

4. Explicați utilitatea filtrului median în netezirea imaginilor. Ce criteriu de optimalitate stă la baza definiției filtrului median?

2.6. Filtre. « Filtrul median »

mai mare, vizibilitatea lor fiind puțin influențată favorabil de operația de filtrare liniară. Concomitent, se manifestă și efectul nefavorabil al filtrelor liniare, de estompare a conturilor. Atât zgomotul binar cât și conturile violează flagrant ipotezele de optimalitate pentru filtrele liniare. Prin comparație, filtrul median 5x5 (Fig. 2.26 b), elimină aproape integral zgomotul binar și în același timp afectează mai puțin redarea conturilor.

Filtrul median este un *operator neliniar*, ce înlocuiește fiecare pixel cu mediana pixelilor aflați într-o fereastră centrată în jurul acestuia. *Mediana* unui șir de numere reprezintă elementul aflat la mijlocul șirului, după *ordonarea* lui. Prin ordonare, vom înțelege în general ordonarea în sensul crescător, deși acest aspect este neimportant din punctul de vedere al definiției medianei. Schema bloc a procedurii de prelucrare pentru un pixel de ieșire este redată în fig.2.27. Vom nota cu f_1, f_2, \dots, f_N pixelii din fereastră și cu $f_{(1)}, f_{(2)}, \dots, f_{(N)}$ pixelii din șirul ordonat. Rezultatul prelucrării este $f_{(m)}$, cu proprietatea: $m = (N + 1) / 2$. De menționat că numărul elementelor din fereastră, N , se alege impar, astfel ca m să fie un număr întreg.

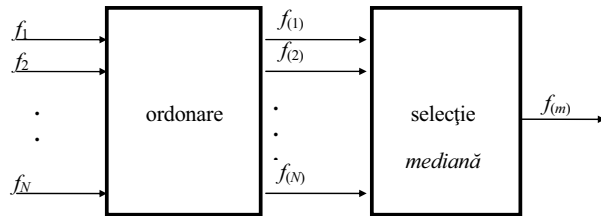


Fig. 2.27 Schema-bloc a filtrului median

Un exemplu de calcul pentru un filtru median cu fereastră pătrată de 3x3 pixeli se dă în Fig. 2.28.

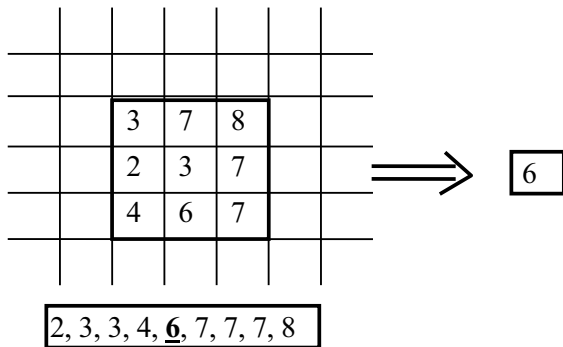


Fig. 2.28. Exemplu de calcul al medianei într-o fereastră pătrată 3x3

Proprietăți ale filtrului median

- *Nelinaritate*

Selecția este o operație neliniară. Astfel,

$$\text{mediana}\{f_1 + f_2\} \neq \text{mediana}\{f_1\} + \text{mediana}\{f_2\}.$$

Totuși,

$$\text{mediana}\{c f\} = c \text{ mediana}\{f\},$$

$$\text{mediana}\{c + f\} = c + \text{mediana}\{f\}.$$

- *Efectul asupra mediei*

Mediana modifică media imaginii, dacă distribuția intensității este nesimetrică.

- *Optimalitate*

Asemenea filtrului de mediere aritmetică, filtrul median posedă o anumită proprietate de optimalitate, în sensul că furnizează o estimare de eroare minimă a intensității dintr-o fereastră cu un nivel constant. În acest caz însă, eroarea minimizată este definită prin suma abaterilor în modul față de nivelul estimat:

$$\varepsilon(f_{(m)}) = \sum_{k=1}^N |f_k - f_{(m)}|.$$

Această proprietate a filtrului median, de a minimiza suma distanțelor la restul eșantioanelor din fereastră, poate servi și ca definiție mai generală a filtrului median, valabilă și pentru date vectoriale, de exemplu imagini color. Prin aplicarea independentă a filtrului median asupra componentelor color (de exemplu R,G,B) nu se mai garantează selecția medianei pentru cele trei componente de la același eșantion din fereastră, putând rezulta culori false, mai ales la zonele de tranziție dintre obiecte.

- *Rejecția zgomotului*

Așa cum s-a menționat deja, filtrul median este deosebit de eficient în rejecția zgomotului binar. Să presupunem că într-o zonă cu nivelul de gri constant se injectează un zgomot conținând în impulsuri, de mare amplitudine. Cât timp proporția pixelilor afectați de zgomot este sub 50% în fereastra de filtrare, filtrul median reconstituie semnalul perfect, ca și cum zgomotul nu ar fi existat! Pe de altă parte, filtrul median are performanțe mediocre în prezența zgomotului gaussian, pentru care filtrele liniare sunt mai bine adaptate.

- *Efectul asupra muchiilor*

Filtrul median păstrează muchiile mult mai bine decât filtrele de netezire liniare. O muchie treptată este redată perfect, pentru că filtrul median nu

4. Explicați utilitatea filtrului median în netezirea imaginilor. Ce criteriu de optimalitate stă la baza definiției filtrului median?

2.6. Filtre. « Filtrul median »

3/3

mediază ci *selectează* un anumit pixel din fereastră. Filtrul median păstrează rampele de luminanță.

- Efectul asupra punctelor, colțurilor și a liniilor subțiri

O proprietate uneori mai puțin favorabilă a filtrului median este aceea că el șterge punctele izolate, colțurile, liniile subțiri, și alte detalii de dimensiuni reduse în comparație cu fereastra de filtrare.

- Aplicare repetată

Filtrul median poate fi aplicat în mod repetat, rezultând o netezire mai pronunțată. După un număr de iterații, ieșirea tinde să se stabilizeze, deși acest lucru nu se întâmplă în mod necesar.

Filtrul median ponderat sau mediana cu repetiții se definește cu ajutorul unei măști cu ponderi, asemănător filtrelor liniare. Ponderea fiecărui pixel indică de câte ori se repetă acel pixel pentru a fi introdus în șirul ordonat. Procedeeul permite să li se acorde pixelilor o importanță dependentă de poziția lor în fereastră. În general, pixelii centrali vor fi ponderați mai puternic. De menționat că ponderile pot fi și numere neîntregi. Mediana se obține pornind de la o extremă a șirului ordonat și însumând ponderile aferente eșantioanelor șirului până când se cumulează jumătate din suma totală a ponderilor acordate. Un exemplu se dă în Fig. 2.29. Filtrul median cu repetiții păstrează contururile mai bine decât filtrul median convențional. În același timp, eficiența lui în eliminarea zgomotului binar este diminuată.

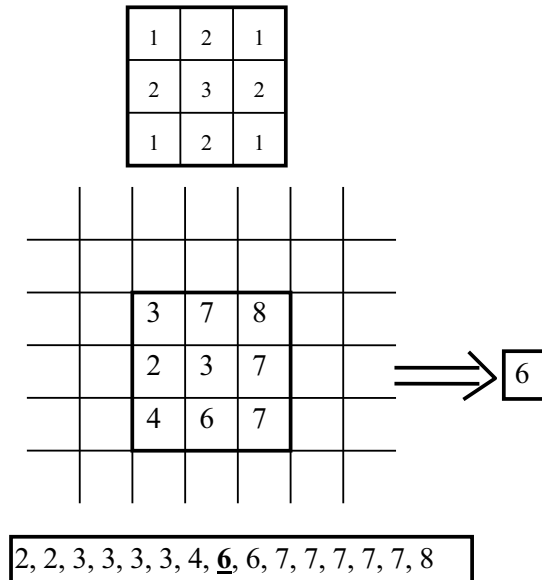


Fig. 2.29. Filtrul median ponderat

5. Descrieți metoda diferențială pentru trasarea cercurilor.

3. Tehnici de trasare. 3.2 Cercuri

3.2 Cercuri

Ecuția unui cerc ce trece prin origine:

$$f(x, y) = x^2 + y^2 - r^2 = 0. \quad (3.4)$$

Dacă centrul este în (x_c, y_c) , se face o translație, definită prin

$$\begin{aligned} x_t &= x + x_c, \\ y_t &= y + y_c. \end{aligned} \quad (3.5)$$

Forma (3.4) nu este adecvată pentru trasare. De exemplu, dând valori lui x , pentru a găsi y trebuie rezolvată o ecuație de gradul doi, trebuie verificat dacă soluțiile sunt reale etc.

Forma diferențială a ecuației cercului este:

$$2x dx + 2y dy = 0, \text{ sau} \quad (3.6)$$
$$\frac{dx}{dy} = -\frac{y}{x} \Leftrightarrow \begin{cases} dx = \varepsilon y, \\ dy = -\varepsilon x. \end{cases}$$

Incrementul ε se alege invers proporțional cu raza r .

Rezultă forma discretă a ecuației cercului:

$$\begin{cases} x_{n+1} = x_n + \varepsilon y_n, \\ y_{n+1} = y_n - \varepsilon x_n. \end{cases} \quad (3.7)$$

Problemă: folosind ecuația de mai sus, se trasează de fapt o spirală, pentru că raza se modifică ușor la fiecare iterație :

$$\begin{aligned} x_{n+1}^2 + y_{n+1}^2 - r^2 &= (x_n + \varepsilon y_n)^2 + (y_n - \varepsilon x_n)^2 - r^2 = \\ x_n^2 + y_n^2 - r^2 + \varepsilon^2 (x_n^2 + y_n^2) &> x_n^2 + y_n^2 - r^2. \end{aligned}$$

Soluție: Se modifică ecuațiile (3.7), în forma

$$\begin{cases} x_{n+1} = x_n + \varepsilon y_n, \\ y_{n+1} = y_n - \varepsilon x_{n+1}. \end{cases} \quad (3.8)$$

Se poate arăta (ca exercițiu lăsat în seama cititorului) că raza rămâne constantă în acest caz.

Algoritmul DDA pentru cerc

Folosind ecuațiile (3.8), se obține următorul algoritm DDA pentru cerc. Algoritmul exploatează simetria cercului și trasează simultan câte 8 puncte. De fapt se generează doar un octant (între 45^0 și 90^0).

```
cercDDA(int x_c, int y_c, int r)
{
    float x = 0, y = r;
    float epsilon = 1 / (float) r;
    while( x < y )
    {
        put8pix( round(x), round(y), x_c, y_c ); // pune 8 pixeli
        x += epsilon * y;
        y -= epsilon * x; // x a fost deja actualizat!
    }
} // end cercDDA()
```

```
put8pix( int x, int y, int x_c, int y_c )
{
    putpixel(x+x_c, y+y_c, 0);
    putpixel(x-x_c, y+y_c, 0);
    putpixel(x+x_c, y-y_c, 0);
    putpixel(x-x_c, y-y_c, 0);
    putpixel(y+y_c, x+x_c, 0);
    putpixel(y-y_c, x+x_c, 0);
    putpixel(y+y_c, x-x_c, 0);
    putpixel(y-y_c, x-x_c, 0);
}
```

6. Descrieți algoritmul Cohen-Sutherland pentru decuparea segmentelor de dreaptă.

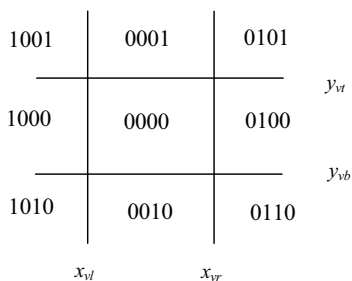
3.5 Tehnici de decupare

3.5 Tehnici de decupare

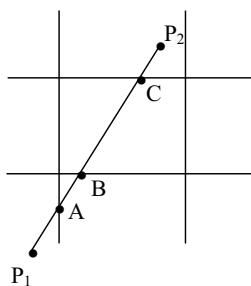
În procesul de manipulare și reprezentare a obiectelor grafice, este posibil ca unele părți ale acestora să nu se încadreze în interiorul ferestrei de vizualizare. O soluție simplă pentru rezolvarea evitărea redării punctelor situate în afara ferestrei de afișare este să se verifice pentru fiecare punct încadrarea în fereastră și să se traseze pixelul corespunzător numai în cazul afirmativ. O asemenea rezolvare nu este însă și rapidă. Pentru forme simple, de tipul dreaptă sau poligon, este mai eficient să se determine partea vizibilă înainte de afișare și să se reprezinte numai partea respectivă. Partea invizibilă se decupează.

Decuparea segmentelor de dreaptă 2D. Algoritmul Cohen-Sutherland

Planul grafic se împarte în nouă regiuni, codificate conform figurii 3.6.



a)



b)

Fig. 3.6. Decuparea unui segment de dreaptă

Codurile se construiesc atribuind fiecăruia din cei 4 biți valori după cum urmează:

- Bitul b_0 este 1 dacă punctul este deasupra marginii superioare, y_{vr}
- Bitul b_1 este 1 dacă punctul este dedesubtul marginii inferioare, y_{vb}
- Bitul b_2 este 1 dacă punctul este la dreapta marginii din dreapta, x_{vr}
- Bitul b_3 este 1 dacă punctul este la stânga marginii din stânga, x_{vl}

Algoritmul primește codurile capetelor segmentului de dreaptă, pe care le combină eficient pentru a identifica mai multe situații posibile. În exemplul din figură, P_1 are codul 1010, iar P_2 are codul 0001.

Pasul 1.

Se calculează funcția logică *pe biți SAU*. Dacă SAU dă zero pe toate pozițiile, ambele puncte sunt în interiorul ferestrei de afișare și nu este nevoie de decupare. Se trece la Pasul 5, pentru a se trasa dreapta. În caz contrar, se trece la pasul următor.

Pasul 2.

Dacă SAU nu dă 0 pe toate pozițiile, se face ȘI logic între coduri. Dacă funcția ȘI nu dă zero pe toate pozițiile (avem cel puțin un 1) segmentul este complet în afara ferestrei de afișare, deci nu avem de trasat nimic și algoritmul se încheie. În caz contrar, segmentul trebuie decupat și se trece la pasul următor.

Pasul 3.

Se inspectează codul lui P_1 . Astfel, dacă P_1 are codul este 0000, P_1 este un punct valid și se trece la pasul următor.

