\_intrare - parametri de intrare trebuie separați cu virgulă și cuprinși între paranteze rotunde. Dacă funcția nu are parametri de intrare, parantezele rotunde nu mai au sens.



### 1.3 Desfășurarea lucrării

1.) Se va analiza efectul execuției directe a unor comenzi referitoare la operații simple cu matrici. Introducerea (directă) de la tastatură a elementelor unei matrici notată cu  $A$  se face cu:

```
>> A = [2 7 9; 3 1 5; 8 1 2]
A =
```

```
2 7 9
3 1 5
8 1 2
```

Comanda:  $a = A(1:3,1)$  creează un vector (coloană) compus din elementele primei coloane a matricii  $A$ .

Răspunsul va fi:

```
a =
    2
    3
    8
```

Calculul sumei elementelor primei coloane din matricea  $A$  se face cu una din comenzile:

```
>>sum (A(1:3,1))    sau    >>sum (A(:,1))
```

Răspunsul va fi:

```
ans =
    13
```

Calculul transpusei matricii  $A$  rezultă din:

```
>>B = A'
B =
    2    3    8
    7    1    1
    9    5    2
```

Extragerea elementelor diagonalei principale din matricea  $A$  rezultă din:

```
>>diag(A)
ans =
    2
    1
    2
```

Analizați în continuare efectul următoarelor comenzi:  $A'$ ,  $A(1,:)$ ,  $A(:,)$ ,  $flipud(A)$ ,  $fliplr(A)$ ,  $sum(A)$ ,  $sum(A')$ ,  $mean(A)$ ,  $mean(A')$ ,  $sum(A,2)$ ,  $mean(A,2)$ ,  $min(A)$ ,  $max(A')$

2.) Generați un vector  $C$ , prin întreteserea elementelor vectorilor:

```
A=[1 3 5 7] și B=[2 4 6 8];
```

Cu secvența MATLAB:

```
A=[1 3 5 7]; B=[2 4 6 8];
n=length(A)+length(B);
C(1:2:n)=A; C(2:2:n)=B;
```

se obține rezultatul:

```
C = 1 2 3 4 5 6 7 8
```

3.) Se editează o funcție *genexp.m*, care generează o secvență exponențială,  $x[n]=a^n$ , de lungime finită  $N$ :

```
function x=genexp(a,n0,N)

if(N<=0)
    error('GENEXP: Lungimea nu e pozitivă!')
end
nn=n0+[1:N]-1;
x=a.^nn;
plot(x);
end
```

4.) Se va realiza un fișier de comenzi pentru analiza spectrală a unui semnal  $x(t)$ , cu trei componente:

$$x(t) = 4 \sin(2\pi f_1 t) + 2 \sin(2\pi f_2 t) + 6 \sin(2\pi f_3 t)$$

cu  $f_1 = 50\text{Hz}$ ,  $f_2 = 100\text{Hz}$ ,  $f_3 = 350\text{Hz}$ , iar frecvența de eșantionare,  $f_e = 1000\text{Hz}$ .

Se editează fișierul următor:

```
t=0:.001:.099; %momentele de esantionare
fe=1000; %frecventa de esantionare
x=4*sin(2*pi*50*t)+2*sin(2*pi*100*t)+6*sin(2*pi*350*t); %semnalul temporal

Xt=fft(x); %transformata Fourier
N=length(x); %lungimea secventei
Xm=abs(Xt); %determina modulul
X=Xm(1,1:N/2+1)/(N/2);
f=[0:N/2]*fe/N; %frecventele pozitive
subplot(211) %selecteaza fereastra superioara
plot(t,x) %reprezinta semnalul continuu
grid %traseaza retea grid
xlabel('t[s]') %eticheteaza axa x
ylabel('x(t)[V]') %eticheteaza axa y
title('Semnalul temporal') %titlu
subplot(212) %selecteaza fereastra inferioara
stem(f,X); %reprezinta valori discrete
xlabel('f[Hz]')
ylabel('X(f)[V]')
grid
title('Analiza spectrala a unui semnal')
```

5.) Să se descrie și să se reprezinte grafic funcția modul, definită de relația:

$$y(t) = \begin{cases} -t, & \text{dacă } -4 \leq t < 0 \\ t, & \text{dacă } 0 \leq t \leq 4 \end{cases}$$

Se editează fișierul următor:

```
a=-4; b=4; d=.01; c=0;
t=a:d:b;
k=max(size(t));
for i=1:k
    if (t(i)>=a)&(t(i)<c)
        y(i)=abs(t(i));

    end

    if (t(i)>=c)&(t(i)<=b)
        y(i)=t(i);
    end
end
plot(t,y)
title('Funcția modul');
```

## 1.4 Exerciții

1. Fie matricea A:

A=[1 2 3 4 5 3; 1 4 6 7 4 3; 2 4 8 9 4 0; 1 4 7 4 9 5; 2 5 7 4 8 9; 1 4 6 8 6 4];

Să se scrie instrucțiunile care permit selectarea din matricea A a:

- liniei a 2-a;
  - coloanei a 3-a;
  - submatricei formată din liniile 1-2 și coloanele 4-6;
  - submatricei formată din liniile 1,4 și coloanele 2,4,5;
- Să se genereze și să se reprezinte grafic unele secvențe elementare
    - impulsul unitar
    - treaptă unitate
  - Să se reprezinte grafic un semnal aleator, utilizând instrucțiunea *rand*
  - Să se reprezinte grafic un semnal sinusoidal cu frecvența de 5, 10, 15 Hz și apoi să se reprezinte suma lor.
  - Să se reprezinte grafic un semnal dreptunghiular și să se determine spectrul său, utilizând instrucțiunea *repmat*

```
N=60
n=0:N-1;
x1=ones(1,N/2);
x2=zeros(1,N/2);
x=[x1 x2];
b=repmat(x,1,3);
figure(1)
plot(b)
y=abs(fft(x));
figure(2)
stem(y)
```

## Lucrarea nr. 2

### Tehnici de codare digitală a semnalelor în banda de bază

#### 2.1 Introducere

Într-o transmisiune de date, informația transmisă poate fi de origine analogică sau numerică. Un semnal este considerat numeric (digital) dacă el este discretizat în timp și în amplitudine, ceea ce înseamnă că amplitudinea sa poate lua doar anumite valori, care rămân constante pe intervale bine precizate de timp (respectiv pe intervalul corespunzător duratei unui simbol). Pentru semnalele analogice, amplitudinea acestora variază de o manieră continuă în timp.

O informație analogică poate fi convertită în numeric, de exemplu semnalele video sau audio. De asemenea și procesul invers este posibil, respectiv conversia din numeric în analogic.

#### 2.2 Codarea semnalelor în banda de bază

Spunem despre o transmisie de date că se face „în banda de bază” dacă semnalul de date nu suferă nici un fel de deplasare spectrală datorată modulației. Semnalele în banda de bază sunt supuse atenuărilor introduse de către liniile de transmisie, ele trebuind regenerate periodic în cazul transmisiilor pe distanțe lungi.

În general, semnalul binar propriu zis nu este transmis pe linia de comunicație sub forma sa brută, ci se utilizează diverse tehnici de codare a acestuia în prealabil. Motivele care stau la baza acestei codări sunt diverse:

- Recuperarea tactului necesar unei transmisii sincrone este facilitată de către secvențele binare care prezintă tranziții cât mai numeroase între două stări care corespund unor simboluri. Este astfel de dorit evitarea transmiterii unor secvențe de date care să corespundă unor șiruri lungi de 1, respectiv 0.
- Formarea spectrală („*spectrum shaping*”) a semnalului ce se transmite fără a utiliza tehnici de modulare sau filtrare. Acest lucru poate fi important de exemplu în aplicațiile pe liniile telefonice, care introduc atenuări puternice ale semnalului la frecvențe mai mari de 300kHz
- Eliminarea componentei continue din semnal
- Utilizarea eficientă a benzii de frecvență. Se pot transmite date cu un debit mai mare utilizând aceeași bandă de frecvență

#### 2.3 Prezentarea unor tehnici de codare

- **Codarea NRZ (Not Return to Zero)**

Acest tip de codare folosește două nivele de tensiune diferite. Astfel un „1” logic este reprezentat printr-un nivel pozitiv de tensiune (+V), în timp ce unui „0” îi corespunde fie o tensiune nulă (0V)- în varianta unipolară NRZ, fie o tensiune negativă (-V) dacă ne referim la NRZ bipolar.

Codarea NRZ cunoaște câteva variante:

- NRZ-L(Level): echivalent cu NRZ (1 - nivel ridicat, 0 – nivel coborât)
- NRZ-M(Mark): 1- apare o tranziție, 0 – nu apare nici o tranziție
- NRZ-S(Space): 1 – nu apare nici o tranziție, 0 – apare o tranziție

Debitul maxim teoretic care poate fi atins într-o transmisie NRZ este egal cu dublul benzii de frecvență ocupată de către semnal (pot fi transmiși 2 biți/Hertz).

Principalul dezavantaj al codării de tip NRZ îl constituie lipsa tranzițiilor în cazul unor secvențe lungi de biți identici, ceea ce poate duce la pierderea sincronizării la receptor.

- **Codarea NRZI (Not Return to Zero Inverted)**

Utilizări: Fast Ethernet (100 Base Fx), FDDI.

Codarea NRZI produce o tranziție în semnal pentru fiecare „1”, iar „0” este reprezentat prin lipsa tranziției. Se poate observa că transmiterea unui șir lung de „0” poate provoca desincronizări. Eficiența de utilizare a benzii este aceeași ca la NRZ.

- **Codarea bifazică**

Se utilizează trei variante ale acestui tip de codare (BIΦ-L, BIΦ-M, BIΦ-S). Prima dintre ele este cunoscută și sub denumirea de codare Manchester, și va fi prezentată ulterior. În ceea ce privește codarea BIΦ-M, ea presupune apariția unei tranziții la începutul oricărui interval de bit. Dacă bitul este de „1”, atunci o a doua tranziție va apare la mijlocul intervalului de bit. Pentru transmisia unui „0” nu se va mai produce nici un fel de tranziție. Codarea BIΦ-S este exact inversa codării BIΦ-M (tranziție la începutul intervalului de bit, urmată de o altă tranziție la jumătatea acestui interval dacă se transmite „0”, sau fără tranziție dacă se transmite „1”).

- **Codarea Manchester**

Utilizări: Ethernet 10Base5, 10Base2, 10BaseT, 10 BaseFL

Ideea care stă la baza codării Manchester este aceea de a determina o tranziție pentru semnalul emis, tranziție care să apară la mijlocul perioadei de bit. Astfel, un „1” este reprezentat printr-o tranziție de la nivelul +V la nivelul -V, în timp ce unei tranziții de la nivelul -V la nivelul +V îi corespunde un „0”. Este evident că în acest fel se asigură sincronizarea între emițător și receptor, chiar și în cazul transmisiei unor secvențe lungi de „0” sau „1”. Mai mult decât atât, întrucât simbolurile binare sunt reprezentate prin tranziții și nu prin nivele constante (stări) ca la codajul de tip NRZ, scade drastic probabilitatea apariției unor erori cauzate de mediul de transmisie. Un zgomot care afectează semnalul poate modifica nivelele transmise, dar este puțin probabil că el va duce la inversarea tranziției sau la lipsa ei, conducând astfel la erori la recepție.

Dezavantajul codării Manchester constă în faptul că, pentru a transmite cu un anumit debit binar, este nevoie de o bandă de frecvențe disponibilă dublă față de cea pe care am utiliza-o în cazul altor tipuri de codare (de exemplu pentru a transmite cu un debit de 10Mbps avem nevoie de o lățime de bandă de 10MHz). Acest inconvenient face codarea Manchester dificil de utilizat pentru debite ridicate.

- **Codarea Manchester diferențială**

Utilizare: rețelele de tip Token-Ring

La baza codării Manchester diferențiale stă prezența sau absența unei tranziții la începutul intervalului de tact. Astfel, un bit de „1” este reprezentat prin lipsa unei tranziții, în timp ce fiecare bit de „0” este semnatificat prin prezența unei tranziții. Avantajele, respectiv dezavantajele acestui tip de codare sunt în general aceleași ca la codarea Manchester nediferențială.

- **Codarea AMI bipolară (AMI-Alternate Mark Inversion)**

Utilizare: transmisia ADSL (Additional Digital Subscriber Loop)

Principiu: zerourile sunt reprezentate printr-un potențial nul (absența semnalului electric pe linie), în timp ce biții de „1” sunt reprezentați alternativ prin tensiuni pozitive (+V), respectiv negative (-V). În acest tip de codare pot exista intervale lungi de lipsă semnal (pentru secvențe lungi de „0”), lucru care poate duce la pierderea sincronizării.

Există și varianta inversată a acestei codări, anume **codajul pseudoternar**, unde lipsa semnalului simbolizează un bit de „1”, iar „0” este reprezentat alternant prin potențiale pozitive și negative.

Plecând de la codajul AMI, s-au dezvoltat o serie de tehnici de codare care tind să îl înlocuiască pe acesta în sistemele moderne de transmisiuni. Vom trata în continuare unele dintre aceste coduri.

- **Codarea B8ZS (Bipolar with 8 Zeros Substitution)**

Utilizare: standardul T1(transmisie rapidă de voce, date pe fire torsadate sau cablu coaxial)

Idee: Plecând de la codajul AMI bipolar, se înlocuiesc secvențele de 8 zerouri consecutive, cu secvențe în care să apară tranziții pentru a se evita astfel pierderea sincronismului. Astfel:

- dacă impulsul anterior acestei secvențe de „0” este de nivel pozitiv, atunci codul corespunzător este 000+-0-+;
- Dacă impulsul anterior acestei secvențe de 8 zerouri este de nivel negativ, atunci codul corespunzător este 000-+0+-;

Utilizarea acestui tip de codare va produce două violări ale alternanței „+ -“, situație care este improbabil să fie cauzată de către un zgomot.

- **Codarea HDB-3 (High Density Bipolar Order 3)**

Utilizare: standardele E1, E3

Principiu: Se dorește din nou evitarea desincronizărilor ce ar putea apare la secvențe de „0” lungi. Acest inconvenient este combătut astfel: dacă apare un șir de 4 zerouri consecutive, ultimul bit este înlocuit cu o tensiune de aceeași polaritate cu a ultimului bit de „1” introdus (viol de bipolaritate). Această măsură ar putea duce însă la apariția unei componente continue semnificative. De exemplu, șirul 100000000, ar putea fi codat astfel : +000+000+. Pentru a evita asemenea situații, fiecare bit de viol de bipolaritate trebuie ales de semn schimbat față de precedentul. Tot pentru a evita introducerea unei componente continue în semnal trebuie respectate regulile:

- dacă numărul de „1” de după ultimul viol de bipolaritate este par, atunci un grup de 4 zerouri consecutive se înlocuiește cu secvența „+00+” în cazul în care ultimul „1” de dinaintea acestei secvențe a fost codat printr-un nivel negativ, respectiv cu „-00-” în caz contrar;
- dacă numărul de „1” ce urmează ultimului viol de bipolaritate este impar, atunci un grup de 4 zerouri consecutive se înlocuiește cu secvența „000+” în cazul în care ultimul „1” de dinaintea acestei secvențe a fost codat printr-un nivel pozitiv, respectiv cu „000-” în caz contrar.