Dacă P_1 are nu are codul este 0000, urmează să fie decupat. Se verifică întâi bitul cel mai semnificativ, b_3 . Dacă $b_3=1$, P_1 este situat la stânga marginii din stânga. În acest caz (valabil pentru figura de mai sus), se calculează intersecția dintre segmentul P_1P_2 și marginea din stânga. Se obțin coordonatele punctului care înlocuiește P_1 . În exemplul din figură, este punctul A. Acesta primește codul corespunzător poziției și algoritmul se reia de la primul pas, cu $P_1=A$. În exemplul din figură, A primește codul 0010 (un punct de pe marginea stângă este considerat valid pentru trasare, deci bitul b_3 este setat pe zero).

Pasul 4.

Se inspectează codul lui P_2 . Se procedează la fel ca la Pasul 3, pentru punctul P_2 .

De observat că la acest punct se ajunge numai după ce punctul P_1 a fost complet rezolvat. În exemplul din figură, pentru că noul P_1 (A) are codul nenul, este din nou decupat, de data aceasta față de marginea de jos, pentru care bitul b_1 este setat pe 1. Se obține punctul B, care devine noul P_1 . La Pasul 4, P_2 va fi decupat și va primi coordonatele lui C.

Pasul 5.

Se trasează segmentul de dreaptă P_1P_2 decupat (devenit segmentul BC). Algoritmul se încheie.

Un exemplu de program C de decupare ce folosește două funcții pentru implementarea algoritmului de decupare Cohen-Sutherland se prezintă în continuare.

7. Ce metode de eliminare a erorilor alias sunt folosite în grafică?

3.6 Erori alias

3.6 Erori alias

Aspectul dreptelor și curbilor pe ecrane cu rezoluție slabă poate fi considerat inestetic, fiind perceptibilă reprezentarea prin puncte distincte. Liniile au un aspect rugos, grosimea este neuniformă. În contextul teoretic al prelucrării semnalelor, problema poate fi identificată ca una de subeșantionare. Aici vom face doar o scurtă discuție, mai degrabă pragmatică și intuitivă asupra modalităților de abordare posibile. Menționăm doar că termenul alias desemnează în limba engleză un nume echivalent. Asemeni spionilor ce-și ascund identitatea reală sub un alias, componente de frecvențe înalte din spectrul semnalului pot reapare cu frecvențele modificate, determinând false componente de joasă frecvență și de aici imposibilitatea reconstruirii corecte a semnalului, la o eșantionare insuficient de fină.

O primă soluție posibilă constă în creșterea rezoluției. Este o soluție costisitoare în sensul consumului de memorie și creșterii timpului de trasare. În plus, nu rezolvă problema de fond, o face doar mai puțin acută.

O soluție alternativă constă în filtrarea imaginilor prin convoluție cu un operator de tip trece jos (filtru de netezire). Asemenea filtre se vor prezenta într-un capitol special dedicat. Filtrarea transformă imaginea binară (cu numai două niveluri de gri, corespunzător albului și negrului) într-o imagine cu nuanțe de gri.

O soluție a cărei idee este foarte simplă, o reprezintă eșantionarea de suprafață neponderată, ilustrată în figura 3.10.

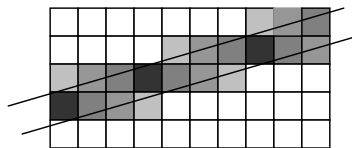


Fig. 3.10. Eșantionarea cu suprafață neponderată

Intensitatea fiecărui pixel este proporțională cu procentul din suprafața sa ce se suprapune cu linia, considerată a avea o grosime finită, precizată. Prin intensitate înțelegem în contextul de față negativul ei. Astfel, intensității maxime, 255 îi corespunde negrul, 0. Valorii 100 îi corespunde $255 - 100$ etc. Se obține o redare cu tranziții mai line a conturilor, cu un aspect mai natural, deși aparent metoda antrenează o pierdere de rezoluție prin îngroșarea de facto a traseului liniei. O redare și mai realistă a liniilor se obține prin eșantionare de suprafață ponderată. Metoda reprezintă o generalizare a ideii precedente, prin care intensitatea unui pixel depinde nu numai de procentul din aria sa suprapus cu traseul liniei ci și de distanța la mediana liniei. O interpretare posibilă a eșantionării neponderate ar fi următoarea. Pe fiecare pixel suprapunem centrat un cub cu latura egală cu cea a pixelului. Dacă linia intersectează pixelul, delimitează un subvolum al cubului (proporțional cu suprafața de intersecție). Intensitatea pixelului este apoi stabilită proporțională cu subvolumul delimitat de linia ce se trasează.

La eșantionarea de suprafață ponderată, figura centrată pe pixel nu mai este un cub. Poate fi înlocuită de exemplu cu un con, a cărui înălțime descrește monoton de la centru spre margini, dar se poate utiliza orice alt corp cu proprietăți similare (sferă, clopot gaussian etc.). Corpurile (ce stabilesc până la urmă dimensiunea pixelului) pot depăși în dimensiuni pasul rețelei, ceea ce înseamnă o suprapunere a zonelor acoperite de pixeli vecini.

Nimic nou în acest aranjament, ținând cont de faptul că acest lucru se întâmplă curent la toate monitoarele bazate pe tuburi catodice, la care spotul de explorare are oalură aproximativ gaussiană și acoperă semnificativ mai mult decât o linie de explorare.

Tehnicile anti alias sunt incorporate uzual direct în algoritmi de trasare.

8. În ce constă utilitatea coordonatelor omogene în grafica computerizată? Exemplificați.

4.2 Coordonate omogene și reprezentări matriciale ale transformărilor 2D,

4.3 Transformări geometrice 3D

4.2 Coordonate omogene și reprezentări matriciale ale transformărilor 2D

Reprezentările matriciale introduse prezintă inconvenientul că translația este descrisă de o ecuație cu formă diferită de cele pentru scalare sau rotație. Acest neajuns poate fi înlăturat simplu prin introducerea coordonatelor omogene. Coordonatele omogene 2D ale unui punct sunt de forma unui triplet (x_h, y_h, w) . Variabila suplimentară nu este propriu-zis o coordonată, deși se poate găsi și o interpretare de acest tip. Rolul ei este de a permite manipularea convenabilă a calculelor cu transformări geometrice în forma matricială. Coordonatele carteziene se obțin pe baza coordonatelor omogene prin ecuațiile:

$$x = x_h / w,$$
$$y = y_h / w.$$

Frecvent se alege $w=1$, împărțirea fiind astfel evitată, dar există situații în care este util ca w să poată lua valori diferite de 1.

În coordonate omogene, translația se poate scrie în forma:

$$\mathbf{p}^* = \mathbf{T}_p, \text{ sau}$$
$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Menționăm că unii autori preferă să scrie ecuația de mai sus cu vectorul linie al coordonatelor premultiplicând matricea de transformare, în speță matricea de translație. În acest caz matricea de translație are forma transpusă, pentru că

$$(\mathbf{T}_p)^T = \mathbf{p}^T \mathbf{T}^T.$$

În coordonate omogene toate cele trei transformări introduse au forma generală comună. Ceea ce diferă este numai forma matricii din ecuația de transformare. Pentru scalare, ecuația ia forma:

$$\mathbf{p}^* = \mathbf{S}_p, \text{ sau}$$
$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Respectiv pentru rotație forma

$$\mathbf{p}^* = \mathbf{R}_p, \text{ sau}$$
$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Ce se întâmplă dacă facem două translații succesive cu (d_{x1}, d_{y1}) și (d_{x2}, d_{y2}) ? Din proprietățile elementare ale operațiilor cu vectori, anticipăm că rezultatul este echivalent cu o translație unică cu vectorul-sumă $(d_{x1} + d_{x2}, d_{y1} + d_{y2})$. Este ușor de verificat că:

$$\mathbf{p}^* = \mathbf{T}_2(\mathbf{T}_1\mathbf{p}) = (\mathbf{T}_2\mathbf{T}_1)\mathbf{p},$$
$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x1} + d_{x2} \\ 0 & 1 & d_{y1} + d_{y2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Similar se comportă și celelalte două transformări. Ele pot fi simplu concatenate, prin multiplicări succesive ale matricilor de transformare. Mai mult, o succesiune de transformări de tipul translație, urmată de rescalare, urmată de a doua translație, urmată de o rotație, se poate realiza prin multiplicare cu o singură matrice $\mathbf{M} = \mathbf{R}\mathbf{T}_2\mathbf{S}\mathbf{T}_1$.

Un exemplu de folosire a proprietății de concatenare a transformărilor constă în rotația unei figuri în jurul unui punct arbitrar, P, nesituat în originea sistemului de coordonate. Transformarea se poate realiza prin succesiunea:

1. Translație ce aduce P în origine
2. Rotație
3. Translație ce readuce P la coordonatele inițiale

Dacă P are coordonatele (x,y) și unghiul de rotație este α , succesiunea de transformări necesare are matricea de transformare

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix} =$$
$$\begin{bmatrix} \cos \alpha & -\sin \alpha & x_1(1 - \cos \alpha) + y_1 \sin \alpha \\ \sin \alpha & \cos \alpha & y_1(1 - \cos \alpha) - x_1 \sin \alpha \\ 0 & 0 & 1 \end{bmatrix}$$

8. În ce constă utilitatea coordonatelor omogene în grafica computerizată? Exemplificați.

4.2 Coordonate omogene și reprezentări matriciale ale transformărilor 2D,

4.3 Transformări geometrice 3D

O abordare similară este posibilă pentru rescalarea unui obiect în jurul unui punct oarecare, de exemplu în jurul centrului de greutate al obiectului. Succesiunea de transformări este: translație, scalare, retranslație.

Este interesant de remarcat faptul că operațiile de translație comută. La fel și cele de scalare sau rotație. Mai mult, scalarea cu factor egal pe direcțiile x și y comută cu rotația. Matricile implicate sunt comutative. În general însă, înmulțirea matricilor este numai asociativă, nu și comutativă.

4.3 Transformări geometrice 3D

Transformările geometrice 3D pot fi exprimate avantajos folosind coordonate omogene 3D. Acestea se obțin adăugând vectorilor 2D o componentă suplimentară, corespunzătoare coordonatei z . Matricile de transformare devin matrici 4×4 .

Există două sisteme de coordonate carteziene 3D (Fig. 4.1), denumite *de mâna stângă* și respectiv *de mâna dreaptă*. În cele ce urmează, vom folosi sistemul de mâna dreaptă.

În spațiul 3D, rotațiile sunt mai complexe decât în spațiul 2D. Rotația 3D are loc în jurul unei axe. Pentru sistemul de mâna dreaptă, prin convenție, sensul de rotație pozitiv este cel trigonometric, privind dinspre axa de rotație spre origine. O rotație pozitivă de 90° transformă una din axe în cealaltă axă, în planul de rotație.

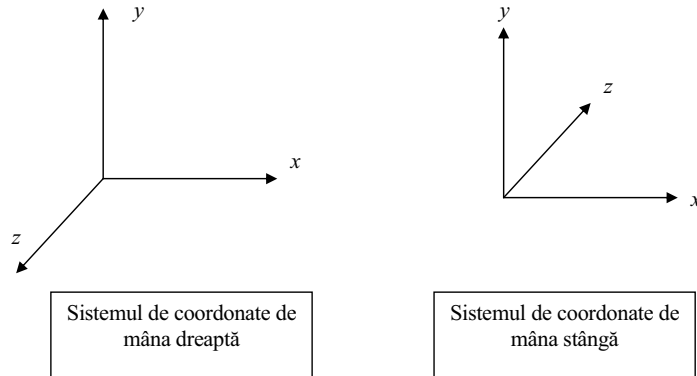


Fig. 4.1. Sisteme de coordonate carteziene 3D utilizate în grafică

Translația 3D se poate scrie în forma:

$$\mathbf{p}^* = \mathbf{T}\mathbf{p}, \text{ sau}$$

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scalarea 3D are forma:

$$\mathbf{p}^* = \mathbf{S}\mathbf{p}, \text{ sau}$$

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Matricea de rotație în jurul axei z (în planul x - y) are forma:

$$\mathbf{R}_z = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matricea de rotație pentru axa x este:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matricea de transformare pentru rotație în jurul axei y este:

$$\mathbf{R}_y = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ultimele două matrici se pot obține din prima prin permutări circulare succesive în submatricea 3×3 din stânga sus.

Rotația în jurul unei axe oarecare, definită de două puncte P_1, P_2 , cu un unghi α se poate realiza, de exemplu, în următorii pași:

1. translație ce aduce P_1 în originea sistemului de coordonate
2. rotație ce aduce P_2 în planul xOz

8. În ce constă utilitatea coordonatelor omogene în grafica computerizată? Exemplificați.

4.2 Coordonate omogene și reprezentări matriciale ale transformărilor 2D,

4.3 Transformări geometrice 3D

3/3

3. rotație în jurul axei y , ce aduce P_2 pe axa z
4. rotația dorită cu unghiul α , în jurul axei z
5. rotație în jurul axei y , pentru inversarea efectului rotației de la punctul 3
6. rotație în jurul axei x , pentru inversarea efectului rotației de la punctul 2
7. translație pentru a readuce P_1 în poziția inițială.

Matricea de transformare se poate scrie în forma:

$$M = T_{P_1} R_x R_y R_z R_y R_x T_{P_1} .$$

4.4 Proiecții

Redarea unui obiect grafic 3D pe ecran presupune o reprezentare 2D. Trecerea la reprezentarea 2D se poate realiza folosind o transformare de *proiecție*. În grafica de calculator se folosesc mai multe tipuri de proiecții. Există două categorii majore:

- proiecții paralele
- proiecția de perspectivă

O proiecție este definită de două elemente geometrice:

- planul de proiecție
- punctul de proiecție

Punctele imaginii proiectate se obțin ca intersecții cu planul de proiecție ale unor segmente de dreaptă ce unesc puncte ale obiectului cu punctul de proiecție.

Proiecțiile paralele corespund reprezentărilor necesare în desenul tehnic ingineresc. În acest caz, punctul de proiecție este situat la infinit. Dreptele de proiecție sunt paralele între ele și uzual paralele și cu una din axele sistemului. Proiecția de perspectivă corespunde vederii naturale și generează imagini cu aspect mai realist. Obiectele îndepărtate au dimensiuni micșorate în conformitate cu efectul de perspectivă. Paralelismul dreptelor și mărimile unghiurilor nu se păstrează, însă, ceea ce poate fi inconvenient pentru reprezentările ingineresti.

Din punct de vedere practic, este mai simplu aranjamentul în care planul de proiecție coincide cu unul din planurile determinate de o pereche de axe de coordonate. Uzual, este planul xOy . Proiecția unui punct din lumea reală pe planul xOy este ilustrată în figura 4.2.

Folosind asemănări de triunghiuri, se poate arăta ușor că

$$\frac{-x_i}{f} = \frac{x_r - x_i}{z_r} = \frac{x_r}{z_r - f},$$

de unde rezultă imediat că:

9. Ce este o curbă Bezier? Definiție, proprietăți utile.

5.3 Curbe Bezier, 5.2 Proprietati utile ale bazelor, 5.4 Algoritmul Casteljau

1/4

Ecuția de sinteză a unei curbe poate fi interpretată în felul următor: fiecare punct de pe curbă reprezintă o medie ponderată a punctelor \mathbf{p}_i , pe care le vom denumi *puncte de control*. Învelișul convex al punctelor de control este *poligonul convex* de arie minimă ce cuprinde toate punctele de control în interiorul sau pe conturul său. Este poligonul care s-ar obține plasând o badă elastică în jurul punctelor de control. Acesta include poligonul de control, al cărui interior este redat umbrat în (fig. 5.1).

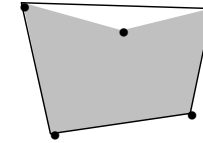


Fig. 5.1. Poligonul convex atașat unui poligon oarecare

Proprietatea învelișului convex constă în faptul că forma generată se află în interiorul învelișului convex dacă:

$$\sum_{i=0}^n b_i(u) = 1,$$
$$b_i(u) \geq 0,$$

adică, funcțiile de bază au suma identic egală cu 1 și sunt nenegative pe intervalul de definiție (uzual u aparține intervalului $[0,1]$).

Invarianța la transformări afine

Transformările afine includ translația, rotația și scalarea. Proprietatea invarianței la transformările afine constă în faptul că figura generată de punctele de control care au suferit o transformare afină corespunde transformării afine a figurii corespunzătoare punctelor inițiale. Notând transformarea afină cu $\Phi()$, proprietatea enunțată presupune că

$$\Phi(\mathbf{r}(u)) = \sum_{i=0}^n \Phi(\mathbf{p}_i) b_i(u).$$

Proprietatea este deosebit de utilă în manipularea curbelor (de exemplu în animație), deoarece nu este necesar să efectuăm transformarea pentru fiecare punct al curbei ci numai pentru punctele de control, utilizând apoi rutinele obișnuite de trasare a curbei.

Observație

O curbă reprezentată parametric prin punctele ei de control poate fi redată cu o rezoluție oricât de mare este necesară. Dacă aceeași curbă ar fi reprezentată ca o mini-imagine, rezoluția ar fi limitată de formatul

5.2. Proprietăți utile ale bazelor

Majoritatea bazelor utilizate curent în grafica de calculator posedă două proprietăți importante:

- proprietatea învelișului convex
- proprietatea invarianței la transformările afine

Proprietatea învelișului convex

9. Ce este o curbă Bezier? Definiție, proprietăți utile.

5.3 Curbe Bezier, 5.2 Proprietati utile ale bazelor, 5.4 Algoritmul Casteljau

2/4

imaginii. Desigur există și pentru imagini posibilitatea de creștere a rezoluției prin tehnici ce folosesc interpolarea, dar rezultatele pentru curbe nu sunt totdeauna cele așteptate. Astfel, prin creșterea rezoluției, o linie cu grosime de un pixel poate deveni o linie cu grosime mai mare de un pixel.

5.3. Curbe Bézier

Funcțiile de bază ale unei curbe Bézier de gradul n sunt:

$$b_{i,n}(u) = C_n^i u^i (1-u)^{n-i},$$

$$C_n^i = \frac{n!}{i!(n-i)!}.$$

Pentru gradul 3, utilizat cel mai frecvent în grafica de calculator, funcțiile de bază Bézier (fig. 2) sunt:

$$b_{0,3}(u) = (1-u)^3$$

$$b_{1,3}(u) = 3u(1-u)^2$$

$$b_{2,3}(u) = 3u^2(1-u)$$

$$b_{3,3}(u) = u^3$$

O curbă Bézier de gradul trei se scrie:

$$\mathbf{r}(u) = \sum_{i=0}^3 \mathbf{p}_i b_{i,3}(u) = \mathbf{p}_0(1-u)^3 + 3\mathbf{p}_1 u(1-u)^2 + 3\mathbf{p}_2 u^2(1-u) + \mathbf{p}_3 u^3.$$

($0 < u < 1$).

Care este avantajul față de baza de putere?

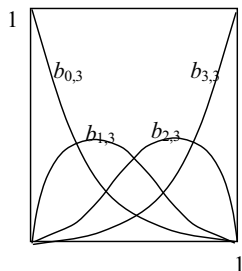


Fig. 5.2. Ilustrare a funcțiilor de bază Bezier

Avantajul esențial este acela că pentru curbele Bezier avem o semnificație utilă a punctelor de control, care permite utilizatorului să anticipeze simplu efectele modificării poziției acestor puncte și astfel să genereze mai ușor forma dorită.

În primul rând, observăm că

$$\mathbf{r}(0) = \mathbf{p}_0,$$

$$\mathbf{r}(1) = \mathbf{p}_3,$$

deci curba Bézier trece prin primul și prin ultimul punct de control.

Pentru a vedea semnificațiile celorlalte două puncte de control, derivăm curba în raport cu u :

$$\frac{d\mathbf{r}(u)}{du} = \mathbf{r}'(u) = 3[(\mathbf{p}_1 - \mathbf{p}_0)(1-u)^2 + 2(\mathbf{p}_2 - \mathbf{p}_1)u(1-u) + (\mathbf{p}_3 - \mathbf{p}_2)u^2]$$

Rezultă că:

$$\mathbf{r}'(0) = 3(\mathbf{p}_1 - \mathbf{p}_0),$$

$$\mathbf{r}'(1) = 3(\mathbf{p}_3 - \mathbf{p}_2).$$

Dezvoltarea funcției în serie Taylor în jurul punctului $u=t$ se scrie:

$$\mathbf{r}(t+h) = \mathbf{r}(t) + h\mathbf{r}'(t),$$

ceea ce arată că $\mathbf{r}'(t)$ este direcția tangentei în punctul $u=t$. Prin urmare, direcția tangentei în punctul inițial este $3(\mathbf{p}_1 - \mathbf{p}_0)$, adică direcția dreptei ce unește primele două puncte de control (multiplicarea vectorului cu factorul 3 nu schimbă direcția). Ecuația de mai sus mai sugerează faptul că, pe măsură ce punctul de control \mathbf{p}_1 este mai îndepărtat de \mathbf{p}_0 , curba înaintează mai mult pe direcția tangentei în origine. O interpretare similară există și pentru tangenta în punctul final.

Funcțiile Bézier respectă condițiile necesare pentru a poseda proprietatea învelișului convex, pentru că funcțiile de bază sunt nenegative și reprezintă de fapt termenii dezvoltării binomului

$$((1-u)+u)^n = 1.$$

În concluzie, punctele de control pot fi utilizate de o manieră simplă pentru asigurarea continuității de ordinul zero și unu ale unei curbe compuse din segmente Bézier de ordinul trei. Ordinul trei este suficient de ridicat pentru a permite o mare varietate de curbe, în general nesimetrice. Totodată, curbele generate sunt netede, o proprietate dorită de cele mai multe ori.

9. Ce este o curbă Bezier? Definiție, proprietăți utile.

5.3 Curbe Bezier, 5.2 Proprietati utile ale bazelor, 5.4 Algoritmul Casteljau

5.4. Algoritmul Casteljau

Funcțiile de bază Bézier de un anumit ordin se pot obține prin interpolarea liniară a funcțiilor de rang inferior, conform ecuației:

$$b_{i,n}(u) = ub_{i,n-1}(u) + (1-u)b_{i-1,n-1}(u),$$

ceea ce se poate verifica ușor, folosind ecuațiile de definiție. În consecință, un punct de pe curbă, situat la coordonata u , poate fi generat recursiv, folosind următorul algoritm:

1. Punctele inițiale sunt chiar punctele de control:

$$\begin{aligned} \mathbf{p}_i^0 &= \mathbf{p}_i, \\ i &= 0, 1, \dots, n - r, \\ r &= 0. // \text{semnifică iterația} \\ u &\text{ se inițializează cu valoarea dorită, în intervalul } 0-1. \end{aligned}$$

2. Pentru $r=1, 2, \dots, n$, se determină succesiv punctele:

$$\mathbf{p}_i^r(u) = (1-u)\mathbf{p}_i^{r-1}(u) + u\mathbf{p}_{i+1}^{r-1}.$$

Se observă că:

- fiecare punct nou se obține prin interpolarea liniară a două puncte consecutive din iterația precedentă
- la fiecare iterație, numărul punctelor calculate se reduce cu unul

Interesant este faptul că ultimul punct care se obține, este $\mathbf{p}_0^n = \mathbf{r}(u)$ și este pe curbă!

Cititorul este invitat să verifice afirmația pentru $n=3$, folosind calculul recursiv indicat de algoritm. O ilustrare grafică a algoritmului Casteljau pentru $u=1/2$ se dă în figura 5.3.

Se poate demonstra că punctul final obținut divide curba Bézier în două curbe Bézier (*subdiviziune*) ale căror puncte de control sunt chiar $\mathbf{p}_0^0, \mathbf{p}_0^1, \mathbf{p}_0^2, \mathbf{p}_0^3$, respectiv $\mathbf{p}_3^3, \mathbf{p}_2^2, \mathbf{p}_1^1, \mathbf{p}_0^0$, ce au fost deja calculate. În consecință, este posibilă trasarea curbei folosind subdiviziunea recursivă. Costul de calcul pentru fiecare punct nou este de numai patru medii ponderate. Dacă se alege $u=1/2$, multiplicările pot fi evitate: împărțirea cu doi se poate implementa ca o operație de deplasare la dreapta cu o unitate (operatorul $\gg 1$ în limbajul C).

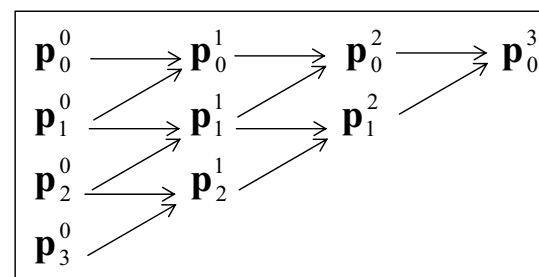
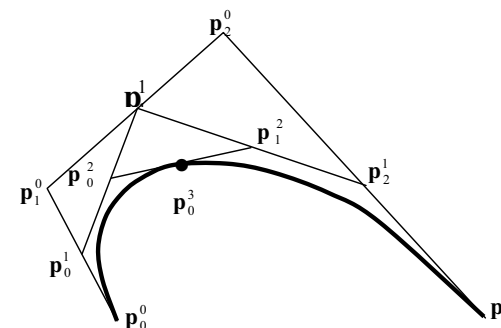


Fig. 5.3. Ilustrare a punctelor generate prin algoritmul Casteljau

Noile puncte de control ale celor două segmente se reduc la:

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{p}_0, & \mathbf{r}_0 &= \mathbf{q}_3, \\ \mathbf{q}_1 &= \frac{\mathbf{p}_0 + \mathbf{p}_1}{2}, & \mathbf{r}_1 &= \frac{\mathbf{p}_1 + \mathbf{p}_2}{4} + \frac{\mathbf{r}_2}{2}, \\ \mathbf{q}_2 &= \frac{\mathbf{q}_1 + \mathbf{p}_1 + \mathbf{p}_2}{4}, & \mathbf{r}_2 &= \frac{\mathbf{p}_2 + \mathbf{p}_3}{2}, \\ \mathbf{q}_3 &= \frac{\mathbf{q}_2 + \mathbf{r}_1}{2}, & \mathbf{r}_3 &= \mathbf{p}_3. \end{aligned}$$

Soluția este extrem de convenabilă și pentru o eventuală implementare hardware. Algoritmul se termină când învelișul convex al poligonului

9. Ce este o curbă Bezier? Definiție, proprietăți utile.

5.3 Curbe Bezier, 5.2 Proprietati utile ale bazelor, 5.4 Algoritmul Casteljau

4/4

de control Bézier este suficient de plat pentru a se putea aproxima segmentul curent de curbă cu o dreaptă, care se trasează.

Algoritmul Casteljau este un punct forte major al reprezentării Bézier a curbelor. El are însă și o importanță teoretică. Pentru că interpolarea liniară este invariantă la transformarea afină, algoritmul demonstrează faptul că *reprezentarea Bézier are proprietatea invarianței la transformări affine.*

5.5. Formularea matricială a curbelor Bézier

O convenție alternativă pentru a specifica o curbă Bézier este cu ajutorul matricii Bézier și a vectorilor bazei de putere:

$$\mathbf{r}(u) = \mathbf{U} \mathbf{M}_B \mathbf{P}_c = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}.$$

De precizat, că vectorii \mathbf{r} și \mathbf{p} sunt vectori-linie în formularea de mai sus. Funcțiile de bază Bézier cubice se obțin din produsul primelor două matrici în ecuația de mai sus.

Formularea matricială a curbelor Bézier este utilă în următoarele două situații:

- implementarea hardware la reprezentarea în altă bază
- conversia între baze

10. Definiți și exemplificați utilizarea algoritmului Casteljau.

5.4 Algoritmul Casteljau

5.4. Algoritmul Casteljau

Funcțiile de bază Bézier de un anumit ordin se pot obține prin interpolarea liniară a funcțiilor de rang inferior, conform ecuației:

$$b_{i,n}(u) = ub_{i,n-1}(u) + (1-u)b_{i-1,n-1}(u),$$

ceea ce se poate verifica ușor, folosind ecuațiile de definiție. În consecință, un punct de pe curbă, situat la coordonata u , poate fi generat recursiv, folosind următorul algoritm:

1. Punctele inițiale sunt chiar punctele de control:

$$\begin{aligned} \mathbf{p}_i^0 &= \mathbf{p}_i, \\ i &= 0, 1, \dots, n - r, \\ r &= 0. // \text{semnifică iterația} \\ u &\text{ se inițializează cu valoarea dorită, în intervalul } 0-1. \end{aligned}$$

2. Pentru $r=1, 2, \dots, n$, se determină succesiv punctele:

$$\mathbf{p}_i^r(u) = (1-u)\mathbf{p}_i^{r-1}(u) + u\mathbf{p}_{i+1}^{r-1}.$$

Se observă că:

- fiecare punct nou se obține prin interpolarea liniară a două puncte consecutive din iterația precedentă
- la fiecare iterație, numărul punctelor calculate se reduce cu unul

Interesant este faptul că ultimul punct care se obține, este $\mathbf{p}_0^n = \mathbf{r}(u)$ și este pe curbă!