Un exemplu de codare HDB-3 este ilustrat mai jos:

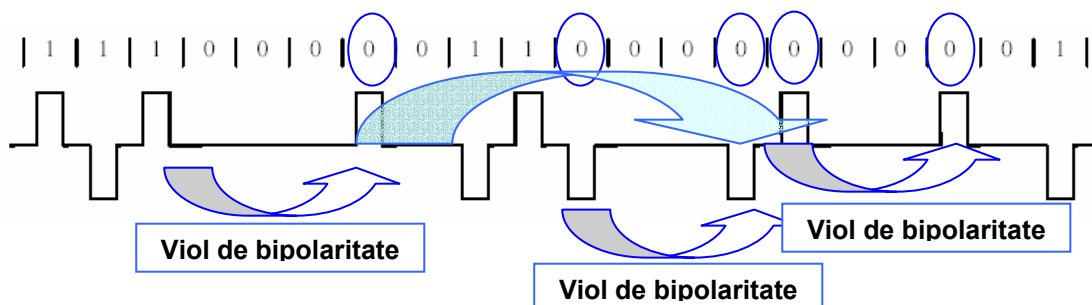


Figura 2.1 Exemplu de codare HDB-3

- **Codarea 4B/5B NRZI**

Utilizare: FDDI, 100Base-X

Această schemă este de fapt o combinație de 2 algoritmi de codare. Pentru a înțelege semnificația acestei alegeri, să considerăm pentru început alternativa simplă a schemei NRZ. Cu NRZ, o stare a semnalului reprezintă „1” binar, și o altă stare reprezintă „0” binar. Dezavantajul clasic al acestei abordări îl reprezintă absența sincronizării. În schema de codare 4B/5B, codarea se face cu 4 biți

odată. Fiecare 4 biți de semnal sunt codăți într-un cuvânt de 5 biți de cod. Eficiența acestei codări este de 80%: pentru a transmite date cu un debit de 100Mbps este necesară o lățime de bandă disponibilă de 125MHz. De remarcat că modul de transmisie al biților cuvântului de cod nu se specifică. Se utilizează de obicei codarea NRZI (ajungându-se la un necesar de bandă de 62.5 MHz) sau codarea MLT-3 (lățime de bandă necesară de 31.25MHz).

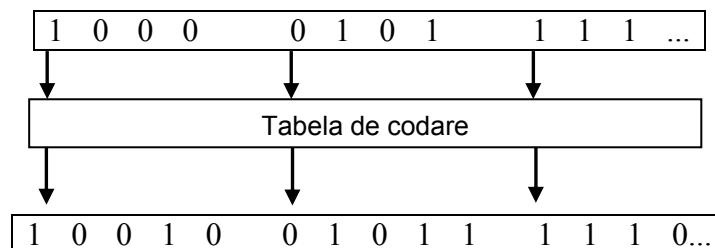
Deoarece pentru a coda cele 16 combinații posibile de câte 4 biți utilizăm doar 16 din cele 32 de combinații posibile de câte 5 biți, rămân alte 16 grupuri care nu sunt utilizate. Cuvintele de cod sunt astfel alese încât să nu existe mai mult de două zerouri succesive și mai puțin de două tranziții (doi de „1”- în codaj NRZI) în cuvântul de cod de 5 biți.

Pentru codare se folosește următorul tabel:

Date	Cod	Date	Cod	Simboluri speciale	Cod
0000	11110	1000	10010	Q (Quiet)	00000
0001	01001	1001	10011	I (Idle)	11111
0010	10100	1010	10110	H (Halt)	00100
0011	10101	1011	10111	J (Start delimiter)	11000
0100	01010	1100	11010	K (Start delimiter)	10001
0101	01011	1101	11011	T (End delimiter)	01101
0110	01110	1110	11100	S (Set)	11001
0111	01111	1111	11101	R (Reset)	00111

**Tabel 2.1 Tabela de codare 4B/5B**

Să considerăm de exemplu secvența binară de intrare 10000101111. Ea este împărțită în grupuri de câte 4 biți, care se codează conform tabelii de codare



**Figura 2.2 Exemplu de codare 4B/5B**

Grupurile de cod care rămân nefolosite sunt fie declarate invalide, fie capătă rolul de simboluri de control a transmisiei, așa cum se observă în tabelul de codare.

- **Codare MLT-3 (MultiLevel Transmission-3)**

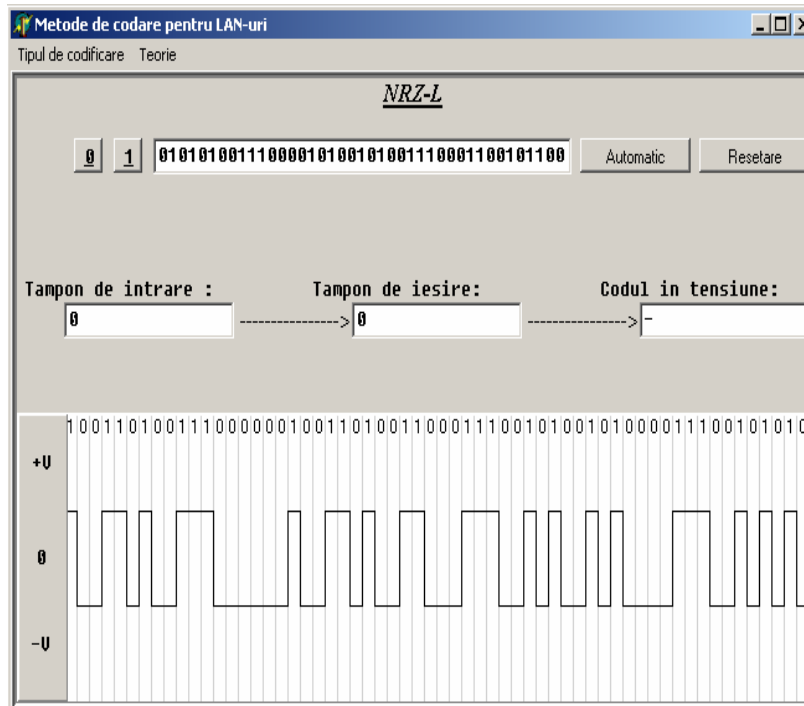
Utilizare: Fast Ethernet 100BaseTx, 100BaseT4

Cu toate că 4B/5B-NRZI este eficace pe fibră optică, el nu poate fi folosit și în transmisiile pe fire torsadate. Cablul UTP acționează asupra semnalului transmis asemănător cu un filtru trece-jos, atenuând componentele de înaltă frecvență și distorsionând astfel puternic secvența de date. Principiul codării MLT este următorul: numai biții de „1” produc o schimbare de stare a semnalului transmis în linie, în timp ce biții de „0” lasă neschimbată starea corespunzătoare bitului anterior. Biții de „1” sunt codăți succesiv prin 3 stări: +V, 0, -V. Marele avantaj al acestui tip de codare este reducerea semnificativă a benzii necesare pentru un debit cerut, grație folosirii celor 3 stări: pentru un debit de 100Mbps lățimea de bandă necesară este de doar 25MHz.

## 2.4 Desfășurarea lucrării

### 2.4.1 Programul de simulare

Pentru simularea modalităților de codare prezentate se va folosi un program de simulare denumit Project1. La accesarea acestui program, se va deschide fereastra principală, prezentată mai jos.



**Figura 2.3 Fereastra principală a programului de simulare**

După cum se poate observa, există în partea de stânga sus a ferestrei butonul „Tipuri de codare” cu ajutorul căruia se poate selecta oricare dintre tipurile de codare prezentate. Este posibilă generarea manuală sau automată a secvenței care va fi codată. Bitul curent generat, modalitatea lui de codare precum și tipul de tensiune corespunzător pot fi de asemenea vizualizate. În partea de jos a ferestrei este redată forma tensiunii trimise în linie conform secvenței binare ce trebuie codată.

### 2.4.2 Sarcini de îndeplinit de către studenți

- Se va citi cu atenție partea teoretică a lucrării, încercându-se înțelegerea principiilor care stau la baza metodelor prezentate.
- Se va genera o secvență de 12 de biți, care va fi codată în mod succesiv utilizând următoarele tipuri de codare : NRZ-L, BIΦ-(L, M, S), Manchester, Manchester diferențial, AMI binar, B8ZS, HDB3 (pentru acesta veti folosi o secvență mai lungă, de 24 de biți, care să conțină cel puțin două subșiruri de patru de „0” consecutivi), 4B/5B NRZI, MLT-3. Se va reprezenta grafic pe hârtie milimetrică evoluția în timp a tensiunii electrice pentru fiecare dintre aceste tipuri de codare.



### Lucrarea nr. 3

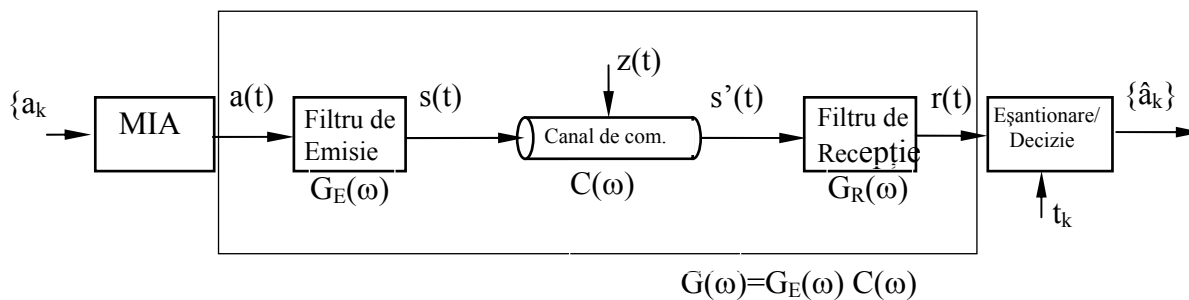
#### Modelarea matematică a sistemelor de transmisiuni în banda de bază.

#### Trecerea unei secvențe de date printr-un filtru global tip “trece-jos ideal”

**Scopul lucrării:** se urmărește familiarizarea studenților cu problematica modelării matematice a unor fenomene și sisteme fizice. Se are în vedere o modelare pe calculator a unui sistem de transmisiuni de date în bandă de bază, modelare ce presupune o prelucrare *numerică*, (în timp discret), în timp ce “modelul” matematic stabilit este o funcție de variabile *continue*.

### 3.1 Modelarea sistemelor de transmisiuni în banda de bază. Teorema lui Nyquist

Prin transmisiune în banda de bază înțelegem acea transmisiune care utilizează porțiunea din apropierea frecvenței zero din banda de frecvență a canalului. Semnalele în banda de bază, impulsuri dreptunghiulare, au un spectru de frecvențe care ocupă, teoretic, o bandă infinită. Însă banda de frecvențe utilizabilă la o transmisie de date este limitată atât datorită canalului cât și din considerente economice. În figura 3.1 se prezintă schema bloc a unui sistem de transmisiuni în banda de bază:



**Figura 3.1** Schema bloc a unui sistem de transmisiuni în banda de bază

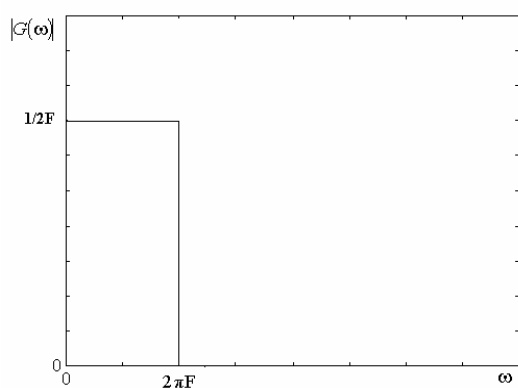
Informația ce trebuie transmisă este conținută în succesiunea de impulsuri modulate în amplitudine,  $a(t)$ , de secvența de date de la intrarea sistemului  $\{a_k\}$ . Amplitudinile posibile sunt  $\pm d, \pm 3d, \dots, \pm(L-1)d$ ,  $L$  fiind 2 pentru semnale binare. La emisie și recepție se utilizează filtrele cu funcțiile de transfer  $G_E(\omega)$  și respectiv  $G_R(\omega)$ . Semnalul recepționat  $r(t)$  este prelucrat de către receptor pentru a se stabili amplitudinea sa și, în consecință, simbolul ce a fost emis.  $C(\omega)$  este funcția de transfer a canalului ideal. Răspunsul în frecvență al întregului lanț de transmisiuni (emițător-canal-receptor),  $G(\omega)$ , este:

$$G(\omega) = G_E(\omega) C(\omega) G_R(\omega) \quad (3.1)$$

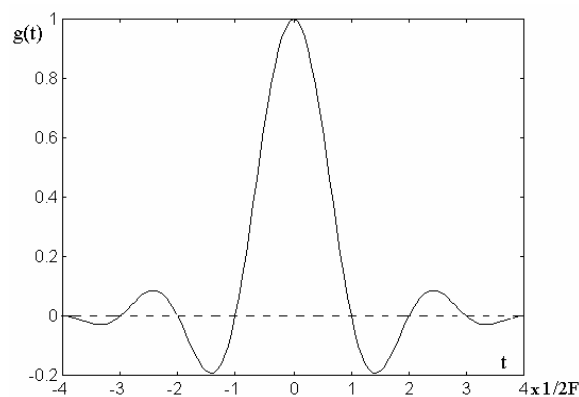
numit și răspunsul în frecvență al canalului echivalent. Deoarece banda de frecvențe utilizabilă la o transmisie de date este limitată, rezultă că acest răspuns în frecvență al canalului echivalent trebuie să fie o funcție de bandă limitată, ce impune însă și o limită maximă a vitezei de transfer a datelor.

**Teorema lui Nyquist** afirmă că într-un canal echivalent cu un *filtru trece-jos ideal*, cu frecvența de tăiere  $F$ , este posibilă transmiterea semnalelor binare independente, cu o viteză de semnalizare mai mică decât  $2F$  simboluri/secundă, fără interferență între simboluri. În figura 3.1a se prezintă

funcția de transfer de tip “*filtru trece-jos ideal*” (cu fază nulă),  $G(\omega)$ , în timp ce figura 3.1b reprezintă răspunsul la impuls,  $g(t)$  corespunzător.



**Figura 3.1a) Răspunsul în frecvență al filtrului trece-jos ideal**



**Figura 3.1b) Răspunsul la impuls corespunzător**

Expresiile analitice ale celor două funcții sunt:

$$G(\omega) = \begin{cases} \frac{1}{2F} & |\omega| \leq \omega_0 = 2\pi F \\ 0 & |\omega| > \omega_0 \end{cases}, g(t) = \frac{\sin(2\pi Ft)}{2\pi Ft} = \text{sinc}(\omega_0 t) \quad (3.2)$$

Considerând o succesiune de impulsuri pe linie, decalate în timp cu durata  $T$ , condiția de interferență nulă între simboluri se poate scrie sub forma:

$$T = k \cdot \frac{1}{2F}, \quad \forall k \in \mathbb{N}^* \quad (3.3)$$

din care rezultă viteza de transmitere maximă:  $v_{max} = \frac{1}{T_{min}} = 2F$ .

Considerând un mesaj  $a(t)$ , ce constă din  $N$  simboluri consecutive, ce apar la intervale de timp  $T$ , acesta are expresia:

$$a(t) = \sum_{n=0}^{N-1} a_n \delta(t - nT) \quad (3.4)$$

Semnalul recepționat,  $r(t)$ , în cazul unui canal ce nu introduce zgomot peste semnalul util ( $z(t)=0$ ), are expresia:

$$r(t) = \sum_{n=0}^{N-1} a_n \cdot g(t - nT) \quad (3.5)$$

Condiția de interferență nulă intersimbol presupune ca eșantionul  $r_k = r(kT)$ , să fie dependent doar de simbolul  $a_k$  și nu și de simbolurile  $a_n$ ,  $n \neq k$ . Notând  $g_k = g(kT)$ , eșantioanele funcției răspuns la impuls  $g(t)$ , condiția de mai sus se verifică pentru:

$$g_{n-k} = 0, \quad \forall n \neq k \quad (3.6)$$

Se observă, analizând relația (3.2), că în condițiile din relația (3.3), se verifică întotdeauna condiția de interferență nulă intersimbol din ecuația (3.6).

## 3.2 Simularea pe calculator a trecerii unei secvențe de date printr-un canal echivalent de tip “trece-jos ideal”

Pentru a putea realiza o simulare a transmisiei cu ajutorul calculatorului, plecând de la filtrul global de tip “trece-jos ideal”, trebuie să găsim un model matematic *discretizat* (în timp discret), echivalent cu modelul în timp continuu stabilit. În acest scop vom utiliza metoda invarianței răspunsului la impuls.