Cititorul este invitat să verifice afirmația pentru $n=3$, folosind calculul recursiv indicat de algoritm. O ilustrare grafică a algoritmului Casteljau pentru $u=1/2$ se dă în figura 5.3.

Se poate demonstra că punctul final obținut divide curba Bézier în două curbe Bézier (*subdiviziune*) ale căror puncte de control sunt chiar $\mathbf{p}_0^0, \mathbf{p}_0^1, \mathbf{p}_0^2, \mathbf{p}_0^3$, respectiv $\mathbf{p}_3^3, \mathbf{p}_2^2, \mathbf{p}_1^1, \mathbf{p}_0^0$, ce au fost deja calculate. În consecință, este posibilă trasarea curbei folosind subdiviziunea recursivă. Costul de calcul pentru fiecare punct nou este de numai patru medii ponderate. Dacă se alege $u=1/2$, multiplicările pot fi evitate: împărțirea cu doi se poate implementa ca o operație de deplasare la dreapta cu o unitate (operatorul $\gg 1$ în limbajul C).

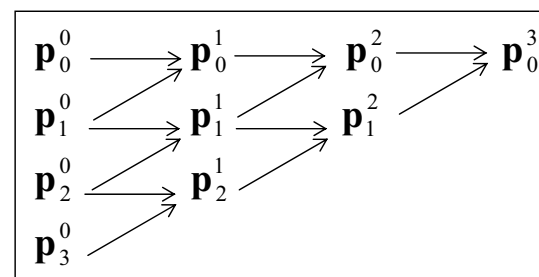
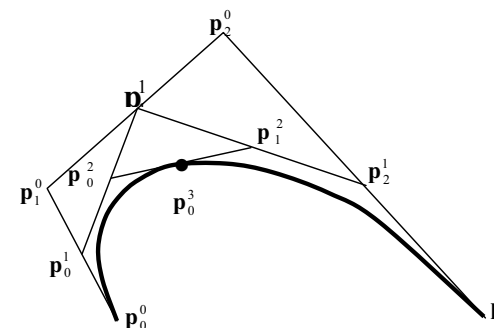


Fig. 5.3. Ilustrare a punctelor generate prin algoritmul Casteljau

Noile puncte de control ale celor două segmente se reduc la:

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{p}_0, & \mathbf{r}_0 &= \mathbf{q}_3, \\ \mathbf{q}_1 &= \frac{\mathbf{p}_0 + \mathbf{p}_1}{2}, & \mathbf{r}_1 &= \frac{\mathbf{p}_1 + \mathbf{p}_2}{4} + \frac{\mathbf{r}_2}{2}, \\ \mathbf{q}_2 &= \frac{\mathbf{q}_1 + \mathbf{p}_1 + \mathbf{p}_2}{4}, & \mathbf{r}_2 &= \frac{\mathbf{p}_2 + \mathbf{p}_3}{2}, \\ \mathbf{q}_3 &= \frac{\mathbf{q}_2 + \mathbf{r}_1}{2}, & \mathbf{r}_3 &= \mathbf{p}_3. \end{aligned}$$

Soluția este extrem de convenabilă și pentru o eventuală implementare hardware. Algoritmul se termină când învelișul convex al poligonului

10. Definiți și exemplificați utilizarea algoritmului Casteljau.

5.4 Algoritmul Casteljau

2/2

de control Bézier este suficient de plat pentru a se putea aproxima segmentul curent de curbă cu o dreaptă, care se trasează.

Algoritmul Casteljau este un punct forte major al reprezentării Bézier a curbelor. El are însă și o importanță teoretică. Pentru că interpolarea liniară este invariantă la transformarea afină, algoritmul demonstrează faptul că *reprezentarea Bézier are proprietatea invarianței la transformări affine.*

5.5. Formularea matricială a curbelor Bézier

O convenție alternativă pentru a specifica o curbă Bézier este cu ajutorul matricii Bézier și a vectorilor bazei de putere:

$$\mathbf{r}(u) = \mathbf{U} \mathbf{M}_B \mathbf{P}_c = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}.$$

De precizat, că vectorii \mathbf{r} și \mathbf{p} sunt vectori-linie în formularea de mai sus. Funcțiile de bază Bézier cubice se obțin din produsul primelor două matrici în ecuația de mai sus.

Formularea matricială a curbelor Bézier este utilă în următoarele două situații:

- implementarea hardware la reprezentarea în altă bază
- conversia între baze

TRANSMISII TELEFONICE

ANUL 3, SEMESTRUL 6

1. Transmisia pe 2 fire – principiu, condiții de stabilitate

[1] pag. 19-20

Capitolul 1

Noțiuni de bază în transmisia telefonică

- Câte porți are un circuit 2F / 4F ? Ce atenuări prezintă pe sensurile posibile ?
- Care este condiția de stabilitate în buclă ?
- Cum se asigură stabilitatea sistemului pe 4 fire ?



1.5. Transmisia pe 2 fire

Într-un sistem de transmisie pe 2 fire, fiecare sens de transmisie dispune doar de repezoare (terminale și intermediare) proprii. Linia de transmisie este utilizată în comun, pe toată lungimea transmisiei, de cele două sensuri ale comunicației.

În aceste condiții, cele două sensuri ale legăturii telefonice trebuie separate în fiecare repetor (terminal și intermediar), conform schemei de principiu din fig. 1.6.

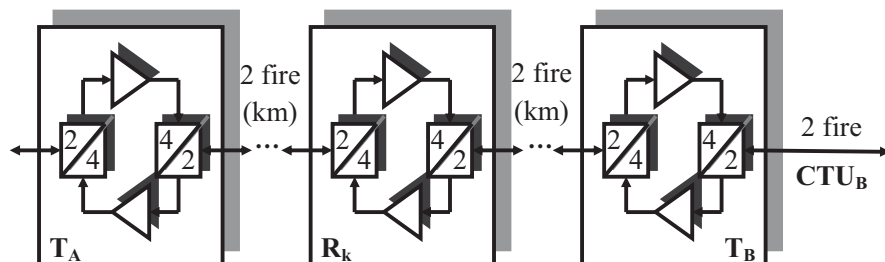


Fig. 1.6. Sistem de transmisie pe 2 fire.

Un sistem de transmisie pe 2 fire se caracterizează prin:

- repezoare (R_k) și terminale (T_A , T_B) **complexe**, ce conțin, pe lângă amplificatoarele necesare refacerii semnalelor pe fiecare sens, câte 2 circuite de trecere de la 2 fire la 4 fire,
- **consum** redus de linie de transmisie (fiind pe 2 fire).

Avantajele și dezavantajele evidențiază faptul că, din punct de vedere economic, soluția este **comparabilă** cu cea pe 4 fire, ambele fiind utilizate, opțiunea pentru una sau alta fiind determinată de criteriile suplimentare.

Principala deosebire este dată de condițiile de stabilitate ale sistemului. În **fiecare repetor** (terminal și intermediar) se formează câte o **buclă**, conform reprezentării din fig. 1.7.

Capitolul 1

Noțiuni de bază în transmisia telefonică

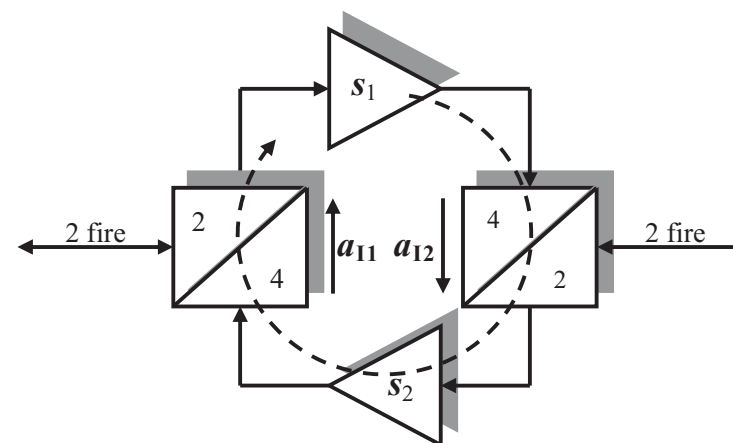


Fig. 1.7. Studiul stabilității unui repetor pe 2 fire.

Bucla conține 2 amplificări, s_1 și s_2 , și 2 atenuări, a_{11} și a_{12} , pe sensurile interzise ale circuitelor 2F / 4F. Pentru stabilitatea buclei, în fiecare repetor, trebuie îndeplinită condiția:

$$a_{11} + a_{12} > s_1 + s_2 . \quad (1.8)$$

Pentru k repezoare intermediare, conform exemplului din fig. 1.6, rezultă $k + 2$ condiții de îndeplinit. **Fiecare repetor** trebuie reglat, prin reducerea uneia din amplificări, astfel ca nici unul să nu oscileze. În cazul apariției instabilității, intervenția nu poate fi efectuată din terminal, ci trebuie identificat și reglat repetorul, în poziția geografică în care se află.

În plus, **orice grup de repezoare consecutive**, cu 2, 3, ..., $k + 1$ repezoare, formează câte o buclă a cărei stabilitate trebuie asigurată. Se poate demonstra că un grup de repezoare poate oscila, chiar dacă fiecare repetor, individual, este stabil.

În **concluzie**, asigurarea stabilității sistemelor de transmisie pe 2 fire este mai sensibilă decât a celor pe 4 fire.

- **Comparați** caracteristicile sistemelor 2 fire și 4 fire.
- **Scrieți** condițiile de stabilitate pentru un sistem de transmisie pe 2 fire, fără repezoare intermediare.
- **Calculați** numărul condițiilor de stabilitate pentru $k=8$.



2. Repetorul în sistemele de transmisie a semnalului vocal de telefonie

[1] pag. 21-22

Capitolul 1

Noțiuni de bază în transmisia telefonică

1.6. Repetorul

Semnalul transmis pe un canal de comunicații suferă mai multe **deformări**:

- **atenuare**, reprezentată prin scăderea nivelului semnalului la recepție, față de emisie; atenuarea depinde atât de tipul canalului folosit, cât și de distanța dintre repezoare;
- **distorsiuni liniare** (distorsiuni de frecvență), reprezentând deformarea semnalului datorită atenuării diferite a frecvențelor din spectrul transmis; fiecare canal de comunicații are o caracteristică de frecvență cunoscută;
- **distorsiuni neliniare** (distorsiuni de amplitudine), reprezentând deformarea semnalului datorită atenuării diferite a nivelurilor semnalului transmis; această caracteristică este mai puțin controlată;
- **zgomote**, perturbații, diafonie, reprezentând factori externi, ce nu pot fi, în general, controlați.

În cazul transmisiilor la **mare distanță**, la intervale bine precizate, semnalul trebuie refăcut, cât mai aproape de semnalul original. Se folosesc, în acest scop, **repezoare**. Acestea sunt *echipamente de linie*, care:

- în cazul transmisiilor **analogice**, realizează întreaga funcție de refacere a semnalului,
- în cazul transmisiilor **numerice**, reprezintă un bloc de prelucrare analogică în cadrul unui regenerator, ce refacem semnalul numeric.

Funcțiile minimale ale repetorului rezultă:

- **amplificare**, pentru compensarea atenuării introdusă de canalul de comunicație;
- **egalizare**, pentru compensarea caracteristicii de frecvență a mediului de transmisie.

Nu este așteptată compensarea distorsiunilor neliniare și a zgomotelor.

Repetorul se realizează întotdeauna pe 4 fire (chiar și când este conectat într-un sistem de transmisie pe 2 fire, prin circuite 2F / 4F), conform schemei bloc din fig. 1.8.

Capitolul 1

Noțiuni de bază în transmisia telefonică

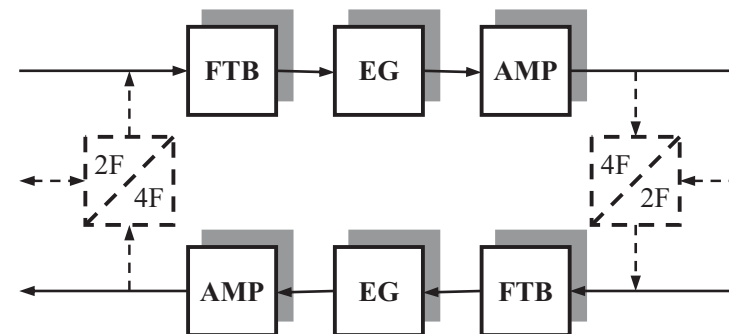


Fig. 1.8. Structura repetorului.

Filtrul trece bandă, FTB, limitează funcționarea repetorului la banda de frecvență utilă (0,3 ÷ 3,4 kHz pentru banda vocală sau o bandă mai largă în cazul transmisiei cu multiplexarea mai multor căi telefonice), în scopul evitării perturbațiilor sau instabilității din afara acestei benzi.

Egalizorul, EG, este un circuit pasiv care introduce o atenuare mai mare la frecvențele care au fost mai puțin atenuate de canalul de transmisie, astfel încât ansamblul *canal + egalizor* să aibă o caracteristică plată de frecvență.

Amplificatorul, AMP, este un circuit activ ce trebuie să compenseze atenuările cumulate pe o secțiune de amplificare (*canal + circuite 2F / 4F*, dacă este cazul + *FTB + EG*), astfel încât nivelul de emisie să fie identic cu cel de la echipamentul terminal. În plus, pentru reglajul stabilității, amplificarea trebuie să fie ajustabilă.

În locul oarecare în care se găsește repetorul, pe traseul cablului de comunicație, nu este disponibilă o sursă de alimentare. Ea se preia de la distanță, pe firele de comunicație, prin **telealimentare**. De aceea, se recomandă proiectarea circuitului cu o singură tensiune de alimentare, de valoare mică.

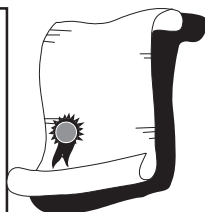


- Ce parametri ai semnalului sunt afectați la transmisie ?
- De ce se construiește repetorul exclusiv pe 4 fire ?
- Ce se înțelege prin telealimentare ?

2. TRANSMISIA CU MULTIPLEXARE ÎN FRECVENȚĂ

Subiecte

- 2.1. Principiul transmisiei cu multiplexare în frecvență
- 2.2. Translația de frecvență
- 2.3. Multiplexarea și demultiplexarea
- 2.4. Filtrarea căilor
- 2.5. Grupurile primare de bază
- 2.6. Formarea unui grup primar
- 2.7. Ierarhia transmisiei cu multiplexare în frecvență



Evaluare: 1. Răspunsuri la întrebările și problemele finale
2. Discuție pe tema: "Structura, parametrii și formarea grupului primar B"

2.1. Principiul transmisiei cu multiplexare în frecvență

În cazul transmisiei telefonice la distanțe mari (interurbane), un număr important de legături telefonice simultane se stabilesc pe un anumit **traseu comun**. Este posibil, în acest caz, ca mai multe căi telefonice să fie transmise pe un **același suport** (cablu metalic, canal radio etc.).

Cum o cale telefonică ocupă o bandă de frecvență îngustă ($0,3 \pm 3,4$ kHz), iar canalele de comunicație asigură benzi de frecvență de ordinul sutelor de kilohertzi sau megahertzi, pentru **utilizarea cât mai eficientă** a suportului de transmisie, este necesară transmiterea unui număr mare de căi telefonice pe același canal.

Tehnica, numită **multiplexare**, poate fi realizată printr-una din următoarele metode:

- multiplexarea în frecvență sau
- multiplexarea în timp (cap. 4).

În acest capitol vom analiza tehnica multiplexării în frecvență, specifică **telefoniei analogice**.

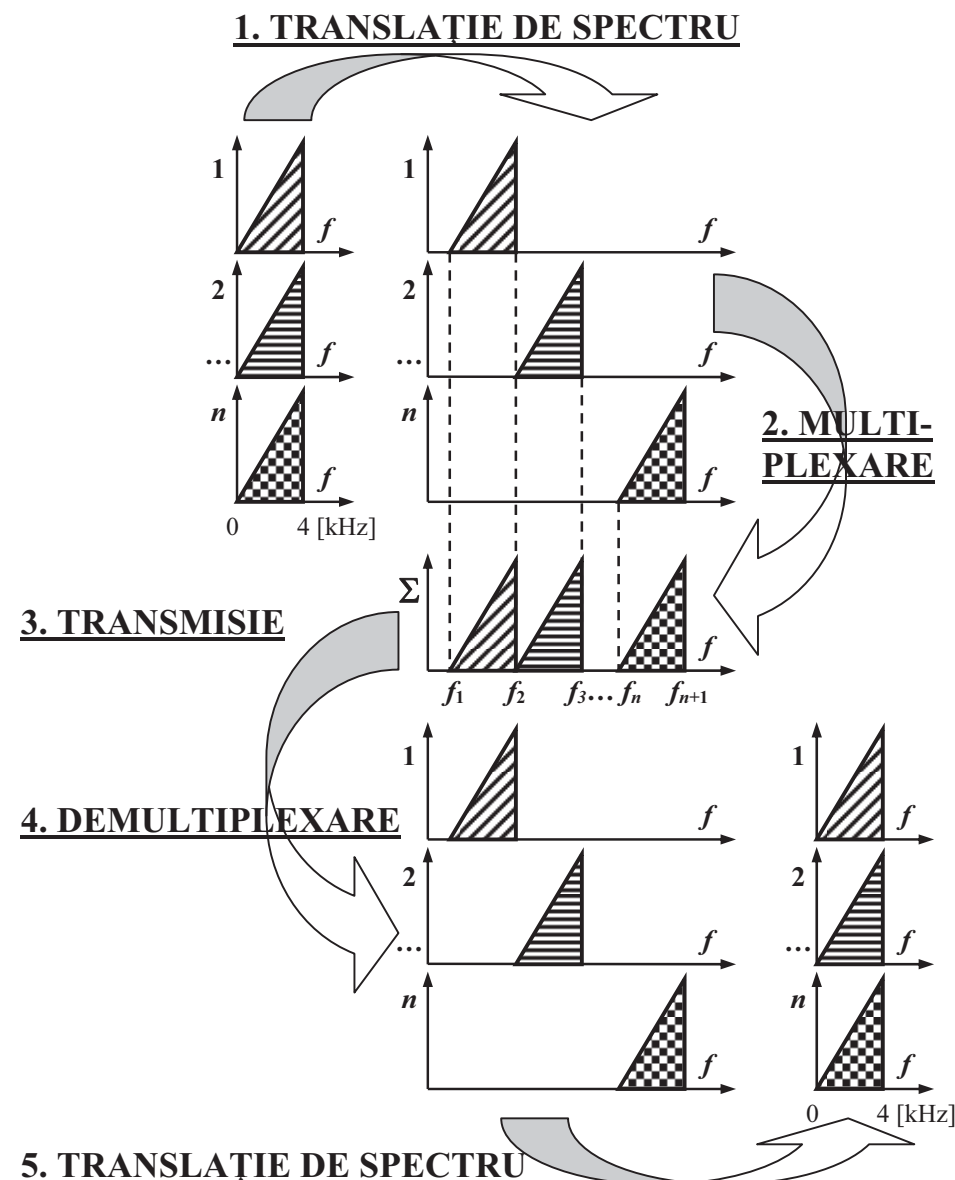


Fig. 2.1. Principiul multiplexării în frecvență.

3. Principiul transmisiei cu multiplexare în frecvență

[1] pag. 25-27

2/2

Capitolul 2 Transmisia cu multiplexare în frecvență

Având în vedere că toate căile telefonice pe care dorim să le multiplexăm ocupă **aceeași bandă** de frecvență ($0,3 \div 3,4$ kHz), ele nu pot fi însumate direct. Ele s-ar perturba reciproc, și-ar pierde identitatea și nu ar mai putea fi extrase din semnalul rezultat. De aceea, multiplexarea în frecvență (reprezentată în fig. 2.1) presupune o **prelucrare a semnalelor**, atât la emisie, cât și la recepție.

La **emisie** se realizează:

- **translația de spectru** (deplasarea în frecvență) cu o frecvență specifică fiecărei căi, ceea ce permite plasarea căilor ce trebuie multiplexate în benzi de frecvență diferite, disjuncte, care să nu se intercaleze prin însumarea semnalelor; operația se realizează prin **modulare**;
- **multiplexarea** (însumarea semnalelor) printr-un circuit care să împiedice influențarea reciprocă a căilor, numit decuplor de căi.

Semnalul astfel rezultat (prin alăturarea spectrelor individuale) conține informația corespunzătoare tuturor căilor. Nu mai avem de a face cu n semnale individuale, ci cu **un singur semnal** multiplexat, care **se transmite** prin canalul de comunicație.

La **recepție**, din semnalul unic, trebuie extrase cele n semnale individuale, fără a fi afectate de procedura de transmisie. Operațiile sunt inverse celor de la emisie:

- **demultiplexarea** (selectarea și extragerea benzilor de frecvență corespunzătoare semnalelor individuale) ce se realizează prin **filtre trece bandă** (cu frecvența specifică fiecărei căi) extrem de precise;
- **translația de spectru** în banda de bază, cu o frecvență proprie căii, prin **demodulare**; atât modularea, cât și demodularea, se realizează cu același tip de circuit: modulator.

- De ce este necesară transmisia prin multiplexare ?
- Ce tehnici de multiplexare pot fi folosite ?
- De ce este necesară translația de frecvență ?
- Cum se realizează demultiplexarea ?
- Ce funcții realizează un modulator ?



4. Principiul transmisiei cu multiplexare în timp

[1] pag. 48-51

Capitolul 3 Discretizarea semnalului vocal

adică:

$$f_E > 2 \times f_M. \quad (3.2)$$

În **concluzie**, condițiile pentru reconstruirea fără distorsiuni a semnalului transmis prin eșantioanele sale sunt :

- eșantionarea cu o frecvență f_E mai mare decât **dublul** frecvenței maxime f_M a semnalului ce trebuie transmis (ceea ce impune, în cazul semnalului vocal, limitarea semnalului la 3,4 kHz, pentru a evita suprapunerea pachetelor spectrale vecine),
- utilizarea unui **filtru trece jos** cu caracteristică plată în banda de trecere (300 ÷ 3400 Hz) și cu un flanc suficient de abrupt pentru separarea pachetelor spectrale (intervalul $f_M \div f_E - f_M$).

Pentru ca filtrul trece jos să poată fi realizat fizic, se recomandă alegerea unei frecvențe de eșantionare mai mare decât limita teoretică dată de teorema eșantionării:

$$f_E = (1,1 \div 1,3) \times 2 \times f_M, \quad (3.3)$$

ceea ce, în cazul semnalului telefonic, înseamnă:

$$f_E = (1,1 \div 1,3) \times 2 \times 3,4 \text{ kHz} = 7,48 \div 8,84 \text{ kHz}. \quad (3.4)$$

Valoarea **standardizată** în telefonia numerică este:

$$f_E = 8 \text{ kHz}. \quad (3.5)$$



- **Calculați perioada de eșantionare a semnalului vocal.**
- **Calculați intervalul de frecvență disponibil flancului FTJ la recepție.**

3.3. Multiplexarea în timp

Transmiterea unui semnal prin eșantioanele sale nu ocupă complet (în timp) canalul de comunicație. Se observă, în fig. 3.1, că:

$$\tau \ll T_E (= 1/f_E = 125 \mu\text{s}), \quad (3.6)$$

Capitolul 3 Discretizarea semnalului vocal

adică majoritatea timpului canalul este **liber**, ceea ce permite transmiterea, între eșantioanele semnalului analizat, și a altor eșantioane, ale altor semnale.

Prin transmiterea simultană a mai multor semnale eșantionate prin același canal de comunicație, se realizează **multiplexarea în timp**. Este de observat că multiplexarea în timp se poate realiza cu semnale MIA, cuantizarea semnalului nefiind o condiție necesară.

Multiplexarea în timp a impulsurilor MIA presupune *intercalarea* lor pentru transmisia pe un canal comun. Aceasta impune eșantionarea semnalelor de transmis în momente de timp diferite:

- pentru a nu se suprapune eșantioanele a două semnale diferite,
- pentru a “umple” complet intervalul liber dintre eșantioanele succesive ale aceluiași semnal.

Pentru un sistem multiplex cu n căi trebuie să generăm n semnale de eșantionare, E_1, E_2, \dots, E_n , cu **faza impulsurilor decalată** astfel ca să se asigure cele două condiții de mai sus. Fig. 3.3 prezintă diagramele de timp ale semnalelor de eșantionare pentru realizarea unui semnal multiplex cu 3 căi.

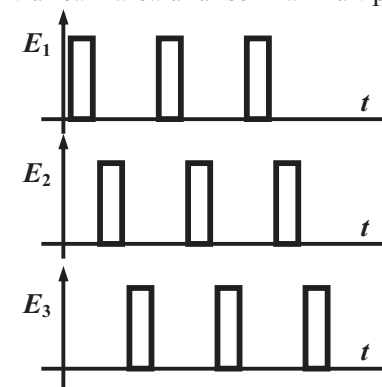


Fig.3.3. Semnale de eșantionare pentru multiplexarea în timp

Prin utilizarea semnalelor de eșantionare E_1, E_2 și E_3 pentru eșantionarea semnalelor de transmis $S_1(t), S_2(t)$ și, respectiv, $S_3(t)$, eșantioanele acestora pot fi transmise intercalate în timp pentru formarea **semnalului multiplex**.

4. Principiul transmisiei cu multiplexare în timp

[1] pag. 48-51

2/2

Capitolul 3 Discretizarea semnalului vocal

În exemplul din fig. 3.4, cele trei semnale sunt: $S_1(t)$ – constant, $S_2(t)$ – liniar crescător, iar $S_3(t)$ – liniar descrescător. La ieșirea blocului **EMISIE** există **un singur semnal**, care conține informația celor trei semnale inițiale (cu eșantioanele multiplexate în timp).

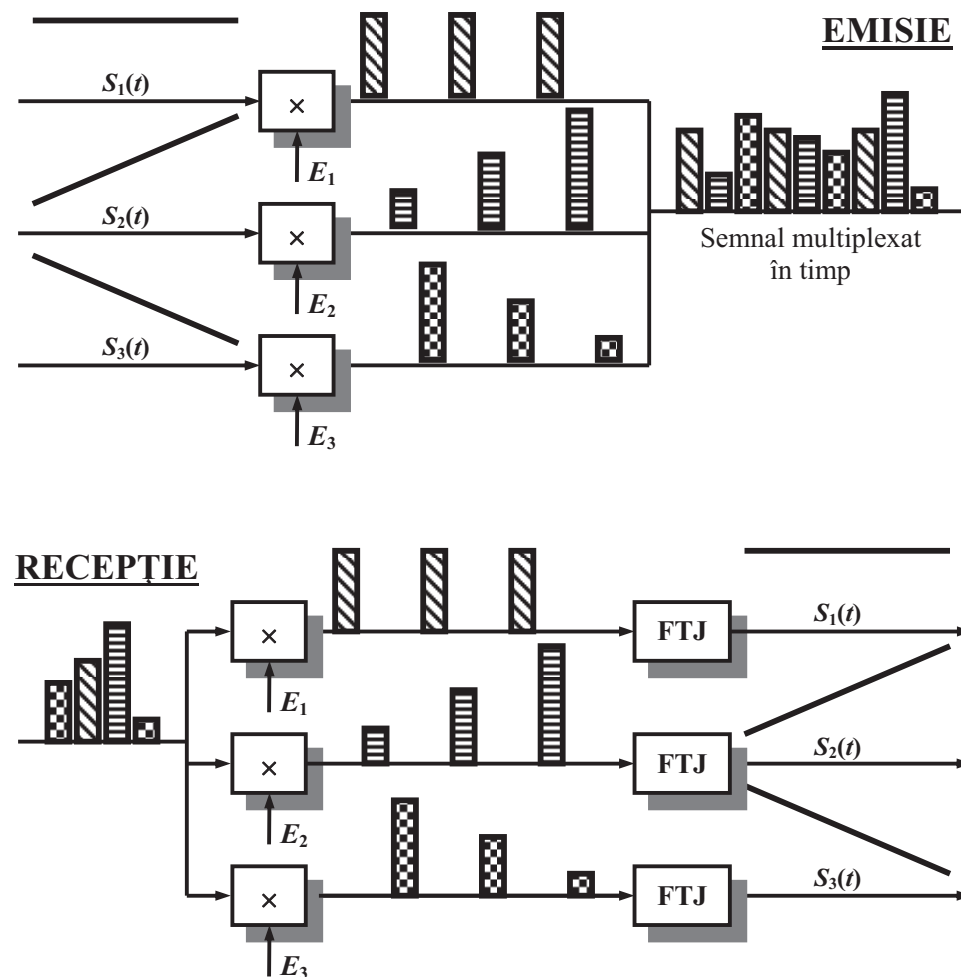


Fig.3.4. Principiul transmiterii semnalelor multiplexate în timp.