### 3.2.1 Metoda de echivalare a sistemelor în timp continuu cu sisteme în timp discret prin invarianța răspunsului la impuls

Metoda presupune considerarea ca și răspuns la impuls al sistemului discret echivalent secvența  $g_d[n] = g(nT_e)$ , obținută prin eșantionarea răspunsului la impuls  $g(t)$  al sistemului în timp continuu, cu un pas de eșantionare  $T_e$ , ce verifică teorema eșantionării. În aceste condiții Transformata Fourier în Timp Discret, (TFTD), a secvenței  $g_d[n]$ ,  $G_d(\Omega)$ , se exprimă în funcție de răspunsul în frecvență  $G(\omega)$  conform:

$$G_d(\Omega) \Big|_{\Omega=\omega T_e} = \frac{1}{T_e} \sum_{k \in Z} G\left(\omega - k \frac{2\pi}{T_e}\right) \quad (3.7)$$

Ea este o funcție periodică în  $\Omega$ , cu perioada  $2\pi$ .

### 3.2.2 Răspunsul unui sistem în timp discret la o secvență de intrare $x[n]$

Notând cu  $y[n]$ , secvența de la ieșirea unui sistem în timp discret, cu răspunsul la impulsul unitate,  $\delta[n]$ , dat de secvența  $g_d[n]$ , la o secvență de intrare  $x[n]$ , dacă sistemul este liniar și invariant în timp, avem:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot g[n-k] \quad (3.8)$$

ceea ce reprezintă o sumă infinită de termeni. Însă, în general, avem de-a face cu semnale de durată finită, caz în care indicele  $k$  din ecuația (3.8) ia valori dintr-un interval finit.

O cale mai rapidă de calcul a valorilor secvenței  $y[n]$  este oferită de utilizarea algoritmului Transformatei Fourier Rapide (notat TFR sau FFT-Fast Fourier Transform), algoritm ce presupune un număr minim de operații necesare. De fapt, aplicând TFTD, ecuației (2.8) rezultă:

$$Y(\Omega) = G_d(\Omega) \cdot X(\Omega) \quad (3.9)$$

unde  $Y(\Omega)$  și  $X(\Omega)$  reprezintă TFTD a secvențelor  $y[n]$  și respectiv  $x[n]$ . Algoritmul TFR permite calculul rapid a valorilor  $Y(\Omega_k)$  cu  $\Omega_k = k \cdot 2\pi/M$ , unde  $M$  este de obicei o putere a lui 2.

Calculul rapid a acestor valori se face conform:

$$Y(\Omega_k) = G_d(\Omega_k) \cdot X(\Omega_k) \quad (3.10)$$

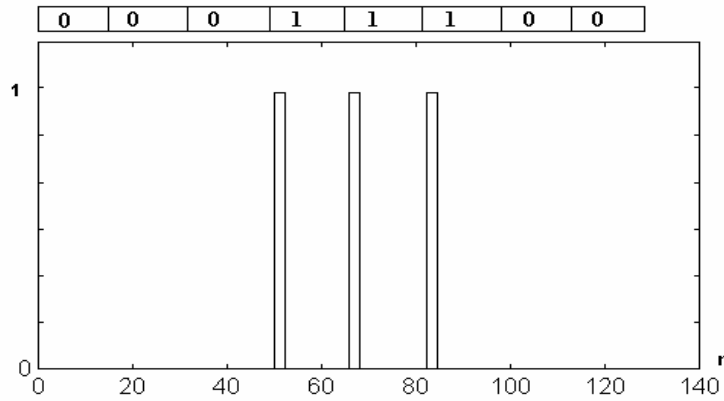
deci utilizând eșantioane ale spectrelor secvențelor  $g_d[n]$  și  $x[n]$ . Aplicând Transformarea Fourier Rapidă Inversă, (TFRI sau IFFT), secvenței de valori spectrale  $Y(\Omega_k)$  se obține secvența răspuns,  $y[n]$ .

### 3.2.3 Generarea cu calculatorul a unei secvențe de date aleatoare

Avem în vedere simularea comportării unui sistem în timp continuu, caracterizat prin funcția răspuns în frecvență și răspunsul la impuls date de ecuația (3.2). Această analiză presupune o secvență de date de intrare de forma dată de ecuația (3.4). Ne vom rezuma doar la o transmisie binară (deci cu doar două nivele posibile ale semnalului de date), unipolară, (nivele normalizate 0 și 1 pentru 0 și respectiv 1 logic), și bipolară, (nivele normalizate -1 și 1 pentru 0 și respectiv 1 logic). În acest scop vom găsi sistemul discret echivalent al sistemului considerat, generând și o versiune discretă a semnalului de intrare  $a(t)$ .

Am văzut în paragraful 3.1., că echivalarea sistemelor are la bază o eșantionare cu o perioadă  $T_e$  a unui semnal în timp continuu. O procedură similară se aplică și semnalelor de intrare/ieșire. Astfel, căutăm semnalul în timp discret  $a_d[n] = a(nT_e)$ , semnal ce se aduce la intrare sistemului discret echivalent. Ținând cont de structura particulară a semnalului de intrare  $a(t)$ , și anume  $N$  simboluri binare de durată  $T$ , alegem perioada  $T_e$  astfel încât să nu rezulte “pierdere sau denaturare de informație” (scăpare de bit sau sondarea diferiților biți în număr diferit de puncte). Acest lucru se asigură alegând raportul  $T/T_e = L_b$  un număr întreg, de obicei o putere întreagă a lui 2.

Considerând un număr  $N$  de biți, tot o putere întreagă a lui 2 rezultă o lungime totală a secvenței numerice,  $M = N \cdot L_b$ , o putere întreagă a lui 2. Se obține o secvență numerică  $a_d[n]$  în care apare o valoare de bit la intervale de timp egale cu  $L_b$ . Dacă această valoare (0 sau 1 în cazul unipolar și -1 sau 1 în cazul bipolar) se generează în mod aleator, (prin ce utilizarea unor funcții ce generează valori aleatoare), obținem o secvență de date de  $N$  simboluri succesive aleatoare.



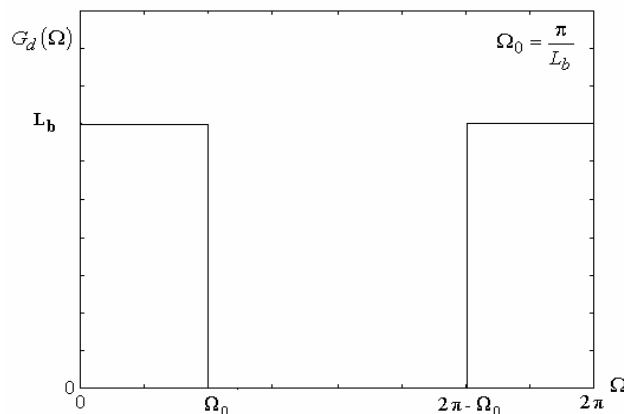
**Figura 3.3 O secvență de impulsuri unipolare**

Un exemplu de o astfel de secvență se prezintă în figura 3.3., unde s-a considerat un număr de  $N=8$  simboluri binare unipolare, cu  $L_b = 2^4 = 16$ , rezultând  $M=128$ , (lungimea totală a secvenței numerice  $a_d[n]$ ).

### 3.2.4 Răspunsul în frecvență al sistemului discret echivalent tip “filtru trece-jos ideal”

Acesta are la bază ecuația (3.7), rezultând o funcție periodică, cu perioada  $2 \cdot \pi$  în variabila  $\Omega$ . Pulsăția de tăiere a filtrului în timp continuu fiind  $\pi/T = 2\pi \cdot F$  (vezi fig. 3.1a), teorema de eșantionare impune verificarea condiției  $T/T_e \geq 1$ . Astfel, se justifică o alegere de forma  $T/T_e = L_b = 2^l$ ,  $l \geq 0$ . Expresia răspunsului în frecvență al sistemului discret echivalent este:

$$G_d(\Omega) = \begin{cases} L_b & |\Omega| \leq \frac{\pi}{L_b} \\ 0 & \frac{\pi}{L_b} < |\Omega| \leq \pi \end{cases}, \quad G_d(\Omega) = G_d(\Omega + 2\pi) \quad (3.11)$$



**Figura 3.4 Răspunsul în frecvență al sistemului discret echivalent**

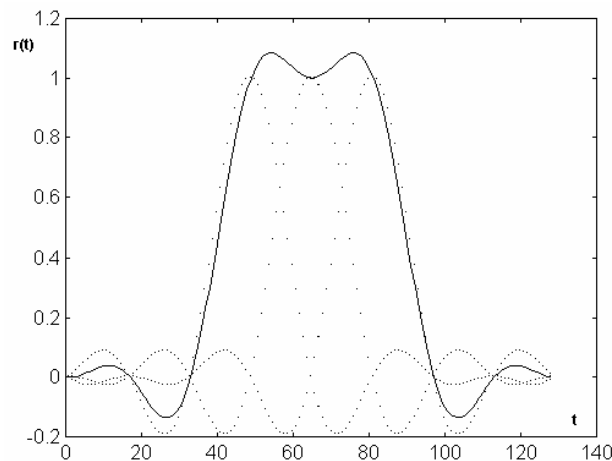
Această funcție, reprezentată în figura 3.4 corespunde unui filtru trece-jos ideal în timp discret, cu pulsația de tăiere  $\Omega_0 = \omega_0 \cdot T_e$ , ce, ținând cont de ecuația (3.3), rezultă a fi:

$$\Omega_0 = 2\pi \cdot F \cdot T_e = \pi \frac{T_e}{T} = \frac{\pi}{L_b} \quad (3.12)$$

De fapt, la simulare se lucrează cu o formă “discretizată” a acestei funcții obținută prin eșantionarea uniformă, în punctele  $\Omega_k = k \cdot \frac{2\pi}{M}$ , a funcției  $G_d(\Omega)$ , rezultând o secvență numerică  $G_d[k] = G_d(\Omega_k)$  de lungime  $M$  (de obicei putere a lui 2).

### 3.2.5 Secvența de la ieșirea sistemului numeric și interpretarea acesteia

La intrarea sistemului numeric echivalent, cu funcția de transfer dată de ec. (3.11), se aduce secvența numerică  $a_d[n]$ , de lungime  $M = N \cdot L_b = 2^m$ , obținută în §3.2.3. Secvența numerică de la ieșirea filtrului trece-jos ideal numeric,  $r_d[n]$ , se obține conform procedurii descrise în secțiunea 3.2.2, utilizând algoritmul FFT (*Fast Fourier Transform* - Transformarea Fourier Rapidă). Această transformată se calculează tot în  $M$  puncte. Secvența  $r_d[n]$  obținută va fi și ea o secvență numerică de lungime  $M$ . Pentru exemplul considerat în secțiunile anterioare, caracterizat prin  $N = 8$  și  $L_b = 16$ , secvența răspuns are forma din figura 3.5.



**Figura 3.5 Răspunsul filtrului numeric la semnalul de intrare din figura 3.3**

Ultima fază a simulării constă în luarea deciziilor asupra biților  $\{a_k\}$  transmiși. Acest lucru se face “eșantionând” semnalul de la ieșirea filtrului, deci secvența  $r_d[n]$ , în momentele de timp

$n_k = k \frac{T}{T_e} = k \cdot L_b$ . Comparând valorile  $r_d[n_k]$  cu o valoare de prag, ( $1/2$  în cazul unei transmisii

binare unipolare și  $0$  în cazul unei transmisii bipolare), se decide asupra simbolurilor  $\{a_k\}$  transmise.

## 3.3 Desfășurarea lucrării

**3.3.1** Utilizând un program de calcul științific (*Mathcad* sau *Matlab*), vizualizați graficul funcției răspuns în frecvență a sistemului în timp discret echivalent,  $G_d(\Omega_k)$  cu  $\Omega_k = k \cdot 2\pi/M$ , în  $M=128$  puncte, și considerând  $L_b = 16$ .

**3.3.2** Aplicând algoritmul TFRI găsiți și reprezentați grafic funcția răspuns la impuls,  $g_d[n]$ , al acestui sistem. Notați pozițiile și valorile maximelor locale ale secvenței. Diferențiind secvența, apreciați viteza de variație a semnalului în jurul trecerilor prin zero. Ce se poate constata ?

**3.3.3** Urmărind procedura descrisă în secțiunea 3.2.3 generați o secvență aleatoare de date unipolare,  $a_d[n]$ , de lungime  $M=128$ , versiunea discretizată a unui semnal  $a(t)$  de  $N=8$  simboluri unipolare.

**3.3.4** Găsiți secvența  $r_d[n]$  ce se obține la ieșirea sistemului numeric dacă la intrarea sa se aduce secvența  $a_d[n]$ . În ce domeniu de valori se încadrează valorile  $r_d[n]$ ? Justificați.

**3.3.5** Stabiliți momentele de timp optime de eșantionare,  $n_k$ , și găsiți secvența de biți obținuți după luarea deciziei,  $\{\hat{a}_k\}$ . Ce se poate constata ? Explicați importanța stabilirii corecte a acestor momente de timp. Ce condiție impune acest lucru sistemului de transmisiuni implementat ?

**3.3.6** Repetați 3.3.3-3.3.5 pentru cazul unei secvențe de date bipolare.

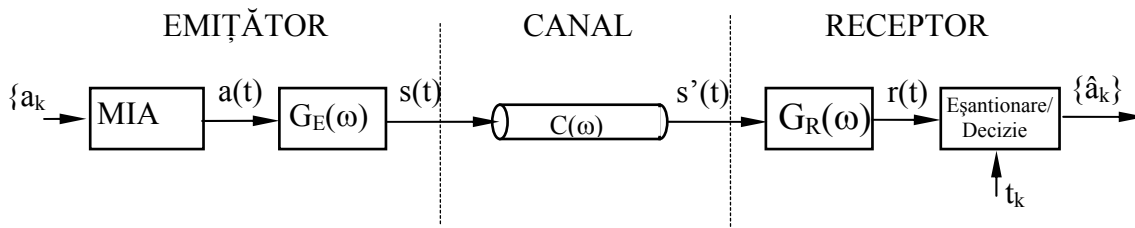
**3.3.7** Creșteți numărul de biți din secvența de intrare, fără modificarea lui  $L_b$ . La ce lungime a secvenței de intrare vom constata o decizie incorectă la recepție?

**Lucrarea nr. 4****Transmisii de date în banda de bază fără interferență intersimbol, canalul fiind fără zgomot**

**Scopul lucrării:** se realizează simularea trecerii unei secvențe de date unipolare și bipolare printr-un canal utilizând ca filtru global un filtru de tip “cosinus pătrat”.

**4.1 Filtrul global de tip “cosinus-ridicat”. Filtrul de tip “cosinus pătrat”**

Structura sistemului de transmisiuni în banda de bază, se prezintă în figura 4.1, unde:



**Figura 4.1 Sistem de transmisiuni de date în banda de bază**

$a(t)$  se consideră a fi un mesaj ce constă din  $N$  simboluri consecutive,  $\{a_k\}$ , ce apar la intervale de timp  $T$ , iar:

$$G(\omega) = G_E(\omega)C(\omega)G_R(\omega) \quad (4.1)$$

reprezintă răspunsul în frecvență al întregului lanț de transmisiuni emițător-canal-receptor. Canalul se consideră ideal,  $C(\omega) = 1$  (cu fază nulă), ce nu perturbă semnalul “injectat” în el. Semnalul recepționat,  $r(t)$ , este eșantionat la momentele de timp  $t_k = kT$ , iar pe baza eșantioanelor obținute  $r_k = r(t_k)$ , se decide asupra simbolului transmis  $a_k$ .