5. Parametrii discretizării semnalului vocal de telefonie – banda de frecvență, frecvența de eșantionare, numărul de biți/eșantion, debit

[1] pag. 45, 48, 61

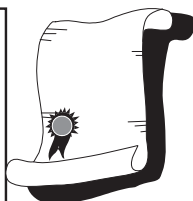
1/2

Capitolul 3 Discretizarea semnalului vocal

3. DISCRETIZAREA SEMNALULUI VOCAL

Subiecte

- 3.1. Definierea noțiunii de discretizare
- 3.2. Eșantionarea
- 3.3. Multiplexarea în timp
- 3.4. Cuantizarea uniformă
- 3.5. Cuantizarea neuniformă
- 3.6. Compandarea
- 3.7. Legea de compresie A



Evaluare: 1. Răspunsuri la întrebările și problemele finale
2. Discuție pe tema: “Prelucrarea semnalului vocal pentru transmisia numerică”

3.1. Definierea noțiunii de discretizare

Semnalul vocal (tensiune sau curent) este un semnal analogic:

- are o **variație continuă în timp**, adică este caracterizat printr-o valoare (nivel) în orice moment;
- are o **variație continuă în nivel**, adică poate lua orice valoare (dintr-un interval de lucru).

În telefonia analogică, semnalul vocal este definit, ca semnalul limitat la **banda de frecvență 300 ÷ 3400 Hz**, ce se transmite în rețea. Pentru telefonia numerică este acceptată **aceeași limitare** a benzii de frecvență.

Prin discretizarea unui semnal analogic se înțelege înlocuirea acestuia cu un set finit de valori (numere) într-un număr finit de momente de timp.

Discretizarea semnalului se realizează, de aceea, în doi pași :

- discretizarea în timp, prin **eșantionare**,
- discretizarea în nivel, prin **cuantizare**.

Capitolul 3 Discretizarea semnalului vocal

adică:

$$f_E > 2 \times f_M. \quad (3.2)$$

În **concluzie**, condițiile pentru reconstruirea fără distorsiuni a semnalului transmis prin eșantioanele sale sunt :

- eșantionarea cu o frecvență f_E mai mare decât **dublul** frecvenței maxime f_M a semnalului ce trebuie transmis (ceea ce impune, în cazul semnalului vocal, limitarea semnalului la 3,4 kHz, pentru a evita suprapunerea pachetelor spectrale vecine),
- utilizarea unui **filtru trece jos** cu caracteristică plată în banda de trecere (300 ÷ 3400 Hz) și cu un flanc suficient de abrupt pentru separarea pachetelor spectrale (intervalul $f_M \div f_E - f_M$).

Pentru ca filtrul trece jos să poată fi realizat fizic, se recomandă alegerea unei frecvențe de eșantionare mai mare decât limita teoretică dată de teorema eșantionării:

$$f_E = (1,1 \div 1,3) \times 2 \times f_M, \quad (3.3)$$

ceea ce, în cazul semnalului telefonic, înseamnă:

$$f_E = (1,1 \div 1,3) \times 2 \times 3,4 \text{ kHz} = 7,48 \div 8,84 \text{ kHz}. \quad (3.4)$$

Valoarea **standardizată** în telefonia numerică este:

$$f_E = 8 \text{ kHz}. \quad (3.5)$$



- **Calculați perioada de eșantionare a semnalului vocal.**
- **Calculați intervalul de frecvență disponibil flancului FTJ la recepție.**

3.3. Multiplexarea în timp

Transmiterea unui semnal prin eșantioanele sale nu ocupă complet (în timp) canalul de comunicație. Se observă, în fig. 3.1, că:

$$\tau \ll T_E (= 1/f_E = 125 \mu\text{s}), \quad (3.6)$$

5. Parametrii discretizării semnalului vocal de telefonie – banda de frecvență, frecvența de eșantionare, numărul de biți/eșantion, debit

[1] pag. 45, 48, 61

2/2

Capitolul 3 Discretizarea semnalului vocal

Pe **axa y** , cele 256 de trepte, corespunzătoare celor 8 biți, sunt egale, conducând la compresia logaritmică A . Excepție de la funcția logaritmică face segmentul 0, care are aceeași pantă cu segmentul 1, conform detaliului din fig. 3.10.

În **concluzie**, în telefonia numerică, cu multiplexare în timp, se folosesc semnale numite MIC (cu *modulația impulsurilor în cod*) sau PCM (*Pulse Code Modulation*), obținute prin:

- eșantionare cu $f_E = 8$ kHz,
- cuantizare (echivalentă) pe 12 biți,
- compresie logaritmică pe $n = 8$ biți.

În aceste condiții, **debitul unei căi vocale numerice** rezultă:

$$D = f_E \times n = 8 \cdot 10^3 \times 8 = 64 \text{ kbiți/s.} \quad (3.8)$$

6. Cuantizarea neuniformă a semnalului vocal

[1] pag. 54-55

Capitolul 3 Discretizarea semnalului vocal

În cazul transmiterii unui **semnal audio**, trebuie ținut seama de **dinamica mare** a acestuia (raportul dintre cel mai puternic sunet transmis și cel mai slab sunet perceput), care impune cuantizarea pe un număr mare de trepte (16 biți pentru muzică). Pentru **semnalul telefonic** se acceptă **12 biți**, respectiv $2^{12} = 4096$ de trepte de cuantizare.

Este de remarcat faptul că ceea ce se transmite sunt niște numere, care aproximează nivelul fiecărui eșantion. La recepție, aceste numere sunt convertite în semnal analogic, care este afectat de aceste erori și, deci, nu mai este identic cu cel de la emisie. De aceea se pune problema reducerii **erorii relative** de cuantizare (raportul semnal / zgomot de cuantizare), prin utilizarea unei alte caracteristici de cuantizare.



- Care sunt dezavantajele cuantizării uniforme ?
- Care este numărul de trepte de cuantizare pentru un semnal reprezentat pe 16 biți ?
- Ce este zgomotul de cuantizare ?

3.5. Cuantizarea neuniformă

Cuantizarea neuniformă își propune realizarea unui raport **semnal / zgomot de cuantizare constant**, independent de nivelul semnalului de transmis. Aceasta conduce la o calitate constantă a transmisiei, evidențiind avantajul cuantizării neuniforme față de cea uniformă.

În acest scop, semnalul de nivel mic trebuie cuantizat cu o **treaptă mai fină**, iar semnalul de nivel mare poate fi cuantizat cu o **treaptă mai brută**. Caracteristica de cuantizare, reprezentată în fig. 3.7, evidențiază pașii de cuantizare inegali, ceea ce conduce la o eroare de cuantizare variabilă cu nivelul semnalului.

În comparație cu cuantizarea uniformă, cuantizarea neuniformă prezintă

- avantajul unei **calități** a transmisiei independentă de nivelul semnalului,
- păstrând **domeniul** acestuia (analog sau numeric).

Capitolul 3 Discretizarea semnalului vocal

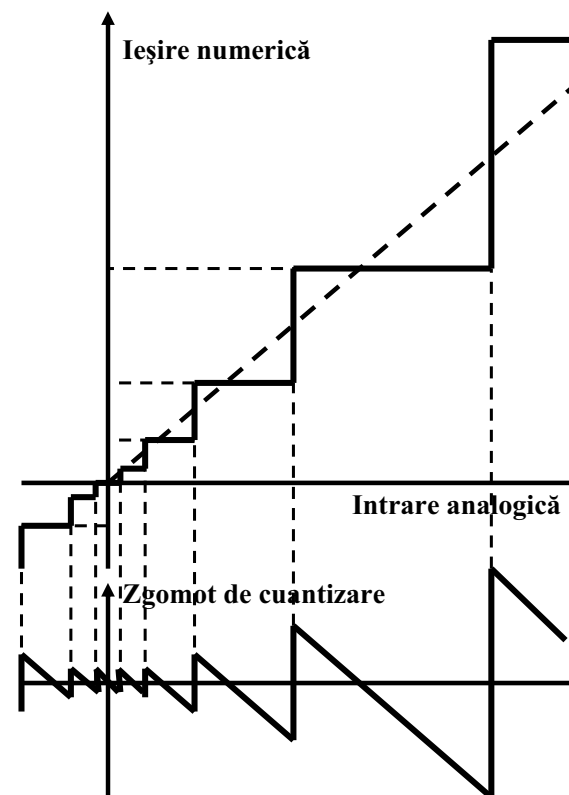


Fig. 3.7. Caracteristica de cuantizare neuniformă și zgomotul de cuantizare.

- Comparați numărul treptelor la cuantizarea uniformă, respectiv, neuniformă, păstrând domeniul și rezoluția (treapta minimă).



7. Legea de compresie A – principiul compresiei, aproximarea prin segmente de dreaptă a caracteristicii de compresie, semnificația biților din semnalul comprimat

[1] pag. 58-61

1/2

Capitolul 3 Discretizarea semnalului vocal

Totuși, la recepție, semnalul refăcut nu ar putea fi folosit, fiind diferit de cel inițial. De aceea, la recepție trebuie efectuată prelucrarea inversă a semnalului transmis: **expandarea**, ce constă în atenuarea nivelurilor mici și amplificarea nivelurilor mari.

Transmisia cu **compresie** la emisie și **expandare** la recepție se numește **compandare**. Principiul acestei prelucrări este prezentat în fig. 3.9.

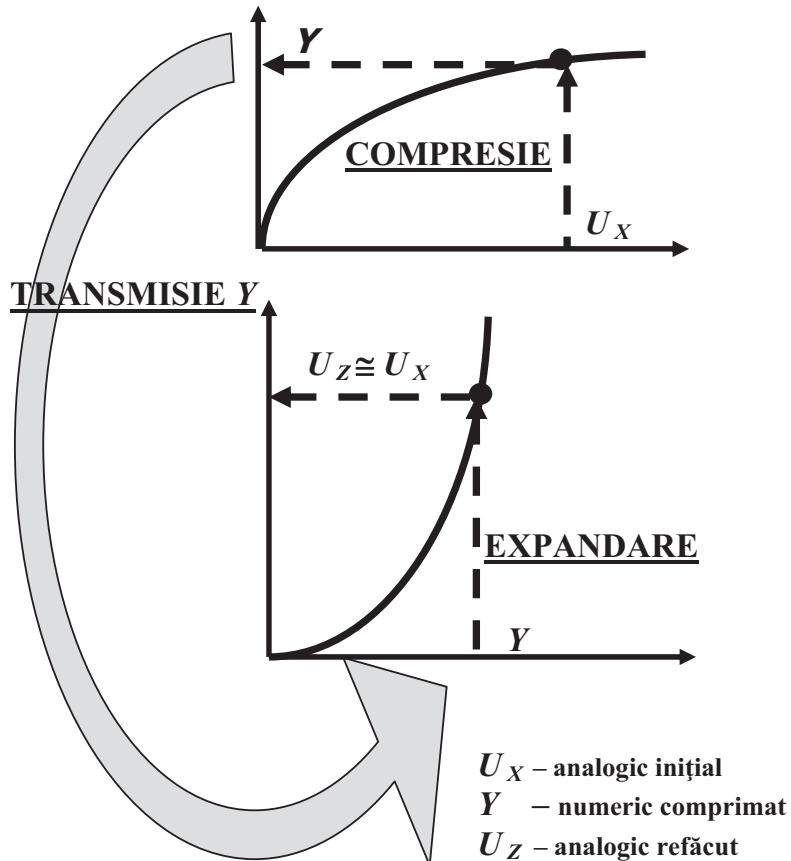


Fig. 3.9. Principiul transmisiei cu compandare.

Capitolul 3 Discretizarea semnalului vocal

- Pe ce tip de cuantizare se bazează compresia semnalului ?
- Care sunt avantajele compresiei logaritmice ?
- Ce operații presupune compandarea ?



3.7. Legea de compresie A

Pentru utilizare într-o rețea publică, prelucrarea semnalului trebuie să respecte aceleași reguli în toate echipamentele care o compun. Din acest motiv, compandarea este **standardizată**:

- legea de compandare A, utilizată în Europa,
- legea de compandare μ , utilizată în S.U.A., Japonia.

Legea A, ca funcție $y(x)$, este definită prin relații matematice. În practică se utilizează, însă, o **aproximare** a legii A, compatibilă cu sistemele numerice de prelucrare și transmisie. Astfel, curba logaritmică este aproximată prin 8 segmente de dreaptă, inegale, iar fiecare segment (liniar) este format din 16 intervale egale. Alura legii de compresie A aproximată prin segmente este prezentată în fig. 3.10.

Este de remarcat că semnalele transmise sunt de ambele polarități, motiv pentru care și legea A conține două cadrane: cadranul 1 pentru alternanța pozitivă, respectiv cadranul 3 (simetric cu primul, dar nefigurat) pentru alternanța negativă.

Pe **axa x** este reprezentat semnalul de intrare (tensiune) analogic, necomprimat.

- Nivelurile de definire a segmentelor de dreaptă sunt date de puteri ale lui 2, corespunzătoare cuantizării neuniforme. Rezultă **8 segmente**, numerotate de la 0 la 7, ce pot fi reprezentate pe 3 biți: $l_1 l_2 l_3$.
- Fiecare segment conține **16 trepte** egale (în cadrul aceluiasi segment), conform detaliului mărit din fig. 3.10, corespunzătoare cuantizării uniforme. Cele 16 trepte sunt numerotate de a 0 la 15 și pot fi reprezentate pe 4 biți: $v_1 v_2 v_3 v_4$.

7. Legea de compresie A – principiul compresiei, aproximarea prin segmente de dreaptă a caracteristicii de compresie, semnificația biților din semnalul comprimat

[1] pag. 58-61

2/2

Capitolul 3

Discretizarea semnalului vocal

- Cele două cadrane simetrice, pentru semnale pozitive și negative, sunt identificate printr-un bit de semn: s .

Rezultă, deci, formatul semnalului numeric comprimat, conform aproximării prin segmente de dreaptă a legii A, pe 8 biți:

$$Y = s l_1 l_2 l_3 v_1 v_2 v_3 v_4. \quad (3.7)$$

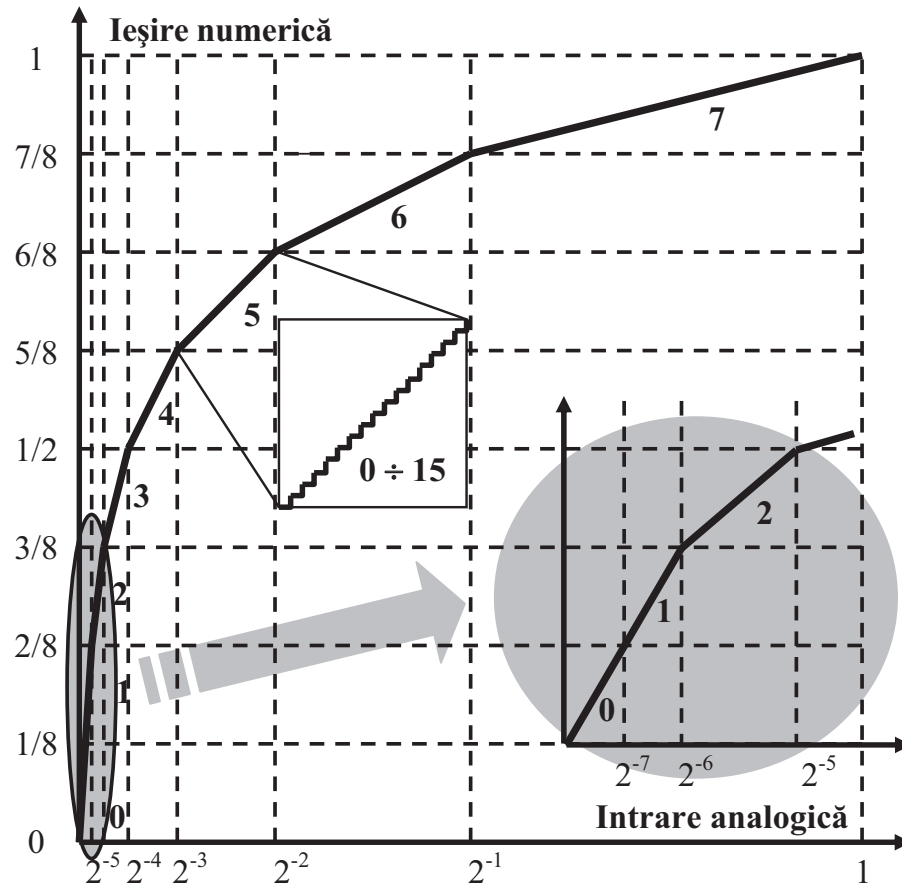


Fig. 3.10. Aproximarea prin segmente de dreaptă a legii de compresie A.

Capitolul 3

Discretizarea semnalului vocal

Pe axa y , cele 256 de trepte, corespunzătoare celor 8 biți, sunt egale, conducând la compresia logaritmică A. Excepție de la funcția logaritmică face segmentul 0, care are aceeași pantă cu segmentul 1, conform detaliului din fig. 3.10.

În **concluzie**, în telefonia numerică, cu multiplexare în timp, se folosesc semnale numite MIC (cu *modulația impulsurilor în cod*) sau PCM (*Pulse Code Modulation*), obținute prin:

- eșantionare cu $f_E = 8$ kHz,
- cuantizare (echivalentă) pe 12 biți,
- compresie logaritmică pe $n = 8$ biți.

În aceste condiții, **debitul unei căi vocale numerice** rezultă:

$$D = f_E \times n = 8.10^3 \times 8 = 64 \text{ kbiți/s.} \quad (3.8)$$

8. Convertorul analog-numeric– structură, funcționare

[1] pag. 68-69

4.4. Calea vocală

Forma numerică a semnalului vocal este dată de relația (3.7). Acest cod de 8 biți este obținut printr-o cuantizare cu compresie conform legii A. Ea se obține prin utilizarea unui convertor analog – numeric cu compresie logaritmică.

Structura acestuia este identică cu cea a unui convertor analog – numeric obișnuit. În fig. 4.4 este prezentată schema bloc a unui **convertor analog – numeric liniar**, cu aproximări succesive.

Funcționarea acestuia este reamintită prin exemplul din fig. 4.3. Tensiunea necunoscută U_x , aplicată la intrarea convertorului, este aproximată pas cu pas prin tensiunea $U_{CNA}(t)$, generată de convertorul numeric – analogic.

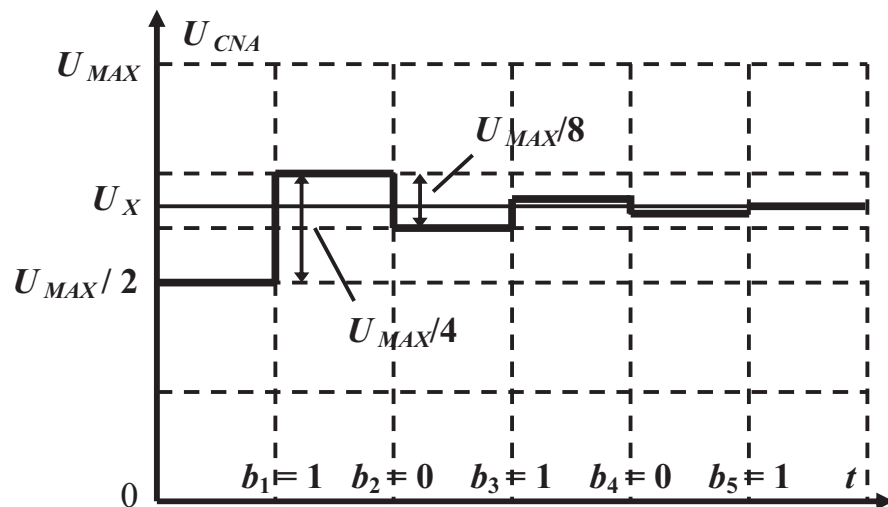


Fig.4.3. Principiul aproximării succesive cu un CAN liniar.

- În **pasul 1** se estimează bitul cel mai semnificativ, b_1 , prin generarea codului (din registrul de aproximări succesive) **1000.0000**. Codul reprezintă mijlocul domeniului, deci tensiunea de comparație generată este $U_{CNA} = U_{MAX}/2$. Prin compararea tensiunii de intrare U_x cu cea de referință,

U_{CNA} , se decide valoarea definitivă a bitului b_1 : dacă $U_x > U_{CNA}$, rezultă $b_1 = 1$, iar dacă $U_x < U_{CNA}$, atunci $b_1 = 0$.

- În **pasul 2** se estimează al doilea bit, b_2 , prin generarea codului **$b_1100.0000$** . Tensiunea de comparație se modifică cu un sfert din domeniu, în sensul aproximării tensiunii necunoscute: $U_{CNA} = b_1 \times U_{MAX}/2 + U_{MAX}/4$. Rezultatul comparației fixează valoarea bitului b_2 .
- În **pasul 3** se decide bitul b_3 , cu ajutorul codului **$b_1b_210.0000$** , care determină tensiunea: $U_{CNA} = b_1 \times U_{MAX}/2 + b_2 \times U_{MAX}/4 + U_{MAX}/8$
- ș.a.m.d.
- Numărul de pași de aproximare determină numărul de biți pe care se face cuantizarea. În fig. 4.3 sunt exemplificate valorile primilor 5 biți, pentru valoarea particulară a tensiunii U_x , reprezentate grafic.

Este de observat că dacă domeniul convertorului este axat pe zero (deci admite tensiuni de intrare pozitive și negative), în primul pas se stabilește **semnul semnalului**.

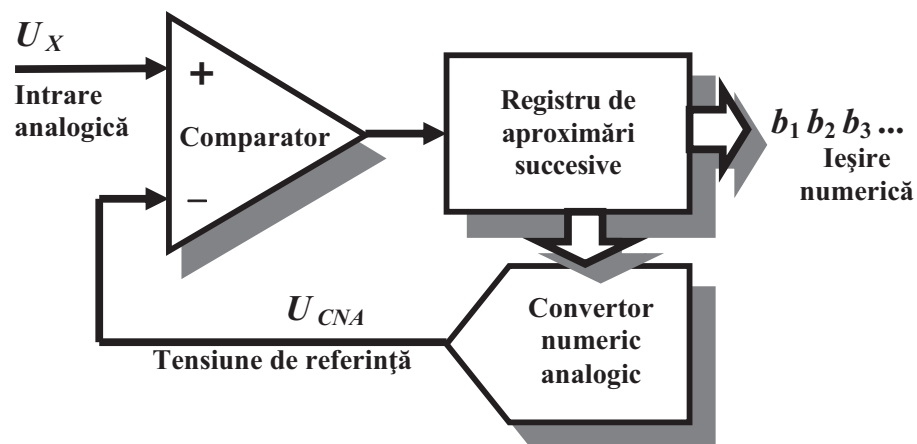


Fig.4.4. Schema bloc a unui CAN (liniar sau cu compresie).

9. Structura (numărul de biți și de intervale temporale) și parametrii (durate, frecvențe, debite) cadrului PCM

[1] pag. 66-67

Capitolul 4 Transmisia numerică PCM

4.2. Semnalele multiplexate

Într-o legătură telefonică trebuie transmise atât semnalul vocal de convorbire, cât și semnalizările necesare stabilirii legăturii și **interacțiunii utilizator – rețea**. În plus, în cazul unei transmisii numerice cu multiplexare în timp, este necesară asigurarea unei **referințe de timp** comune pentru emisie și recepție. În acest scop se transmite un semnal de sincronizare. Semnalele necesare a fi multiplexate, sunt, deci:

- semnale **vocale** – corespunzătoare căilor telefonice,
- **semnalizări** – asociate fiecărei căi telefonice,
- semnale de **sincronizare** a recepției cu emisia.

Nu există un standard unic pentru **multiplexul PCM**. În acest manual vom trata doar **standardul european**, care prevede pentru multiplexul primar PCM, organizat în 32 de intervale de timp:

- 30 de căi telefonice,
- 1 interval pentru semnalizări,
- 1 interval pentru sincronizare.

Având în vedere că o cale vocală este cuantizată și comprimată pe 8 biți, celelalte intervale sunt organizate tot pe 8 biți, fiind adoptată, deci, tehnica **întrețeserii octet cu octet**.

Aceasta prezintă avantajul că fiecare interval (al unei căi telefonice) are o **semnificație fizică**: reprezintă codul PCM al unui eșantion (spre deosebire de tehnica întrețeserii bit cu bit, unde semnificația eșantionului s-ar pierde).

În acest mod, multiplexul primar PCM devine un multiplex de **eșantioane**, ce pot fi manipulate individual (de exemplu, plasate în altă ordine temporală). Acest avantaj permite comutația temporală a căilor telefonice, ce stă la baza comutației numerice în centralele telefonice digitale (cap.5).

În **concluzie**, multiplexul primar PCM este un flux de date folosit în **transmisie și comutație**, în **telefonie numerică**.

- De ce este necesar un semnal de sincronizare ?
- Care sunt avantajele multiplexării octet cu octet ?
- Ce format numeric are eșantionului codat PCM ?

Capitolul 4 Transmisia numerică PCM

4.3. Cadrul primar PCM

Fluxul de date este organizat, în timp, în **cadre**. Un cadru conține câte un eșantion (un octet) din fiecare semnal multiplexat, având, deci, frecvența:

$$f_{CD} = f_E = 8 \text{ kHz}, \quad (4.1)$$

respectiv o durată $t_{CD} = 125 \mu\text{s}$.

Cadrul primar PCM, reprezentat în fig. 4.2 (cu detalii la scări extinse de timp) conține 32 de intervale, fiecare cu durata:

$$t_{INT} = \frac{t_{CD}}{32} = \frac{125}{32} = 3,9 \mu\text{s}, \quad (4.2)$$

numerotate de la 0 la 31:

- 0 - codul pentru **sincronizarea** cadrului,
- 1 ÷ 15 - căile telefonice numerotate de la 1 la 15,
- 16 - **semnalizările** pentru căile de la 1 la 30,
- 17 ÷ 31 - căile telefonice numerotate de la 16 la 30.

Debitul multiplexului primar PCM rezultă :

$$D = 8 \text{ kHz} \times 32 \text{ intervale} \times 8 \text{ biți} = \mathbf{2048 \text{ kbiți / s.}} \quad (4.3)$$

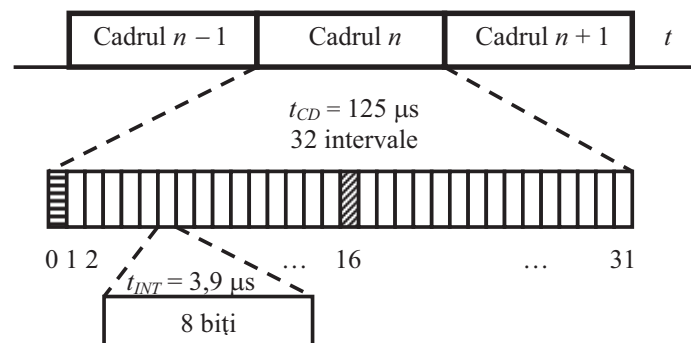


Fig.4.2. Structura și parametrii cadrului primar PCM.

10. Semnalizarea în cadrul PCM– multicadrul de semnalizări, [1] pag.75-76

4.6. Multicadrul de semnalizare

Într-o rețea telefonică automată, pe lângă informația vocală, care asigură comunicația directă a utilizatorilor, trebuie transmise și **semnalizări**, care să asigure funcțiile necesare stabilirii, menținerii și eliberării legăturilor între terminale.