Modelul matematic al unui sistem de transmisiuni de tip “*filtrul trece-jos ideal*” nu este singurul posibil ce permite transmiterea de date fără interferență intersimbol și, în plus, prezintă și dezavantajul major că, de regulă, implică filtre de emisie și/sau recepție ce nu sunt fizic realizabile (deseori funcțiile sistem ce rezultă corespund unor sisteme necauzale). Din acest motiv sistemele de transmisiuni de date în banda de bază ce permit o semnalizare fără interferență intersimbol sunt modelate prin alte caracteristici globale. O familie de astfel de modele este cunoscută sub numele de “*filtre de tip cosinus ridicat*”. Răspunsul în frecvență al sistemului echivalent are expresia:



$$G(\omega) = \begin{cases} \frac{\pi}{\omega_0} = T & |\omega| \leq \omega_0 \cdot (1 - \alpha) \\ \frac{\pi}{2\omega_0} \cdot \left[ 1 - \sin\left(\frac{\pi\omega}{2\alpha\omega_0} - \frac{\pi}{2\alpha}\right) \right] & \omega_0 \cdot (1 - \alpha) \leq |\omega| \leq \omega_0 \cdot (1 + \alpha) \\ 0 & |\omega| > \omega_0 \cdot (1 + \alpha) \end{cases} \quad (4.2)$$

unde  $\alpha$  este un parametru variabil între 0 și 1, denumit și exces de bandă.

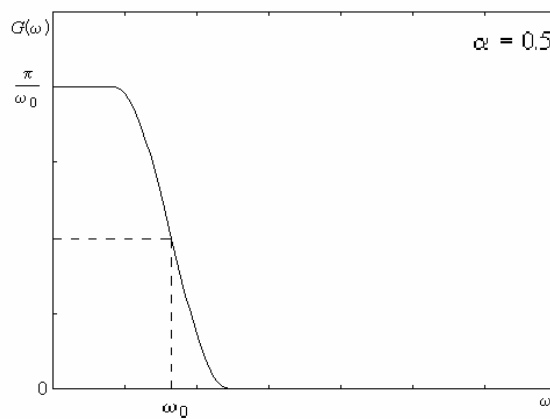
Funcția răspuns la impuls corespunzătoare,  $g(t)$ , este:

$$g(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) e^{j\omega t} d\omega = \frac{\sin \frac{\pi t}{T}}{\frac{\pi t}{T}} \cdot \frac{\cos \frac{\alpha \pi t}{T}}{1 - \left(\frac{2\alpha t}{T}\right)^2} \quad (4.3)$$

ce verifică condiția de interferență nulă intersimbol:

$$g((n - k)T) = 0, \quad \forall n \neq k \quad (4.4)$$

În figura 4.2 se prezintă funcția răspuns în frecvență  $G(\omega)$  pentru  $\alpha = 0.5$ .



**Figura 4.2 Răspunsul în frecvență al unui filtru de tip “cosinus-ridicat”**

Alegând  $\alpha = 1$  se obține funcția de transfer global ce poartă numele de “*filtru de tip cosinus pătrat*”, relația (4.2) devine:

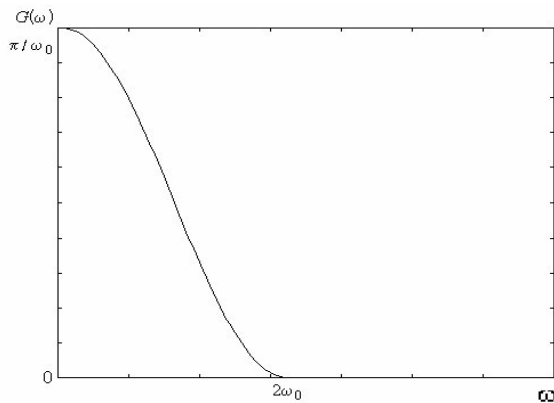
$$G(\omega) = \begin{cases} \frac{\pi}{\omega_0} \cos^2 \frac{\pi\omega}{4\omega_0} & |\omega| \leq 2\omega_0 \\ 0 & |\omega| > 2\omega_0 \end{cases} \quad (4.5)$$

Funcția răspuns la impuls are expresia:

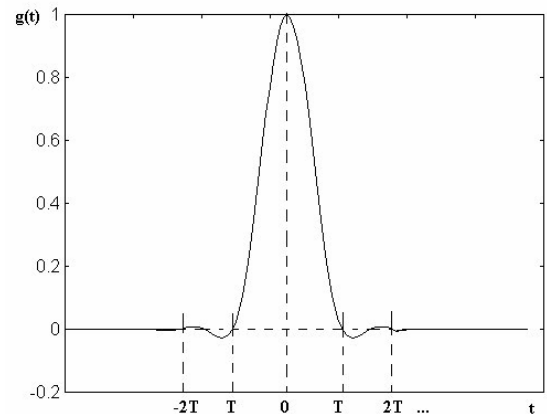
$$g(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) e^{j\omega t} d\omega = \frac{\sin \frac{\pi t}{T}}{\frac{\pi t}{T}} \cdot \frac{\cos \frac{\pi t}{T}}{1 - \left(\frac{2t}{T}\right)^2} \quad (4.6)$$

ce verifică deasemenea condiția de interferență nulă intersimbol, ecuația (4.4).

Figura 4.3a prezintă funcția răspuns în frecvență,  $G(\omega)$  (ecuația (4.5)), în timp ce figura 4.3b prezintă funcția răspuns la impuls corespunzător,  $g(t)$  (ecuația (4.6)).



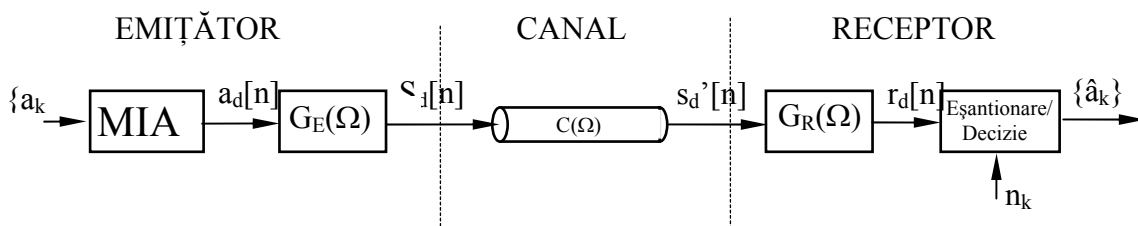
**Figura 4.3a) Răspunsul în frecvență al filtrului "cosinus-pătrat"**



**Figura 4.3b) Răspunsul la impuls corespunzător**

## 4.2 Procedura de simulare a lanțului de transmisiuni în banda de bază

Echivalând sistemul de transmisiuni în timp continuu din figura 3.1 cu un sistem în timp discret și discretizând, deasemenea, semnalele de intrare/ieșire, se obține structura sistemului de analizat din figura 4.4.



**Figura 4.4 Sistemul în timp discret echivalent sistemului din figura 3.1**

Procedura de simulare implică:

**4.2.1** Discretizarea semnalului de intrare  $a(t)$ , prin eșantionare cu pasul  $T_e$ . Se obține astfel o secvență numerică  $a_d[n] = a(nT_e)$ , unde se alege  $T/T_e = L_b = 2^l, l > 0$ , rezultând o lungime totală  $M = N \cdot L_b$  a secvenței. Alegând numărul de simboluri de transmis  $N$  ca fiind o putere întregă a lui 2, va rezultă o lungime totală a secvenței  $M$ , tot o putere întregă a lui 2.

**4.2.2** Echivalarea sistemului în timp continuu cu un sistem în timp discret pe baza metodei de echivalare prin invarianța răspunsului la impuls. Se găsește răspunsul în frecvență al sistemului numeric echivalent,  $G_d(\Omega)$ , funcție ce, în condițiile  $T/T_e = L_b = 2^l, l > 0$  (ce respectă teorema eșantionării), va avea expresia:

$$G_d(\Omega) = \begin{cases} L_b \cdot \cos^2 \frac{L_b \Omega}{4} & |\Omega| \leq \frac{2\pi}{L_b} = \Omega_0 \\ 0 & \Omega_0 < |\Omega| < \pi \end{cases}, \quad G_d(\Omega) = G_d(\Omega + 2\pi) \quad (4.7)$$

Se observă că aceasta este o funcție de o variabilă continuă. În vederea reprezentării pe calculator a acestei funcții ea se discretizează, eșantionând uniform în punctele frecvențiale  $\Omega_k = k \cdot \frac{2\pi}{M}$ , rezultând o secvență numerică  $G_d[k] = G_d(\Omega_k)$  periodică, cu perioada  $M$  (putere întregă a lui 2). Fiind o secvență periodică, memorarea ei implică reținerea doar a  $M$  valori succesive, (o perioadă), ale secvenței.

**4.2.3** Se găsește secvența răspuns,  $r_d[n]$ , de la ieșirea filtrului de recepție. Ținând cont de faptul că, prin ipoteză, canalul nu introduce zgomot peste semnalul util, aceasta se calculează în domeniul Fourier, prin:

- găsirea secvenței  $A_d[k] = FFT(a_d[n])$ , calculată în  $M$  puncte, ce nu reprezintă altceva decât secvența eșantioanelor  $A_d(\Omega_k)$ ,  $\Omega_k = k \cdot \frac{2\pi}{M}$ , ale Transformatei Fourier în Timp Discret (TFTD) a secvenței numerice  $a_d[n]$ ,  $A_d(\Omega)$ ;

- calculul produsului  $R_d[k] = A_d[k] \cdot G_d[k]$ , tot în  $M$  puncte;

- calculul secvenței  $r_d[n] = IFFT(R_d[k])$ .

**4.2.4** Găsirea eșantioanelor  $r_k = r_d[n_k]$ , cu  $n_k = k \cdot L_b$ , ce corespunde eșantionării semnalului în timp continuu,  $r(t)$ , de la ieșirea filtrului de recepție din figura 4.1, în momentele de timp  $t_k = kT$  (de fapt  $n_k = t_k/T_e$ ). Eșantionarea se poate face înmulțind secvența  $r_d[n]$ , cu o secvență de "eșantionare",  $p[n]$ , de forma:

$$p[n] = \begin{cases} 1 & n: L_b \\ 0 & \text{in rest} \end{cases} \quad (4.8)$$

**4.2.5** Luarea deciziilor asupra simbolurilor  $a_k$  transmise pe baza eșantioane-lor  $r_k$ . În acest scop se compară fiecare valoare de eșantion  $r_k$  cu un prag  $d$  ( $1/2$  în cazul unei secvențe unipolare și  $0$  în cazul unei secvențe bipolare), și se decide simbolul:

$$\hat{a}_k = \begin{cases} 1 & \text{daca } r_k \geq d \\ 0 & \text{daca } r_k < d \end{cases} \quad (4.9)$$

### 4.3 Desfășurarea lucrării

**4.3.1** Utilizând un program de calcul științific (*Mathcad* sau *Matlab*), și folosind o funcție de generare de valori aleatoare, ținând cont de recomandările din §4.2.1, generați o secvență aleatoare de date unipolare,  $a_d[n]$ , de lungime  $M=128$ , versiunea discretizată a unui semnal  $a(t)$  de  $N=8$  simboluri unipolare. Cât rezultă  $L_b$ ?

**4.3.2** Găsiți secvența răspuns în frecvență a sistemului în timp discret echivalent,  $G_d[k]$ , în  $M=128$  puncte (vezi §4.2.2).

**4.3.3** Aplicând algoritmul FFT invers, (IFFT), găsiți și reprezentați funcția răspuns la impuls,  $g_d[n]$ , al acestui sistem. Găsiți și pozițiile și valorile maximelor locale ale secvenței. Apreciați, deasemenea, viteza de variație a secvenței în jurul trecerilor prin zero ale semnalului. Cum sunt aceste valori în raport cu cele constatate în cazul “*filtru global tip trece-jos ideal*”?

**4.3.4** Urmărind procedura descrisă în §4.2.3, găsiți secvența  $r_d[n]$  ce se obține la ieșirea sistemului numeric dacă la intrarea să se aduce secvența  $a_d[n]$  generat la punctul §4.2.3. În ce domeniu de valori se încadrează valorile  $r_d[n]$ ? Justificați!

**4.3.5** Stabiliți momentele de timp optime de eșantionare,  $n_k$  și găsiți secvența de biți decși,  $\{\hat{a}_k\}$  conform procedurii descrise în §4.2.4 și §4.2.5. Ce se poate constata? Ce valori au eșantioanele la momentele  $n_k = (k + 1/2) \cdot L_b$  )? Indicați o modalitate de asigurare a sincronizării de bit între emițător și receptor.

**4.3.6** Repetați 4.2.1-4.2.5 pentru cazul unei secvențe de date bipolare!

## Lucrarea nr. 5

### Transmisii de date în banda de bază fără interferență intersimbol, canalul fiind cu zgomot

**Scopul lucrării:** se realizează simularea trecerii unor secvențe de date unipolare și bipolare printr-un canal utilizând ca filtru global un filtru de tip “*cosinus pătrat*”, în condițiile în care canalul introduce zgomot aditiv peste semnalul util.

#### 5.1 Transmisii de date prin canale cu perturbații

Modelarea unui sistem de transmisiuni printr-o caracteristică globală de tip “*trece-jos ideal*” sau “*cosinus-ridicat*” cu idealizarea canalului permitea, din punct de vedere matematic o abordare relativ simplă a sistemului considerat. În realitate, însă, canalul este o componentă esențială a lanțului de transmisiuni, componentă ce în general are un comportament ce afectează semnalul util transmis prin el și, de obicei este variabil în timp. Acest efect asupra semnalului este de obicei un efect dăunător, ce duce la degradarea calității semnalului transmis și îngreunarea procesului de recepție. Semnalul transmis printr-un astfel de canal este, în general, atenuat (datorită pierderilor pe cale), întârziat (datorită vitezei de propagare limitate), distorsionat, (datorită diferitelor viteze de grup cu care se propagă diferitele componente spectrale ale semnalului), și afectat de influența altor semnale, efect ce, de obicei este unul imprevizibil, având un caracter profund aleator, motiv din care le vom numi zgomote (ce se suprapun peste semnalul util).

În figura 5.1 se prezintă structura sistemului de transmisiuni în banda de bază, unde:

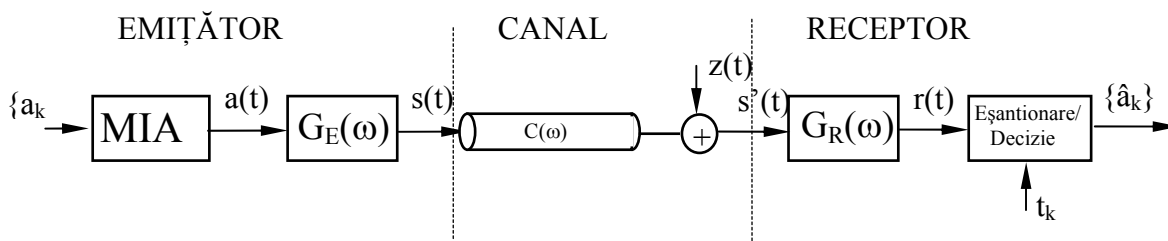


Figura 5.1 Schema de principiu a unui sistem de transmisiuni în banda de bază

$$G(\omega) = G_E(\omega)C(\omega)G_R(\omega) \quad (5.1)$$

reprezintă răspunsul în frecvență al întregului lanț de transmisiuni emițător-canal-receptor, unde în mod normal avem:

$$C(\omega) = A(\omega) \cdot e^{j\varphi(\omega)} \quad (5.2)$$

în care  $A(\omega)$  pune în evidență o atenuare ponderată a diverselor componente spectrale ale semnalului, iar  $\varphi(\omega)$  indică relația de fază ce va exista între aceste componente. O funcție  $A(\omega)$  independentă de frecvență denotă o atenuare uniformă, iar o funcție de fază  $\varphi(\omega)$  liniară denotă un canal ce nu distorsionează forma semnalului. În schema din figura 5.1 influența zgomotelor introduse

de canal se pune în evidență prin însumarea semnalului de la ieșirea canalului,  $s(t)$ , cu un “zgomot”,  $z(t)$ . În cele ce urmează vom neglija efectele de întârziere și de distorsionare ale canalului ținând cont doar de zgomotul introdus de el pe care-l vom considera ca fiind unul aditiv, alb și cu distribuție gaussiană, cu dispersia constantă  $\sigma_z$ . Modelul de sistem este cel de filtru global tip “cosinus pătrat”, cu fază nulă, cu funcția răspuns în frecvență dată de:

$$G(\omega) = \begin{cases} T \cdot \cos^2 \frac{\omega T}{4} & |\omega| \leq \frac{2\pi}{T} \\ 0 & |\omega| > \frac{2\pi}{T} \end{cases} \quad (5.3)$$

Funcția răspuns la impuls,  $g(t)$ , corespunzătoare, verifică condiția de interferență nulă intersimbol:

$$g((n-k)T) = 0, \quad \forall n \neq k \quad (5.4)$$

Considerând un mesaj  $a(t)$ , ce constă din  $N$  simboluri consecutive, ce apar la intervale de timp  $T$ , acesta are expresia:

$$a(t) = \sum_{n=0}^{N-1} a_n \delta(t - nT) \quad (5.5)$$

Notând cu  $g_E(t)$  răspunsul la impuls al filtrului de emisie, la ieșirea acestuia se obține semnalul:

$$s(t) = \sum_{n=0}^{N-1} a_n \cdot g_E(t - nT) \quad (5.6)$$

semnal ce urmează a fi transmis prin canal. Canalul fiind idealizat, deci nu atenuază semnalul, nu-l întârzie sau dispersează, și ține cont doar de zgomotele ce le introduce, la intrarea filtrului de recepție vom avea semnalul:

$$s'(t) = s(t) + z(t) \quad (5.7)$$

relație ce pune în evidență cele două componente ale unui semnal ce trece printr-un canal zgomotos: componenta de semnal util,  $s(t)$ , și componenta de zgomot,  $z(t)$ .

Dacă filtrul de recepție este un sistem liniar atunci și semnalul de la ieșirea acestuia va fi compus din două componente, una de semnal util și cea de zgomot. Astfel, semnalul recepționat,  $r(t)$  se poate scrie:

$$r(t) = r_u(t) + z_0(t) \quad (5.8)$$

unde  $r_u(t)$  reprezintă componenta de semnal util, deci răspunsul filtrului de recepție la semnalul  $s(t)$ , iar  $z_0(t)$  este componenta de zgomot, deci răspunsul filtrului de recepție la zgomotul  $z(t)$  din canal. Componenta de semnal util are expresia:

$$r_u(t) = \sum_{n=0}^{N-1} a_n \cdot g(t - nT) \quad (5.9)$$

Semnalul recepționat  $r(t)$  este eșantionat, iar pe baza eșantionului  $r_k = r(kT)$  se ia decizia asupra simbolului binar  $a_k$  transmis. Semnalul recepționat  $r(t)$  este eșantionat la momentele de timp  $t_k = kT$ , iar pe baza eșantioanelor obținute  $r_k = r(t_k)$ , se decide asupra simbolului transmis  $a_k$ . Cum  $r_{uk} = r_u(kT) = a_k$ , decizia corectă la recepție presupune ca eșantionul de zgomot  $z_{0k} = z_0(kT)$ , în modul, să nu depășească un anumit prag ( $1/2$  în cazul unei transmisii unipolare și  $0$  în cazul unei transmisii bipolare).

Se definește *raportul semnal/zgomot*,  $RSZ$ , la ieșirea filtrului de recepție la momentul  $t_0$  ca fiind:

$$RSZ_0 = \frac{|r_u(t_0)|^2}{P_{z_0}} \quad (5.10)$$

unde  $P_{z_0}$  este puterea zgomotului de la ieșirea filtrului de recepție.

În cazul unui zgomot alb gaussian din canal, de putere  $\sigma_i^2$ , puterea zgomotului la ieșirea filtrului de recepție este:

$$P_{z_0} = \sigma_0^2 = \sigma_i^2 \cdot \frac{1}{2\pi} \int_{-\infty}^{\infty} |G_R(\omega)|^2 d\omega \quad (5.11)$$

Semnalul util recepționat, la momentul  $t_0 = 0$ , se poate scrie:

$$r_u(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \cdot G_R(\omega) d\omega \quad (5.12)$$

Se caută filtrul de recepție optim,  $G_R(\omega)$ , care să maximizeze raportul semnal/zgomot la ieșirea filtrului de recepție la momentul  $t_0 = 0$ . Acest filtru se găsește ca fiind unul cu expresia răspunsului în frecvență:

$$G_R(\omega) = k \cdot S^*(\omega) \quad (5.13)$$

deci, filtrul de recepție este adaptat la forma semnalului din canal.

Necunoscând însă, apriori, informația ce se va transmite prin canal, cerința (5.13) este dificil de realizat. O soluție mai puțin pretențioasă, însă doar suboptimală, acceptată în practică ca și una de compromis, este oferită de condiția de maximizare a raportului semnal/zgomot la ieșirea filtrului de recepție în cazul transmiterii doar a unui singur simbol de informație. Acest lucru presupune:

$$\begin{aligned} a_0 &= 1 \\ a_k &= 0, \quad \forall k \neq 0 \end{aligned} \Rightarrow S(\omega) = G_E(\omega) \quad (5.14)$$

ce, ținând cont de ecuația (5.13), implică:

$$G_R(\omega) = k \cdot G_E(\omega) \Rightarrow G(\omega) = k \cdot |G_E(\omega)|^2 \quad (5.15)$$

Condiție satisfăcută, cu  $k=1$ , de alegerea:

$$|G_R(\omega)| = |G_E(\omega)| = \sqrt{|G(\omega)|} \quad (5.16)$$

Relația (5.16) nu impune nici o restricție deosebită caracteristicii de fază a celor două funcții răspuns în frecvență,  $G_E(\omega)$  respectiv  $G_R(\omega)$ , singura condiție ce trebuie să fie verificată este aceea de sumă nulă a celor două faze la orice frecvență. Pentru o funcție  $G(\omega)$  reală, cum este și cazul nostru, putem alege:

$$G_E(\omega) = G_R(\omega) = \sqrt{G(\omega)} \quad (5.17)$$

## 5.2 Procedura de simulare a lanțului de transmisiuni în banda de bază

Echivalând sistemul de transmisiuni în timp continuu din figura 5.1. cu un sistem în timp discret, și discretizând și semnalele de intrare/ieșire, se obține structura lanțului de transmisiuni echivalent din figură:

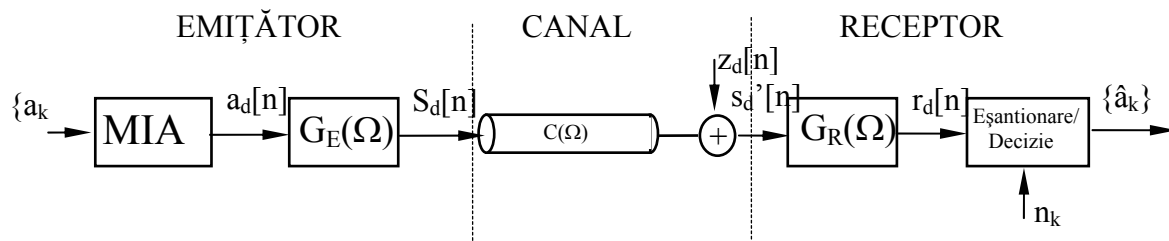


Figura 5.2. Sistemul în timp discret echivalent sistemului din figura 5.1.

Procedura de simulare implică:

**5.2.1.** Discretizarea semnalului de intrare  $a(t)$ , prin eșantionare cu pasul  $T_e$ . Se obține astfel secvența  $a_d[n] = a(nT_e)$ . Alegând  $T_0/T_e = L_b = 2^l, l > 0$ , rezultă o lungime totală  $M = N \cdot L_b$  a secvenței. Alegând  $N = 2^n$ , va rezultă o lungime totală a secvenței,  $M$ , tot o putere întregă a lui 2.

**5.2.2.** Echivalarea filtrelor în timp continuu, dimensionate conform ecuației (5.17), cu filtre în timp discret. Echivalarea se face pe baza metodei de echivalare prin invarianța răspunsului la impuls. Se găsesc funcțiile răspuns în frecvență ale filtrelor numerice echivalente,  $G_{d_E}(\Omega)$  și, respectiv  $G_{d_R}(\Omega)$ , funcții ce, în condițiile  $T_0/T_e = L_b = 2^l, l > 0$  (ce respectă teorema eșantionării), au expresia:



$$G_{d_E}(\Omega) = G_{d_R}(\Omega) = \sqrt{G_d(\Omega)} = \begin{cases} \sqrt{2L_b} \cdot \sqrt{\cos \frac{L_b \Omega}{2}} & |\Omega| \leq \frac{\pi}{L_b} = \Omega_0, \\ 0 & \Omega_0 < |\Omega| < \pi \end{cases} \quad (5.18)$$

$$G_{d_E}(\Omega) = G_{d_E}(\Omega + 2\pi), \quad G_{d_R}(\Omega) = G_{d_R}(\Omega + 2\pi)$$

În vederea reprezentării lor pe calculator funcțiile se discretizează, eșantionând uniform în punctele  $\Omega_k = k \cdot \frac{2\pi}{M}$ , rezultând secvențele numerice periodice  $G_{d_E}[k] = G_{d_E}(\Omega_k)$  și  $G_{d_R}[k] = G_{d_R}(\Omega_k)$ , perioada lor fiind  $M$  (o putere întreagă a lui 2). Memorarea lor implică reținerea a doar  $M$  valori succesive (o perioadă completă) din fiecare.

**5.2.3.** Se găsește secvența  $s_d[n]$ , de la ieșirea filtrului de emisie. Aceasta se calculează în domeniul Fourier, prin:

- găsirea secvenței  $A_d[k] = FFT(a_d[n])$ , calculată în  $M$  puncte, (de fapt se obține secvența eșantioanelor  $A_d[k] = A_d(\Omega_k)$ ,  $\Omega_k = k \cdot \frac{2\pi}{M}$ , ale Transformatei Fourier în Timp Discret a secvenței numerice  $a_d[n]$ ,  $A_d(\Omega)$ );

- calculul produsului  $S_d[k] = A_d[k] \cdot G_d[k]$ , tot în  $M$  puncte;

- calculul secvenței  $s_d[n] = IFFT(S_d[k])$ .

Se obține astfel semnalul ce urmează a fi transmis prin canal, semnal ce trebuie însumat cu zgomotul canalului.

**5.2.4.** Generarea unei secvențe de zgomot,  $z_d[n]$ , de putere  $\sigma_i^2$ , ce trebuie însumată cu secvența de semnal util  $s_d[n]$ , obținând:

$$s'_d[n] = s_d[n] + z_d[n] \quad (5.19)$$

**5.2.5.** Se găsește secvența răspuns  $r_d[n]$  de la ieșirea filtrului de recepție. Procedura utilizată poate fi una similară celei descrise în §5.2.3 (prin calcul în domeniu frecvență).

**5.2.6.** Găsirea eșantioanelor  $r_k = r_d[n_k]$ , cu  $n_k = -L_b/2 + k \cdot L_b$ , ce corespunde eșantionării semnalului în timp continuu,  $r(t)$ , de la ieșirea filtrului de recepție din figura 5.1, în momentele de timp  $t_k = kT$ . În acest scop se înmulțește secvența  $r_d[n]$ , cu o secvență de “eșantionare”,  $p[n]$ , de forma:

$$p[n] = \begin{cases} 1 & (n + L_b/2) : L_b \\ 0 & \text{in rest} \end{cases} \quad (5.20)$$

**5.2.7.** Luarea deciziilor asupra simbolurilor  $a_k$  transmise pe baza eșantioane-lor  $r_k$ . În acest scop se compară fiecare valoare de eșantion  $r_k$  cu un prag  $d$  ( $1/2$  în cazul unei secvențe unipolare și  $0$  în cazul unei secvențe bipolare), și se decide simbolul:

$$\hat{a}_k = \begin{cases} 1 & \text{daca } r_k \geq d \\ 0 & \text{daca } r_k < d \end{cases} \quad (5.21)$$

### 5.3 Desfășurarea lucrării

**5.3.1** Utilizând un program de calcul științific (*Mathcad* sau *Matlab*), și folosind o funcție de generare de valori aleatoare, ținând cont de recomandările din §5.2.1, generați o secvență aleatoare de date unipolare,  $a_d[n]$ , de lungime  $M=128$ , versiunea discretizată a unui semnal  $a(t)$  de  $N=8$  simboluri binare unipolare. Cât rezultă  $L_b$  ?

**5.3.2** Găsiți secvențele răspuns în frecvență ale filtrelor în timp discret de emisie și recepție,  $G_{dE}[k]$ ,  $G_{dR}[k]$ , în  $M=128$  puncte (vezi §5.2.2).

**5.3.3** Aplicând algoritmul FFT invers, (IFFT), găsiți și reprezentați funcția răspuns la impuls,  $g_{dE}[n]$ , a filtrului de emisie.

**5.3.4** Urmărind procedura descrisă în §5.2.3, găsiți secvența  $s_d[n]$  ce se obține la ieșirea filtrului de emisie dacă la intrarea sa se aduce secvența  $a_d[n]$ .

**5.3.5** Utilizând o funcție de generare de valori aleatoare generați o secvență de valori aleatoare,  $z_d[n]$ , de lungime  $M=128$ . Distribuția valorilor este gaussiană, cu o dispersie de  $\sigma_i = 0.5$ . Reprezentați secvența sumă  $s'_d[n] = s_d[n] + z_d[n]$ .

**5.3.6** Urmărind indicațiile din §5.2.5, găsiți secvența  $r_d[n]$  ce se obține la ieșirea filtrului de recepție.

**5.3.7** Stabiliți momentele de timp optime de eșantionare,  $n_k$ , și găsiți secvența de biți obținuți în urma deciziei,  $\{\hat{a}_k\}$ , conform procedurii descrise în §5.2.6 și §5.2.7. Ce se poate constata ?

**5.3.8** Repetați 5.3.1-5.3.7 pentru cazul unui semnal de  $N=512$  simboluri binare aleatoare. Modificând valoarea dispersiei zgomotului, schițați grafic evoluția probabilității deciziei incorecte în funcție raportul semnal/zgomot. Se vor considera valorile  $\sigma_i = 0.75, 1, 1.25, 1.5$ .

**5.3.9** Repetați 5.3.1-5.3.8 pentru cazul unei secvențe de date bipolare.

## Lucrarea nr. 6

### Transmisii de date în banda de bază cu interferență controlată intersimbol

**Scopul lucrării:** se realizează simularea trecerii unei secvențe de date unipolare și bipolare printr-un canal de comunicații utilizând ca filtru global un filtru de tip “cosinus”, în condițiile în care canalul este fără zgomot, iar apoi canalul introduce un zgomot aditiv peste semnalul util

#### 6.1 Transmisii de date în banda de bază cu interferență controlată intersimbol, canalul fiind fără zgomot

Avem în vedere sistemul de transmisii în banda de bază, din figura 6.1:

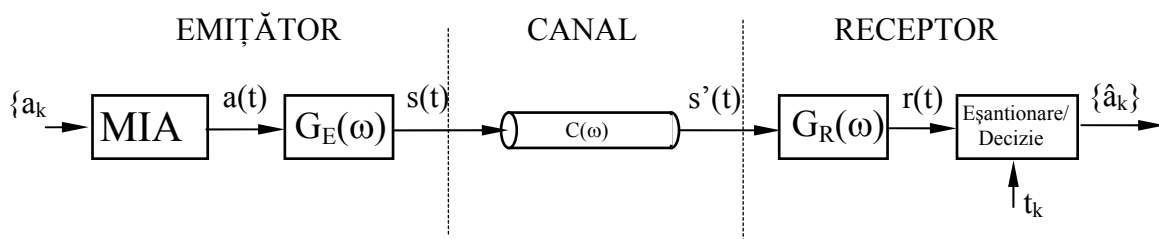


Figura 6.1 Sistem de transmisii de date în banda de bază

unde :

$$G(\omega) = G_E(\omega)C(\omega)G_R(\omega) \quad (6.1)$$

reprezintă răspunsul în frecvență al lanțului de transmisii emițător-canal-receptor (caracteristica globală a sistemului). Canalul se consideră ideal,  $C(\omega) = 1$  (cu fază nulă), ce nu perturbă semnalul transmis prin el.

Modelul “*filtru trece-jos ideal*” al caracteristicii globale  $G(\omega)$  asigură rata de transfer maximă a transmisiei datelor fără interferență intersimbol, în condiția unei limitări a benzii de frecvență (teorema lui Nyquist). Însă modelul este unul ideal ce nu corespunde situațiilor reale. Celelalte modele stabilite, familia de filtre tip “cosinus-ridicat”, permit la aceeași bandă de frecvențe disponibilă, o rată de transfer maximă mai redusă. Se poate arată că, permițând un anumit grad de interferență între simbolurile transmise, se găsesc și alte caracteristici de transfer ce permit rata maximă permisă de teorema lui Nyquist în condiții similare de bandă. Un astfel de model este cunoscut sub denumirea de “*filtru global tip sinus*”. Răspunsul în frecvență al sistemului este dat de relația:

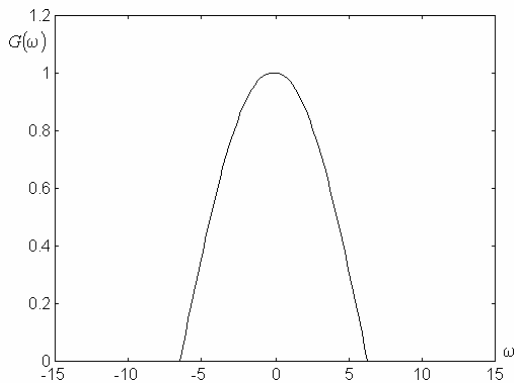
$$G(\omega) = \begin{cases} T \cdot \cos \frac{\omega T}{4} & |\omega| \leq \frac{2\pi}{T} \\ 0 & |\omega| > \frac{2\pi}{T} \end{cases} \quad (6.2)$$

limitat în bandă la pulsația  $2\pi/T$ .

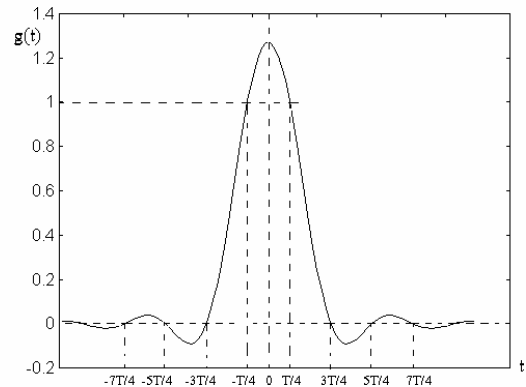
Răspunsul la impuls al filtrului este:

$$g(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) e^{j\omega t} d\omega = \frac{4}{\pi} \cdot \frac{\cos \frac{2\pi t}{T}}{1 - \left(\frac{4t}{T}\right)^2} \quad (6.3)$$

În figura 6.2a se prezintă funcția răspuns în frecvență,  $G(\omega)$ , din ecuația (6.2), cu valoarea  $T=2$ , în timp ce în figura 6.2b se prezintă funcția răspuns la impuls,  $g(t)$ , corespunzător (generat în  $N=128$  puncte).



**Figura 6.2a) Răspunsul în frecvență al filtrului “cosinus”**



**Figura 6.2b) Răspunsul în frecvență al filtrului “cosinus”**

Funcția răspuns la impuls  $g(t)$  verifică:

$$\begin{aligned} g(T/4) &= g(-T/4) = 1 \\ g(-T/4 + k \cdot T/2) &= 0 \quad \forall k \neq 0,1 \end{aligned} \quad (6.4)$$

Această proprietate permite o transmisie de date cu durata  $T_0 = T/2$  a simbolurilor. Astfel, dacă se consideră un semnal  $a(t)$ , de intrare, ca fiind un mesaj ce constă din  $N$  simboluri consecutive,  $\{a_k\}$  ce apar la intervale de timp  $T_0$ , de forma: (6.5)

$$a(t) = \sum_{n=0}^{N-1} a_n \delta(t - nT_0) \quad (6.5)$$

atunci, semnalul recepționat,  $r(t)$  în cazul canalului  $C(\omega)$  ce nu introduce zgomot peste semnalul util, are expresia:

$$r(t) = \sum_{n=0}^{N-1} a_n g(t - nT_0) \quad (6.6)$$

Eșantionând semnalul  $r(t)$  în momentele de timp  $t_k = -\frac{T_0}{2} + kT_0$ , avem:

$$r(t_k) = r_k = \sum_{n=0}^{N-1} a_n \cdot g\left(-\frac{T_0}{2} - (n-k) \cdot T_0\right) = a_k g\left(-\frac{T_0}{2}\right) + a_{k-1} g\left(\frac{T_0}{2}\right) + \sum_{n=0, n \neq k, k-1}^{N-1} a_n \cdot g\left(-\frac{T_0}{2} - (n-k) \cdot T_0\right) = a_k + a_{k-1} \quad (6.7)$$

deci eșantionul prelevat la un moment  $t_k$ ,  $r_k$ , depinde atât de simbolul curent cât și de cel anterior transmis. Într-un sistem binar monopolar de transmisiuni, simbolul  $a_k$  poate fi 0 sau 1, în timp ce un sistem binar polar va emite simboluri  $a_k$  cu nivelele posibile -1 și 1. Valorile posibile ale eșantioanelor  $r_k$ , în cele două cazuri, se prezintă în tabelul 6.1.

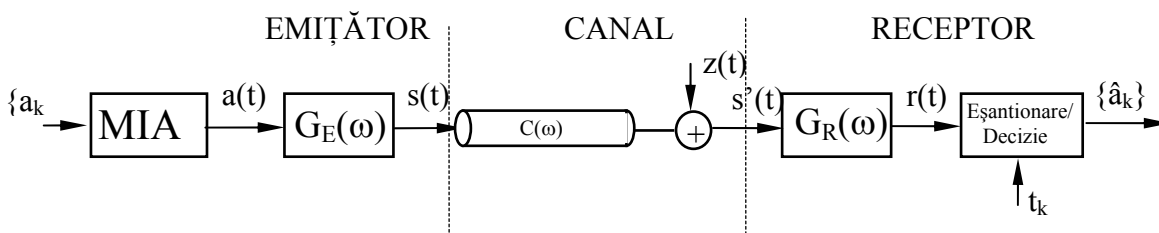
**Tabelul 6.1** Valorile posibile ale eșantioanelor  $r_k$ , în funcție de simbolurile transmise

Sistem unipolar			Sistem bipolar		
$a_k$	$a_{k-1}$	$r_k$	$a_k$	$a_{k-1}$	$r_k$
0	0	0	-1	-1	-2
+1	0	+1	+1	-1	0
0	+1	+1	-1	+1	0
+1	+1	+2	+1	+1	+2

La recepție, decizia asupra simbolului binar transmis,  $a_k$ , notată  $\hat{a}_k$ , se poate face pe baza eșantionului curent  $r_k$  și a deciziei anterioare  $\hat{a}_{k-1}$ . Metoda de detecție utilizată permite deasemenea detectarea unor decizii incorecte. Astfel, de exemplu, în cazul unei transmisiuni bipolare, o valoare  $r_k$  de -2 nu poate fi urmată de o valoare  $r_{k+1} = +2$ . Această facilitate însă nu elimină posibilitatea luării unor decizii incorecte, mai mult, din cauza condiționării deciziei curente de decizia anterioară, apare posibilitatea “propagării” unei decizii incorecte (luări de decizii incorecte până la apariția unei noi erori de decizie sau apariția unei violări de semn).

## 6.2 Transmisiuni de date cu interferență controlată intersimbol prin canale cu perturbații

Modelarea unui sistem de transmisiuni printr-o caracteristică globală de tip “cosinus”, cu idealizarea canalului permitea o analiză a metodei de transmitere cu interferență controlată intersimbol. În realitate însă, trebuie să ținem cont și de influența canalului de comunicații asupra semnalului transmis prin el (atenuare, întârziere, distorsionare, adăugare de zgomot).



**Figura 6.3** Schema de principiu a unui sistem de transmisiuni în banda de bază

În figura 6.3 se prezintă structura unui sistem de transmisiuni în banda de bază, ce modelează un canal zgomotos. În cele ce urmează vom neglija efectele de întârziere și de distorsionare ale canalului, ținând cont doar de zgomotul introdus de el, pe care-l vom considera ca fiind unul aditiv,

alb și cu distribuție gaussiană, cu dispersia constantă  $\sigma_i$ . Astfel, vom considera  $C(\omega) = 1$ . Influența zgomotelor introduse de canal se pune în evidență prin însumarea semnalului de la ieșirea canal,  $s(t)$ , cu un “zgomot”,  $z(t)$ .

Notând cu  $g_E(t)$  răspunsul la impuls al filtrului de emisie, la ieșirea acestuia se obține semnalul:

$$s(t) = \sum_{n=0}^{N-1} a_n g_E(t - nT_0) \quad (6.8)$$

semnal ce este transmis prin canal. Canalul fiind idealizat, deci fiind unul ce nu atenuază și nici nu întârzie sau dispersează, la intrarea filtrului de recepție vom avea semnalul:

$$s'(t) = s(t) + z(t) \quad (6.9)$$

relație ce pune în evidență cele două componente ale unui semnal ce trece printr-un canal zgomotos: componenta de semnal util,  $s(t)$ , și componenta de zgomot,  $z(t)$ .

Filtrul de recepție fiind un sistem liniar rezultă că și semnalul de la ieșirea acestuia se poate descompune în două componente, una de semnal util și cea de zgomot. Astfel, semnalul recepționat,  $r(t)$ , se poate scrie:

$$r(t) = r_u(t) + z_0(t) \quad (6.10)$$

unde  $r_u(t)$  reprezintă componenta de semnal util, deci răspunsul filtrului de recepție la semnalul  $s(t)$ , iar  $z_0(t)$  este componenta de zgomot, deci răspunsul filtrului de recepție la zgomotul  $z(t)$  din canal.

Eșantionând semnalul  $r(t)$  în momentele de timp  $t_k = -\frac{T_0}{2} + kT_0$ , avem:

$$\begin{aligned} r(t_k) &= r_u(t_k) + z_0(t_k) = r_k = r_{uk} + z_k = \\ &= \sum_{n=0}^{N-1} a_n \cdot g\left(-\frac{T_0}{2} - (n-k) \cdot T_0\right) + z_k = a_k g\left(-\frac{T_0}{2}\right) + a_{k-1} g\left(\frac{T_0}{2}\right) + \\ &+ \sum_{n=0, n \neq k, k-1}^{N-1} a_n \cdot g\left(-\frac{T_0}{2} - (n-k) \cdot T_0\right) + z_k = a_k + a_{k-1} + z_k \end{aligned} \quad (6.11)$$

La recepție, decizia asupra simbolului binar transmis,  $a_k$ , notat  $\hat{a}_k$ , se poate face pe baza eșantionului curent  $r_k$  și deciziei anterioare  $\hat{a}_{k-1}$ . Această condiționare a deciziei curente de decizia anterioară face posibilă “propagarea” unei decizii incorecte (ce se manifestă prin luări de decizii incorecte până la apariția unei erori noi). În practică această posibilitate se elimină prin precodarea secvenței de date ce urmează a fi transmis. Astfel, la emisie, în loc de secvența  $\{a_k\}$  se transmite secvența  $\{b_k\}$  dată de:

$$b_k = a_k \oplus b_{k-1} \quad (6.12)$$

Se definește raportul semnal/zgomot,  $RSZ$ , la ieșirea filtrului de recepție la momentul  $t_0$  ca fiind:

$$RSZ_0 = \frac{|r_u(t_0)|^2}{P_{z_0}} \quad (6.13)$$

unde  $P_{z_0}$  este puterea zgomotului de la ieșirea filtrului de recepție.

Se caută filtrul de recepție optim,  $G_R(\omega)$ , care să maximizeze raportul semnal/zgomot la ieșirea filtrului de recepție la momentul  $t_0 = 0$ . Acest filtru se găsește ca fiind unul cu expresia răspunsului în frecvență:

$$G_R(\omega) = k \cdot S^*(\omega) \quad (6.14)$$

deci, filtrul adaptat la forma semnalului din canal.

Necunoscând însă, apriori, informația ce se va transmite prin canal, cerința (6.14) este dificil de realizat. O soluție mai puțin pretențioasă, însă doar suboptimală, acceptată în practică ca și una de compromis, este oferită de condiția de maximizare a raportului semnal/zgomot la ieșirea filtrului de recepție în cazul transmiterii doar a unui singur simbol de informație. Acest lucru conduce la o expresie forma:

$$G_R(\omega) = k \cdot G_E(\omega) \Rightarrow G(\omega) = k \cdot |G_E(\omega)|^2 \quad (6.15)$$

În cazul în care  $k=1$ , rezultă:

$$|G_R(\omega)| = |G_E(\omega)| = \sqrt{|G(\omega)|} \quad (6.16)$$

Relația (6.16) nu impune nici o restricție deosebită caracteristicii de fază a celor două funcții răspuns în frecvență,  $G_E(\omega)$  respectiv  $G_R(\omega)$ , singura condiție ce trebuie să fie verificată este aceea de sumă nulă a celor două faze la orice frecvență. Pentru o funcție  $G(\omega)$  reală, cum este și în cazul nostru, putem alege:

$$G_E(\omega) = G_R(\omega) = \sqrt{G(\omega)} \quad (6.17)$$

### 6.3 Procedura de simulare a lanțului de transmisiuni în banda de bază

Echivalând sistemele de transmisiuni în timp continuu din figurile 6.1 și 6.3 cu un sistem în timp discret și discretizând, deasemenea, semnalele de intrare/ieșire, se obține structura sistemului de analizat din figură:

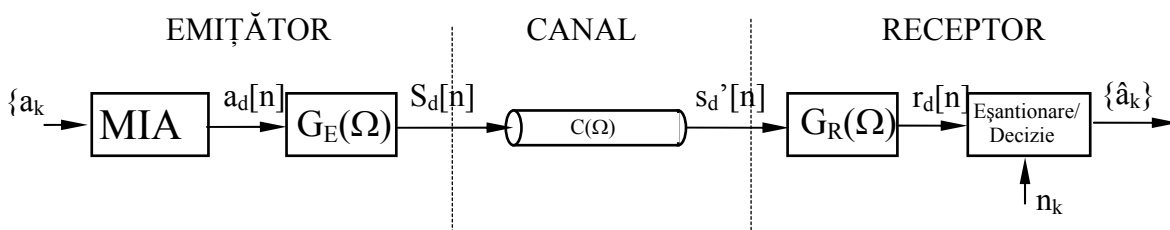
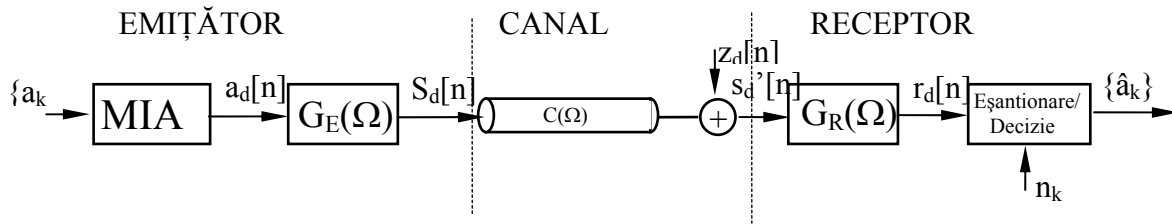


Figura 6.4 Sistemul în timp discret echivalent sistemului din figura 6.1

și respectiv:



**Figura 6.5** Sistemul în timp discret echivalent sistemului din figura 6.3

Procedura de simulare în cazul canalului fără zgomot implică:

**6.3.1** Discretizarea semnalului de intrare  $a(t)$ , prin eșantionare cu pasul  $T_e$ . Se obține astfel o secvență numerică  $a_d[n] = a(nT_e)$ , unde se alege  $T_0/T_e = L_b = 2^l, l > 0$ , rezultând o lungime totală  $M = N \cdot L_b$  a secvenței. Alegând numărul de simboluri de transmis  $N$  ca fiind o putere întreagă a lui 2, va rezulta o lungime totală a secvenței,  $M$ , tot o putere întreagă a lui 2.

**6.3.2** Echivalarea sistemului în timp continuu cu un sistem în timp discret pe baza metodei de echivalare prin invarianța răspunsului la impuls. Se găsește funcția răspuns în frecvență a sistemului numeric echivalent,  $G_d(\Omega)$ , funcție ce, în condițiile  $T_0/T_e = L_b = 2^l, l > 0$  (ce respectă teorema eșantionării), va avea expresia:

$$G_d(\Omega) = \begin{cases} 2L_b \cdot \cos \frac{L_b \Omega}{2} & |\Omega| \leq \frac{\pi}{L_b} = \Omega_0 \\ 0 & \Omega_0 < |\Omega| < \pi \end{cases}, \quad G_d(\Omega) = G_d(\Omega + 2\pi) \quad (6.18)$$

În vederea reprezentării pe calculator a acestei funcții ea se discretizează, eșantionând uniform în punctele  $\Omega_k = k \cdot 2\pi/M$ , rezultând o secvență numerică  $G_d[k] = G_d(\Omega_k)$  periodică, cu perioada  $M$  (o putere întreagă a lui 2). Secvența se memorează reținând  $M$  valori succesive, (o perioadă) ale ei.

**6.3.3** Se găsește secvența răspuns,  $r_d[n]$ , de la ieșirea filtrului de recepție. Ținând cont de faptul că, prin ipoteză, canalul nu introduce zgomot peste semnalul util, aceasta se calculează în domeniul frecvență, prin:

- găsirea secvenței  $A_d[k] = FFT(a_d[n])$ , calculată în  $M$  puncte;
- calculul produsului  $R_d[k] = A_d[k] \cdot G_d[k]$ , tot în  $M$  puncte;
- calculul secvenței  $r_d[n] = IFFT(R_d[k])$ .

**6.3.4** Găsirea eșantioanelor  $r_k = r_d[n_k]$ , cu  $n_k = -\frac{L_b}{2} + k \cdot L_b$ , ce corespund eșantionării semnalului în timp continuu,  $r(t)$ , de la ieșirea filtrului de recepție din figura 6.1, în momentele de timp

$t_k = -\frac{T_0}{2} + kT_0$ . Eșantionarea se poate face înmulțind secvența  $r_d[n]$ , cu o secvență de

“eșantionare”,  $p[n]$ , de forma:



$$p[n] = \begin{cases} 1 & (n + L_b/2) : L_b \\ 0 & \text{in rest} \end{cases} \quad (6.19)$$

**6.3.5** Luarea deciziilor asupra simbolurilor  $a_k$  transmise pe baza eșantioanelor  $r_k$ . În acest scop se scade, din eșantionul curent  $r_k$ , nivelul corespunzător deciziei anterioare,  $\hat{a}_{k-1}$ , iar rezultatul,  $r'_k$ , se compară cu un prag  $d$  ( $1/2$  în cazul unei secvențe unipolare și  $0$  în cazul unei secvențe bipolare), și se decide simbolul:

$$\hat{a}_k = \begin{cases} 1 & \text{daca } r'_k \geq d \\ 0 & \text{daca } r'_k < d \end{cases} \quad (6.20)$$

Procedura de simulare în cazul canalului cu perturbații implică:

**6.3.6** Discretizarea semnalului de intrare  $a(t)$ , prin eșantionare cu pasul  $T_e$ . Se obține secvența numerică  $a_d[n] = a(nT_e)$ . Alegând  $T_0/T_e = L_b = 2^l, l > 0$ , rezultă o lungime totală  $M = N \cdot L_b$  a secvenței. Alegând  $N = 2^n$ , va rezulta o lungime totală a secvenței,  $M$ , tot o putere întreagă a lui 2.

**6.3.7** Echivalarea filtrelor în timp continuu, dimensionate conform ecuației (6.17), cu filtre în timp discret. Echivalarea se face pe baza metodei de echivalare prin invarianța răspunsului la impuls. Se găsesc funcțiile răspuns în frecvență ale filtrelor numerice echivalente,  $G_{d_E}(\Omega)$  și, respectiv  $G_{d_R}(\Omega)$ , funcții ce, în condițiile  $T_0/T_e = L_b = 2^l, l > 0$ , (ce respectă teorema eșantionării), au expresia:

$$G_{d_E}(\Omega) = G_{d_R}(\Omega) = \sqrt{G_d(\Omega)} = \begin{cases} \sqrt{2L_b} \cdot \sqrt{\cos \frac{L_b \Omega}{2}} & |\Omega| \leq \frac{\pi}{L_b} = \Omega_0 \\ 0 & \Omega_0 < |\Omega| < \pi \end{cases} \quad (6.21)$$

$$G_{d_E}(\Omega) = G_{d_E}(\Omega + 2\pi), \quad G_{d_R}(\Omega) = G_{d_R}(\Omega + 2\pi)$$

În vederea reprezentării lor pe calculator funcțiile se discretizează, eșantionând uniform în punctele  $\Omega_k = k \cdot \frac{2\pi}{M}$ , rezultând secvențele numerice periodice  $G_{d_E}[k] = G_{d_E}(\Omega_k)$  și  $G_{d_R}[k] = G_{d_R}(\Omega_k)$ , perioada lor fiind  $M$  (o putere întreagă a lui 2). Memorarea lor implică reținerea a doar  $M$  valori succesive, (o perioadă completă), din fiecare.

**6.3.8** Se găsește secvența  $s_d[n]$ , de la ieșirea filtrului de emisie. Aceasta se calculează în domeniul Fourier, prin:

- găsirea secvenței  $A_d[k] = FFT(a_d[n])$ , calculată în  $M$  puncte, (de fapt se obține secvența eșantioanelor  $A_d[k] = A_d(\Omega_k)$ ,  $\Omega_k = k \cdot \frac{2\pi}{M}$ , ale Transformatei Fourier în Timp Discret a secvenței numerice  $a_d[n]$ ,  $A_d(\Omega)$ ;

- calculul produsului  $S_d[k] = A_d[k] \cdot G_d[k]$ , tot în  $M$  puncte;
- calculul secvenței  $s_d[n] = IFFT(S_d[k])$ .

Se obține astfel semnalul de la intrare în canal, ce se însumează cu zgomotul din acesta.

**6.3.9** Generarea unei secvențe de zgomot,  $z_d[n]$ , de putere  $\sigma_i^2$ , ce se însumează cu secvența de semnal util  $s_d[n]$ , obținând:

$$s'_d[n] = s_d[n] + z_d[n] \quad (6.22)$$

**6.3.10** Se găsește secvența răspuns  $r_d[n]$  de la ieșirea filtrului de recepție. Procedura utilizată poate fi una similară celei descrise în §6.3.8 (prin calcul în domeniu frecvență).

**6.3.11** Găsirea eșantioanelor  $r_k = r_d[n_k]$ , cu  $n_k = -L_b/2 + k \cdot L_b$ , ce corespunde eșantionării semnalului în timp continuu,  $r(t)$ , de la ieșirea filtrului de recepție din figura 6.3, în momentele de timp  $t_k = -\frac{T_0}{2} + kT_0$ . În acest scop se înmulțește secvența  $r_d[n]$ , cu o secvență de “eșantionare”,  $p[n]$ , de forma:

$$p[n] = \begin{cases} 1 & (n + L_b/2) : L_b \\ 0 & \text{in rest} \end{cases} \quad (6.23)$$

**6.3.12** Luarea deciziilor asupra simbolurilor  $a_k$  transmise pe baza eșantioanelor  $r_k$ . În acest scop se scade, din eșantionul curent  $r_k$ , nivelul corespunzător deciziei anterioare,  $\hat{a}_{k-1}$ , iar rezultatul,  $r'_k$ , se compară cu un prag  $d$  ( $1/2$  în cazul unei secvențe unipolare și  $0$  în cazul unei secvențe bipolare), și se decide simbolul:

$$\hat{a}_k = \begin{cases} 1 & \text{daca } r'_k \geq d \\ 0 & \text{daca } r'_k < d \end{cases} \quad (6.24)$$

## 6.4 Desfășurarea lucrării

**6.4.1** Utilizând un program de calcul științific (*Mathcad* sau *Matlab*) găsiți secvența răspuns în frecvență a sistemului în timp discret echivalent,  $G_d[k]$ , în  $M=128$  puncte cu  $L_b=16$  (vezi §5.2.2).

**6.4.2** Aplicând algoritmul FFT invers, (IFFT), găsiți și reprezentați funcția răspuns la impuls,  $g_d[n]$ , al acestui sistem. Găsiți pozițiile în care secvența ia valoarea  $1$ . Cât rezultă intervalul dintre aceste poziții?

**6.4.3** Utilizând o funcție de generare de valori aleatoare și ținând cont de recomandările din §6.3.1, generați o secvență aleatoare de date unipolare,  $a_d[n]$ , de lungime  $M=128$ , versiunea discretizată a unui semnal  $a(t)$  de  $N = M/L_b = 8$  simboluri unipolare.

**6.4.4** Urmărind procedura descrisă în §6.3.3, găsiți secvența  $r_d[n]$  ce se obține la ieșirea sistemului numeric dacă la intrarea sa se aduce secvența  $a_d[n]$ . În ce domeniu de valori se încadrează valorile  $r_d[n]$ ? Justificați.

**6.4.5** Stabiliți momentele de timp optime de eșantionare,  $n_k$ , și găsiți secvența de biți obținuți în urma deciziei,  $\{\hat{a}_k\}$ , conform procedurii descrise în §6.3.4 și §6.3.5. Ce se poate constata? Ce valori au eșantioanele la momentele  $n_k = k \cdot L_b$ ?

**6.4.6** Repetați 6.4.3-6.4.5 pentru cazul unei secvențe de date bipolare.

**6.4.7** Utilizând un program de calcul științific (*Mathcad* sau *Matlab*), și folosind o funcție de generare de valori aleatoare, ținând cont de recomandările din §6.2.1, generați o secvență aleatoare de date unipolare,  $a_d[n]$ , de lungime  $M=128$ , versiunea discretizată a unui semnal  $a(t)$  de  $N=8$  simboluri unipolare. Cât rezultă  $L_b$ ?

**6.4.8** Găsiți secvențele răspuns în frecvență ale filtrelor în timp discret de emisie și recepție,  $G_{dE}[k]$ ,  $G_{dR}[k]$ , în  $M=128$  puncte (vezi §6.3.7).

**6.4.9** Aplicând algoritmul FFT invers, (IFFT), găsiți și reprezentați funcția răspuns la impuls,  $g_{dE}[n]$ , al filtrului de emisie.

**6.4.10** Urmărind procedura descrisă în §6.3.8, găsiți secvența  $s_d[n]$  ce se obține la ieșirea filtrului de emisie dacă la intrarea sa se aduce secvența  $a_d[n]$ .

**6.4.11** Utilizând o funcție de generare de valori aleatoare, generați o secvență de valori aleatoare,  $z_d[n]$ , de lungime  $M=128$ . Distribuția valorilor este gaussiană, cu o dispersie de  $\sigma_i = 0.5$ . Reprezentați secvența sumă  $s'_d[n] = s_d[n] + z_d[n]$ .

**6.4.12** Urmărind indicațiile din §6.3.9 găsiți secvența  $r_d[n]$  ce se obține la ieșirea filtrului de recepție.

**6.4.13** Stabiliți momentele de timp optime de eșantionare,  $n_k$ , și găsiți secvența de biți decizi,  $\{\hat{a}_k\}$ , conform procedurii descrise în §6.3.10 și §6.3.11. Ce se poate constata? Dar dacă creștem valoarea dispersiei zgomotului la  $\sigma_i = 1$ ?

**6.4.14** Repetați 6.3.6-6.3.12 pentru cazul unei secvențe de date bipolare.

**6.4.15** Cum se modifică strategia de decizie dacă se aplică precodarea, conform ecuației (6.20), înainte de transmiterea datelor? Ce avantaje prezintă acest lucru? Repetați 6.4.7-6.4.14 folosind precodarea la emisie.

**Anexe****Lucrarea 3****Trecerea unei secvențe de date unipolare printr-un filtru de tip sinus cardinal (filtru trece-jos ideal)**

```

clear;
t=0:1/32:8-1/32;
M=length(t);

%parametrii programului
Te=1/32;
T=1;
Lb=T/Te;
Omega0=pi/Lb;
N=M/Lb;
punct_taiere=(M*Omega0)/(2*pi);

%generarea unei secvențe aleatoare unipolare de N=8 simboluri și supraesantionarea ei cu Lb=32
simbol=[];
for i=1:N,
    sursa(i)=round(rand);
    simbol=[simbol sursa(i) zeros(1,31)];
end
Simbol=fft(simbol,256);

%definirea caracteristicii de frecvență a filtrului trece-jos ideal
G=[Lb.*ones(1,punct_taiere) zeros(1,(M-2*punct_taiere)) Lb.*ones(1,punct_taiere)];

frecv=(0:2*(M*Omega0)/(2*pi)-1);

%calculul răspunsului la impuls al filtrului trece-jos ideal
g=ifft(G,256);
g=[g(129:256) g(1:128)];

%calculul secvenței de date după trecerea prin filtru
Simbol_rec=Simbol.*G;
simbol_rec=ifft(Simbol_rec);

%detectia datelor de la ieșirea filtrului prin compararea cu pragul
simbol_estimat=round(abs(simbol_rec(1:Lb:M)));
simbol_detectat=[];
for i=1:N,
    simbol_detectat=[simbol_detectat simbol_estimat(i) zeros(1,31)];
end

%se calculează cum răspunde filtrul la fiecare dintre bitii de date de la intrare
separat{1}=[simbol(1) zeros(1,255)];
Separat{1}=fft(separat{1},256);
Out{1}=Separat{1}.*G;
out{1}=real(ifft(Out{1}));
for i=2:N,

```

```

    index1=(i-1)*Lb;
    index2=(i-1)*Lb+1;
    separat{i}=[zeros(1,index1) simbol(index2) zeros(1,M-index2)];
    Separat{i}=fft(separat{i},256);
    Out{i}=Separat{i}.*G;
    out{i}=real(iff(Out{i}));
end

figure(1);
plot(simbol);
title('Simboluri de date emise si supraesantionate');hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;

figure(2);
subplot(211);
plot(G);grid;
title('Caracteristica de frecventa a filtrului trece-jos ideal');
subplot(212);
plot((1:256),g);grid;
title('Raspunsul la impuls al filtrului de tip sinus cardinal(FTJ ideal)');

figure(3);
plot([0:31],G(1:32));
title('Zoom pe caracteristica de frecventa a filtrului trece-jos ideal');
grid;

culoare=['g','r','c','m','y','k','g','r','c'];
figure(4);
for i=1:N,
    plot([1:M],out{i},culoare(i));hold on;
end
plot([1:M],real(simbol_rec),'-o',[1:M],real(simbol_rec));
hold off;
grid;
axis([0 255 -0.5 1.5]);
title('Raspunsul filtrului la secventa de date de la intrare');

figure(5);
plot(real(simbol_rec));hold on;
%stem((1:Lb:M),simbol_rec(1:Lb:M),'r');hold off;grid;
stem((1:Lb:M),real(simbol_rec(1:Lb:M)),'r');hold off;grid;
title('Semnalul de la iesirea filtrului, cu sublinierea momentelor de esantionare');

figure(6);
subplot(211);
plot(simbol);
title('Simboluri emise');hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;
subplot(212);
plot(simbol_detectat);
title('Simboluri detectate');hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;

```

**Lucrarea 4****Trecerea unei secvente de date unipolare printr-un filtru de tip cosinus patrat, canalul fiind fara zgomot**

```
t=0:1/32:8-1/32;
```

```
M=length(t);
```

```
%parametrii programului
```

```
Te=1/32;
```

```
T=1;
```

```
Lb=T/Te;
```

```
Omega0=2*pi/Lb;
```

```
N=M/Lb;
```

```
%generarea unei secvente aleatoare unipolare de N=8 simboluri si supraesantionarea ei cu Lb=32
```

```
simbol=[];
```

```
for i=1:N,
```

```
    sursa(i)=round(rand);
```

```
    simbol=[simbol sursa(i) zeros(1,31)];
```

```
end
```

```
%definirea caracteristicii de frecventa a filtrului cosinus patrat
```

```
for k=1:M/2,
```

```
    if k<=(M*Omega0/(2*pi)),
```

```
        G(k)=Lb*(cos((Lb*k*2*pi)/(4*M)))^2;
```

```
    else
```

```
        G(k)=0;
```

```
    end
```

```
end
```

```
for k=M:-1:M/2+1,
```

```
    G(k)=G(M-k+1);
```

```
end
```

```
%Calculul raspunsului la impuls al filtrului cosinus patrat
```

```
g=ifft(G,256);
```

```
g=[g(129:256) g(1:128)];
```

```
%gasirea reprezentarii spectrale a semnalului de la intrarea filtrului
```

```
Simbol=fft(simbol,256);
```

```
%Calculul secventei de date dupa trecerea prin filtru
```

```
Simbol_rec=Simbol.*G;
```

```
simbol_rec=ifft(Simbol_rec);
```

```
%detectia datelor de la iesirea filtrului prin compararea cu pragul
```

```
simbol_estimat=round(simbol_rec(1:Lb:M));
```

```
simbol_detectat=[];
```

```
for i=1:N,
```

```
    simbol_detectat=[simbol_detectat simbol_estimat(i) zeros(1,31)];
```

```
end
```

```
%se calculeaza cum raspunde filtrul la fiecare dintre bitii de date de la intrare
```

```
separat{1}=[simbol(1) zeros(1,255)];
```

```
Separat{1}=fft(separat{1},256);
```

```

Out{1}=Separat{1}.*G;
out{1}=real(iff(Out{1}));
for i=2:N,
    index1=(i-1)*Lb;
    index2=(i-1)*Lb+1;
    separat{i}=[zeros(1,index1) simbol(index2) zeros(1,M-index2)];
    Separat{i}=fft(separat{i},256);
    Out{i}=Separat{i}.*G;
    out{i}=real(iff(Out{i}));
end

```

```

figure(1);
plot(simbol);
title('Simboluri de date emise si supraesantionate');hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;

```

```

figure(2);
subplot(211);
plot(frecv,G(1:2*(M*Omega0/(2*pi))));grid;
title('Caracteristica de frecventa a filtrului cosinus patrat');
subplot(212);
plot((1:256),g);grid;
title('Raspunsul la impuls al filtrului de tip cosinus patrat');

```

```

culoare=['g','r','c','m','y','k','g','r','c'];
prag=0.5*ones(1,256);
figure(3);
for i=1:N,
    plot([1:M],out{i},culoare(i));hold on;
end
plot([1:M],real(simbol_rec),'-o',[1:M],real(simbol_rec));hold on;
plot([1:M],prag,'k--');
hold off;
grid;
axis([0 255 -0.5 1.2]);
title('Raspunsul filtrului la fiecare din bitii de date de la intrare');

```

```

figure(4);
plot(real(simbol_rec));hold on;
stem((1:Lb:M),real(simbol_rec(1:Lb:M)),'r');hold off;grid;
title('Semnalul de la iesirea filtrului, cu sublinierea momentului de esantionare');
axis([0 256 0 1]);

```

```

figure(5);
subplot(211);
plot(simbol);
title('Simboluri emise');hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;
subplot(212);
plot(simbol_detectat);hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;
title('Simboluri detectate');

```

**Lucrarea 5**

**Transmisie in banda de baza, fara interferenta inter-simbol, canalul fiind cu zgomot. Se utilizeaza o pereche de filtre de tip "radical din cosinus patrat"**

```

clear;
t=0:1/32:8-1/32;
M=length(t);

%parametrii programului
Te=1/32;
T=1;
Lb=T/Te;
Omega0=2*pi/Lb;
N=M/Lb;

%generarea unei secvente aleatoare unuipolare de N=8 simboluri si supraesantionarea ei cu Lb=32
x=[];
for i=1:N,
    sursa(i)=round(rand);
    x=[x sursa(i) zeros(1,31)];
end

%generarea unui zgomot alb, gaussian, de varianta unitara
for i=1:M,
    zgomot(i)=sqrt(0.1)*randn;
end

%definirea caracteristicii de frecventa a filtrului radical din cosinus patrat(emisie)
for k=1:M/2,
    if k<=(M*Omega0/(2*pi)),
        Ge(k)=sqrt(Lb)*(cos((Lb*k*2*pi)/(4*M)));
    else
        Ge(k)=0;
    end
end
for k=M:-1:M/2+1,
    Ge(k)=Ge(M-k+1);
end

%Filtrul "pereche" celui de la emisie
Gr=Ge;

frecv=(0:2*(M*Omega0/(2*pi))-1);

%calculul raspunsului la impuls al filtrului radacina din cosinus patrat
ge=ifft(Ge,256);
ge=[ge(129:256) ge(1:128)];

%Gasirea reprezentarii spectrale a semnalului de la intrarea filtrului
X=fft(x,256);

%calculul secventei de date dupa trecerea prin filtru

```



```

X1=X.*Ge;
x1=ifft(X1);

%calculul raportului semnal/zgomot inaintea filtrului adaptat
Psemnal=sum(abs(x1).^2);
Pzgomot=sum(abs(zgomot).^2);
RSZin=Psemnal/Pzgomot;

%peste acest semnal se suprapune zgomot in canal
x2=x1+zgomot;

%se calculeaza spectrul semnalului cu zgomot
X2=fft(x2);

%semnalul se trece prin filtrul de la partea de receptie
Y1=X2.*Gr;
y1=ifft(Y1);

%detectia datelor de la iesirea filtrului prin compararea cu pragul
yes=real(y1(1:Lb:M));
for i=1:N,
if yes(i)>0.5,
    y(i)=1;
else y(i)=0;
end
end

x_receptionat=[];
for i=1:N,
    x_receptionat=[x_receptionat y(i) zeros(1,31)];
end

%daca peste semnalul util nu s-ar fi suprapus zgomot...
Y1fzg=X1.*Gr;
y1fzg=ifft(Y1fzg);
Pzgout=sum(abs(y1fzg-y1).^2);
Psout=sum(abs(y1fzg).^2);
RSZout=Psout/Pzgout;
IRSZ=RSZout/RSZin;

figure(1);
plot(x);
title('Simboluri de date emise si supraesantionate');hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;

figure(2);
subplot(211);
plot(frecv,Ge(1:2*(M*Omega0/(2*pi))));grid;
title('Caracteristica de frecventa a filtrului radacina din cosinus patrat(Ge)');
subplot(212);
plot((1:256),ge);grid;
title('Raspunsul la impuls al filtrului de tip radacina din cosinus patrat');

```

```
figure(3);
plot(real(x1));
title('Semnalul util care se transmite in canal');
grid;
```

```
figure(4);
plot(zgomot);
title('Zgomotul suprapus peste semnalul util');
grid;
```

```
figure(5);
plot(real(x2));
title('Semnalul perturbat de zgomot adus la intrarea filtrului de la receptie');
grid;
```

```
disp('Puterea medie a semnalului este :');disp(Psemnal/M);
disp('Puterea medie a zgomotului este :');disp(Pzgomot/M);
disp('RSZ la intrarea filtrului adaptat este :');disp(RSZin);
disp('Apasati o tasta pentru a continua...');
pause;
```

```
figure(6);
plot(real(y1));hold on;
stem((1:Lb:M),real(y1(1:Lb:M)),'r');hold off;grid;
title('Semnalul la iesirea filtrului, cu sublinierea momentului de esantionare');hold on;
prag=(0.5).*ones(1,M);
plot([0:M-1],prag,'b--');hold off;
```

```
figure(7);
plot(real(y1));hold on;plot(real(y1fzg),'g-');hold on;
title('O comparatie intre forma semnalului receptionat cu/fara zgomot aditiv');
legend(' : semnal afectat de zgomot',' : semnal neafectat de zgomot');
stem((1:Lb:M),zeros(1,N),'r');hold off;grid;
```

```
disp('Puterea semnalului util la iesire este :');disp(Psout/M);
disp('Puterea zgomotului la iesire este :');disp(Pzgout/M);
disp('RSZ la iesirea filtrului adaptat este :');disp(RSZout);
disp('Raportul semnal zgomot s-a imbunatatit de : ');disp(IRSZ);
disp('Apasati o tasta pentru a continua...');
pause;
```

```
figure(8);
subplot(211);
plot(x);
title('Simboluri emise');hold on;
subplot(212);
plot(x_receptionat);
title('Simboluri detectate');
```

**Lucrarea 6****Transmisiune de date in banda de baza cu interferenta controlata inter-simbol. Trecerea unei secvente de date printr-un filtru de tip cosinus, canalul fiind fara zgomot**

```

clear;

t=0:1/32:8-1/32;
M=length(t);

%parametrii programului
Te=1/32;
T=1;
Lb=T/Te;
Omega0=pi/Lb;
N=M/Lb;

%generarea unei secvente aleatoare unuipolare de N=8 simboluri si supraesantionarea ei cu Lb=32
x=[];
for i=1:N,
    sursa(i)=randn;
    if sursa(i)<0,
        sursa(i)=-1;
    else sursa(i)=1;
    end
    x=[x sursa(i) zeros(1,31)];
end

%generarea unei secvente de zgomot alb, gaussian si de medie nula
zgomot=sqrt(0.05).*randn(1,M);

%definirea caracteristicii de frecventa a filtrului cosinus
for k=1:M/2,
    if k<=(M*Omega0/(2*pi)),
        G(k)=2*Lb*(cos((Lb*k*pi)/M));
    else
        G(k)=0;
    end
end
for k=M:-1:M/2+1,
    G(k)=G(M-k+1);
end

%definirea unei caracteristici de tip "square root cosine"(radical din cosinus)
for i=1:M,
    Ge(i)=sqrt(G(i));
end
Gr=Ge;

omega=[(128:-1:1).*(-2*pi/M) (1:128).*(2*pi/M)];

```

*%calculul raspunsului la impuls al filtrului cosinus*

```
g=ifft(G,256);
g=[g(129:256) g(1:128)];
```

*%calculul raspunsului la impuls al filtrului de emisie tip "radical din cosinus"*

```
ge=ifft(Ge,256);
ge=[ge(129:256) ge(1:128)];
```

*%gasirea reprezentarii spectrale a semnalului de la intrarea filtrului*

```
X=fft(x,256);
```

*%calculul secventei de date dupa trecerea prin filtru*

```
X1=X.*G;
x1=ifft(X1);
```

*%se calculeaza cum raspunde filtrul cosinus la fiecare bit aplicat la intrare*

```
separat{1}=[x(1) zeros(1,255)];
Separat{1}=fft(separat{1},256);
Out{1}=Separat{1}.*G;
out{1}=real(ifft(Out{1}));
for i=2:N,
    index1=(i-1)*Lb;
    index2=(i-1)*Lb+1;
    separat{i}=[zeros(1,index1) x(index2) x(1,M-index2)];
    Separat{i}=fft(separat{i},256);
    Out{i}=Separat{i}.*G;
    out{i}=real(ifft(Out{i}));
end
```

*%esantionarea la receptie si detectia datelor de la iesirea filtrului prin compararea cu pragul*

```
y1(1)=real(x1(Lb/2));
if y1(1)>=0,
    decizie=1;
    y(1)=decizie;
else decizie=-1;
    y(1)=decizie;
end
```

```
for i=2:N,
    y1(i)=real(x1(-Lb/2+i*Lb));
    if (y1(i)-decizie)>=0,
        y(i)=1;
    else
        y(i)=-1;
    end
    decizie=y(i);
end
```

```
x_receptionat=[zeros(1,15)];
for i=1:N,
    x_receptionat=[x_receptionat y(i) zeros(1,31)];
end
```

```
x_receptionat=[zeros(1,15) x_receptionat(240) zeros(1,31) x_receptionat(16) zeros(1,31)
x_receptionat(16+32) zeros(1,31) x_receptionat(16+2*32) zeros(1,31) x_receptionat(16+3*32)
zeros(1,31) x_receptionat(16+4*32)zeros(1,31) x_receptionat(16+5*32) zeros(1,31)
x_receptionat(16+6*32) zeros(1,16)]
```

```
figure(1);
plot(x);
title('Simboluri de date emise si supraesantionate');hold on;
stem((1:Lb:M),zeros(1,M/Lb),'r');hold off;
```

```
figure(2);
subplot(211);
plot(omega,[G(129:256) G(1:128)]);grid;
title('Caracteristica de frecventa a filtrului cosinus');
xlabel('omega');axis([-pi/2 pi/2 0 60]);
subplot(212);
plot((1:256),g);grid;
title('Raspunsul la impuls al filtrului de tip cosinus ');
```

```
culoare=['g','r','c','m','y','k','g','r','c'];
figure(3);
for i=1:N,
    plot([1:M],out{i},culoare(i));hold on;
end
plot([1:M],real(x1),'-o');
hold off;
grid;
axis([0 256 -2 2]);
title('Raspunsul filtrului la secventa de date de la intrare');
```

```
figure(4);
plot(real(x1));hold on;
stem((Lb/2:Lb:M-Lb/2),y1,'r');hold off;grid;
title('Semnalul la iesirea filtrului, cu sublinierea momentului de esantionare');
```

```
figure(5);
subplot(211);
plot(x);
title('simboluri emise');
subplot(212);
plot(x_receptionat);
title('Simboluri detectate');
```

*%daca se utilizeaza o pereche de filtre...*

```
clc;
disp('In continuare vom retransmite secventa de date, utilizand o pereche de filtre de tip "square root cosine"...');
pause;
```

```
figure(6);
plot(real(ge));
grid;
```

```

title('raspunsul la impuls al filtrului de tip "square root cosine');

figure(7);
plot(omega,[Ge(129:256) Ge(1:128)]);grid;
title('Caracteristica de frecventa a filtrului "square root cosine"');
xlabel('omega');axis([-pi/4 pi/4 0 10]);

%calculul secventei de date dupa trecerea prin filtrul de emisie
Xe=X.*Ge;
xe=ifft(Xe,256);
figure(8);
plot(real(xe));
title('Simbolurile de date dupa trecerea prin filtrul de emisie');
grid;

%peste semnalul util se adauga zgomot
xr1=xe+zgomot;

figure(9);
plot(real(xr1));hold on;
grid;
title('Semnalul afectat de zgomot care va fi prelucrat de filtrul de receptie');
plot(real(xe),'r--');hold off;

%calculul secventei de date de la iesirea filtrului de receptie;
Xr1=fft(xr1);
Xr=Xr1.*Gr;
xr=ifft(Xr);

figure(10);
plot(real(xr));
hold on;
moment_esant=[16:Lb:M];
stem(moment_esant,real(xr(moment_esant)),'r');
hold off;
title('Semnalul de la iesirea filtrului cu sublinierea esantioanelor ce vor fi prelevate');
grid;

%luarea deciziei asupra simbolurilor emise...
yr(1)=real(xr(Lb/2));
if yr(1)>=0,
    decizie=1;
    yr(1)=decizie;
else decizie=-1;
    yr(1)=decizie;
end

for i=2:N,
yr(i)=real(xr(-Lb/2+i*Lb));
if (yr(i)-decizie)>=0,
    yr(i)=1;
else
    yr(i)=-1;

```

```
end
decizie=yr(i);
end
%rearanjarea datelor...

yreceptionat=[zeros(1,15) yr(N) zeros(1,Lb) yr(1) zeros(1,Lb) yr(2) zeros(1,Lb) yr(3) zeros(1,Lb)
yr(4) zeros(1,Lb) yr(5) zeros(1,Lb) yr(6) zeros(1,Lb) yr(7) zeros(1,Lb/2)];

figure(11);
subplot(211);
plot(x);
title('simboluri emise');
subplot(212);
plot(yreceptionat);
title('Simboluri detectate');
```