Din structura cadrului PCM (fig. 4.2) se remarcă, pentru cele 30 de căi telefonice transmise, că există un **singur interval** (16) pentru semnalizări. Este evident că cei 8 biți ai intervalului de semnalizare nu pot acoperi necesitățile de semnalizare pentru 30 de căi.

De aceea sunt necesare mai multe cadre, fiecare cu intervalul său de semnalizări, grupate într-un **multicadru** de semnalizare. Această structură standardizată este formată din 16 cadre primare PCM, conform reprezentării din fig. 4.6.

Au fost alocați câte 4 biți de semnalizare (a b c d) fiecărei căi telefonice, ceea ce înseamnă că în intervalul 16 al unui cadru pot fi transmise semnalizările pentru 2 căi.

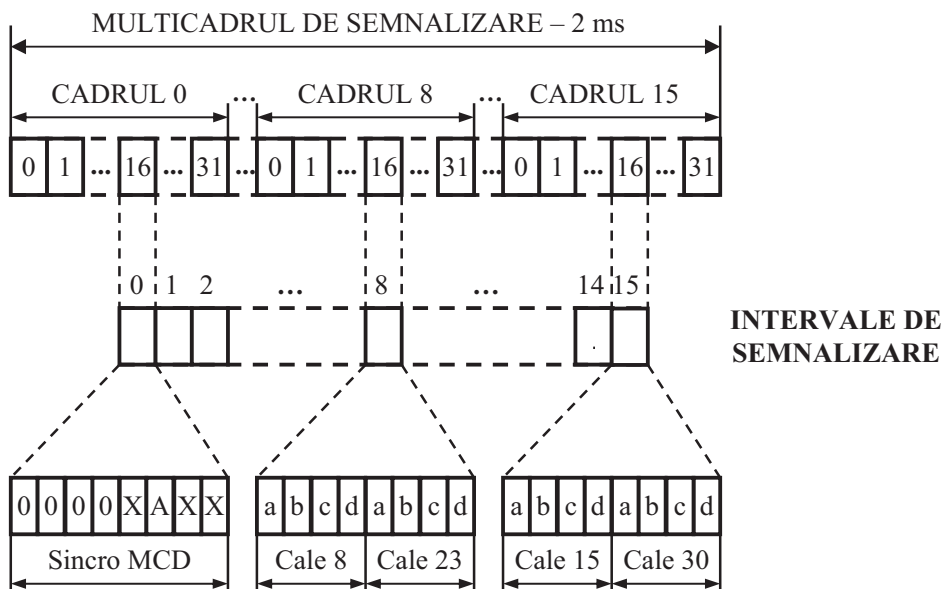


Fig.4.6. Organizarea multicadrului de semnalizare.

Pentru celelalte căi, semnalizările sunt transmise în cadrele următoare. Sunt necesare, deci, intervalele 16 din 15 cadre pentru semnalizările celor 30 de căi.

Există o legătură bine precizată (reprezentată și în fig. 4.6) între numărul căii (23), cadrul în care se transmite semnalizarea asociată (8) și poziția celor 4 biți de semnalizare (ultimii 4 biți ai intervalului 16). Pentru ca la recepție să poată fi contorizate cadrele (de la 1 la 15), este necesară transmiterea unei referințe de timp, care să marcheze care este primul cadru.

Se transmite, în acest scop, un cod de **sincronizare pe multicadru de semnalizări**, în intervalul 16 al cadrului 0. Rezultă, deci, 16 cadre primare într-un multicadru.

Cuvântul de sincronizare pe multicadru este o structură fixă de 4 biți: **0000**. Ceilalți 4 biți au semnificații asemănătoare cu cei analizați la sincronizarea pe cadru: **A** are rol de alarmă, în cazul pierderii sincronizării pe multicadru, iar biții **X** formează un canal de date disponibil.

Este de observat că **pierderea sincronismului** pe multicadru nu afectează sincronismul cadrelor, adică legăturile pot continua normal. De aceea, sincronizarea pe multicadru este mai puțin rigidă: cuvântul de sincronizare este scurt (4 biți), cu frecvență mai mică (la 16 cadre) și cu o procedură mai dinamică de schimbare a stării de funcționare (declarația pierderii sincronizării după 2 detecții eronate consecutive și declarația restabilirii sincronismului la prima detecție corectă).

Conform structurii de multicadru, semnalizările corespunzătoare unei căi au o frecvență mai mică decât a eșantioanelor vocale. Aceasta nu reprezintă o limitare, având în vedere că debitul acestora este mult mai mic și datorită faptului că întârzierile ce pot apărea sunt insesizabile pentru utilizator.



- De ce este necesară gruparea mai multor cadre ?
- De ce este necesară sincronizarea și pe multicadru ?
- Care este frecvența multicadrului de semnalizări ?
- Calculați debitul canalului de semnalizare al unei căi ?

BAZE DE DATE
ANUL 3, SEMESTRUL 6

1. Ce presupune normalizarea unei baze de date relationale si cum poate fi ea realizata? (3.2. – pag.16)

Comenzi

IdComanda	IdClient	Valoare	Data
1	2	134.67	2005-07-12
2	1	23.9	2006-01-20
3	3	150	2006-03-13
4	1	1234	2006-09-24

3.2. Proiectarea unei baze de date

Proiectarea unei baze de date este un proces foarte important care cuprinde urmatoorii pasi:

- analiza problemei;
- identificarea datelor;
- normalizarea.

Normalizarea unei baze de date consta in principal in descompunerea modelului bazei de date in mai multe relatii astfel incat sa se reduca la maxim redundanta datelor si implicit sa elimine anomaliiile de actualizare.

Forme normale:

1NF – domeniul atributelor sa cuprinda valori atomice; se interzic campurile compuse sau relatii in relatii;

2NF – 1NF; orice atribut neprim (care nu face parte din cheia primara) sa fie complet dependent functional de cheia primara a relatiei;

3NF – 2NF; nu exista nici un atribut neprim care sa fie dependent tranzitiv de cheia primara a relatiei;

In interiorul tabelor prezinta importanta campurile sau grupurile de campuri care identifica unic inregistrările sau care ajuta la legarea datelor din mai multe tabele. Cheia primara reprezinta un atribut sau un grup de attribute care identifica unic o inregistrare. Se numeste cheie externa un atribut sau un grup de attribute care constituie o cheie primara intr-o alta relatie.

Identificati in tabelele clienti si comenzi attributele care sunt chei primare, respectiv externe.

2. Definiti conceptul de cheie externa. Exemplificati. (3.2. - pag.16)

Comenzi

IdComanda	IdClient	Valoare	Data
1	2	134.67	2005-07-12
2	1	23.9	2006-01-20
3	3	150	2006-03-13
4	1	1234	2006-09-24

3.2. Proiectarea unei baze de date

Proiectarea unei baze de date este un proces foarte important care cuprinde urmatoorii pasi:

- analiza problemei;
- identificarea datelor;
- normalizarea.

Normalizarea unei baze de date consta in principal in descompunerea modelului bazei de date in mai multe relatii astfel incat sa se reduca la maxim redundanta datelor si implicit sa elimine anomaliiile de actualizare.

Forme normale:

1NF – domeniul atributelor sa cuprinda valori atomice; se interzic campurile compuse sau relatii in relatii;

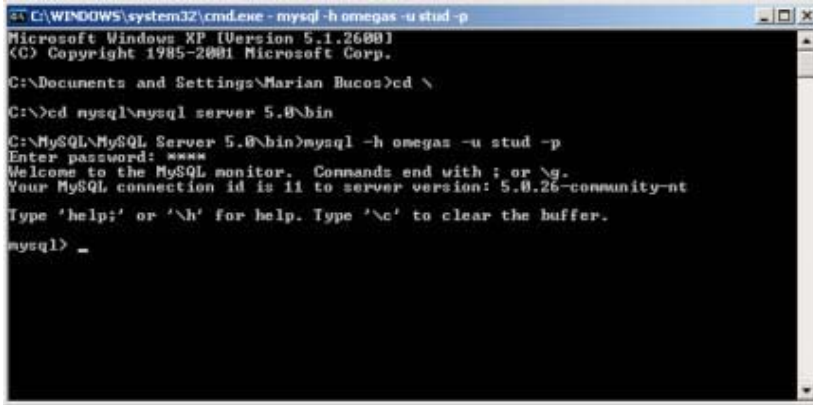
2NF – 1NF; orice atribut neprim (care nu face parte din cheia primara) sa fie complet dependent functional de cheia primara a relatiei;

3NF – 2NF; nu exista nici un atribut neprim care sa fie dependent tranzitiv de cheia primara a relatiei;

In interiorul tabelor prezinta importanta campurile sau grupurile de campuri care identifica unic inregistrările sau care ajuta la legarea datelor din mai multe tabele. Cheia primara reprezinta un atribut sau un grup de attribute care identifica unic o inregistrare. Se numeste cheie externa un atribut sau un grup de attribute care constituie o cheie primara intr-o alta relatie.

Identificati in tabelele clienti si comenzi attributele care sunt chei primare, respectiv externe.

3. Cum poate fi rulat un fisier de comenzi SQL din linia de comanda? (2.5. – pag.12)



2.4. Interogari de stare

Dupa conectare pot fi rulate interogari care furnizeaza date despre starea serverului: versiunea programului, data curenta de pe server.

! Verificati versiune serverului de baza de date omegas, si data la un moment dat. `mysql> select version(), current_date;`

Obs. Interogari se pot despartii pe mai multe randuri si se incheie cu ; . Pot contine orice combinatie de caractere mari/mici, limbajul de interogare nefiind case sensitive.

Pentru a obtine informatii despre server (bazele de date de pe server, tabelele dintr-o baza de date, coloanele dintr-un tabel al unei baze de date) se pot rula interogari show.

! Din clientul linie de comanda se vor verifica urmatoarele interogari show:

```
mysql> show databases;
```

```
mysql> show tables from mysql;
```

```
mysql> show columns from mysql.user;
```

```
mysql> show columns from user from mysql;
```

2.5. Rularea unui fisier extern de comenzi SQL

Clientul `mysql.exe` se poate utiliza si pentru a rula un fisier text de comenzi SQL. In acest caz poate fi utilizata comanda:

```
mysql -h nume_server -u nume_utilizator -p < nume_fisier.sql
```

Deasemenea, fisierele de comenzi SQL pot fi rulate din linia de comanda `mysql` utilizand sintaxa:

```
mysql> source nume_fisier.sql
```

shareware – software distribuit gratuit (sau pentru o taxa simbolica), pentru care se aplica anumite reguli;

multi-thread – descrie un program care este proiectat pentru a avea parti ale codului executate concurent;

API – Application Programming Interface

2.6. Sistemul de privilegii al MySQL

Funcția principală a sistemului de privilegii al MySQL este aceea de a autentifica și autoriza utilizatorii conectați la server.

Autorizarea se referă la permisiunea de a rula interogari precum SELECT, INSERT, UPDATE sau DELETE. O clasă aparte de privilegii se referă la drepturile de administrare și de interacțiune cu sistemul de operare.

La instalarea sistemului este creat implicit un utilizator numit root care are toate drepturile activate. Acest utilizator trebuie folosit, din motive de securitate, doar pentru administrare.

Pentru fiecare utilizator care va interacționa cu serverul trebuie creat un utilizator.

Privilegiu	Semnificatie
select	permite selectarea(vizualizarea) datelor
insert	permite adaugarea de noi inregistrari
update	permite modificarea datelor
delete	permite stergerea inregistrarilor
index	permite crearea/stergerea indecsilor
alter	permite redenumirea sau modificarea structurii tabeli

4. Precizati cinci tipuri de date puse la dispozitie de limbajul SQL (Structured Query Language). (3.5.1 – pag.17)

3.3. Crearea bazei de date in MySQL

Pentru a crea o baza de date se utilizeaza comanda:

```
create database [if not exists] nume_baza_date;
```

Clauza *if not exists* inhiba afisarea unui mesaj de eroare in cazul in care in sistem exista o alta baza de date cu acelasi nume.

Stergerea unei baze de date se realizeaza cu sintaxa:

```
drop database [if exists] nume_baza_date;
```

Daca sintaxa comenzii *drop* include clauza *if exists* nu sunt afisate mesajele de eroare care pot aparea daca se incearca stergerea unei baze de date care nu exista.

Comanda *use* stabileste baza de date pentru care se vor executa interogari ulterioare. La un moment dat poate fi activa o singura baza de date.

3.4. Tabele

Pentru a crea o tabela se va utiliza comanda:

```
create table [if not exists] nume_tabel [definitie]
```

, unde *definitie*:

```
nume_coloana tip [not null | null] [default valoare]
[auto_increment] [primary key]
```

Stergerea unei tabele se face folosind urmatoarea sintaxa:

```
drop table [if exists] nume_tabel;
```

3.5. Tipuri de date, operatori si functii

3.5.1. Tipuri de date

Tipuri de date: numerice, logice, data calendaristica, timp, siruri de caractere, date binare mari (blob).

Optiuni: unsigned, zerofill, binary.

1. Tipuri de date numerice: tinyint, smallint, mediumint,int, bigint, float, double, real, decimal.

2. Tipuri de date data calendaristica si timp: date, datetime, timestamp, time, year.

3. Tipuri de date sir de caractere: char, bit, bool, varchar.

4. Tipuri de date binare mari: tinyblob, tinytext, blob, text, mediumblob, mediumtext, longblob,mediumtext.

3.5.2. Constante, identificatori, comentarii

Constante tip sir de caractere: 'constanta sir', "alta constanta sir".

Constante intregi, reale: 1345, 543.36 .

Constanta NULL – nici o valoare; NULL<>0 ; NULL<>' ' .

Numele pentru baza de date, tabele, coloane trebuie sa indeplineasca conditiile normale pentru identificatori in limbaje de programare: sa fie o combinatie de litere, cifre si semne grafice care incep cu o litera.

Comentarii pe o singura linie: # comentariu 1

-- comentariu 2 --

Comentariu pe mai multe linii: /* comentariu 3.1

comentariu 3.2 */

3.5.3. Operatori

- grupare: (,)

- aritmetici: +, -, *, /

- logici: NOT !, OR ||, AND &&

- de comparare: =, <>, !=, <, >, <=, >=, IS NULL, expr BETWEEN min AND max, expr IN (value...), IF (expr1,expr2,expr3)

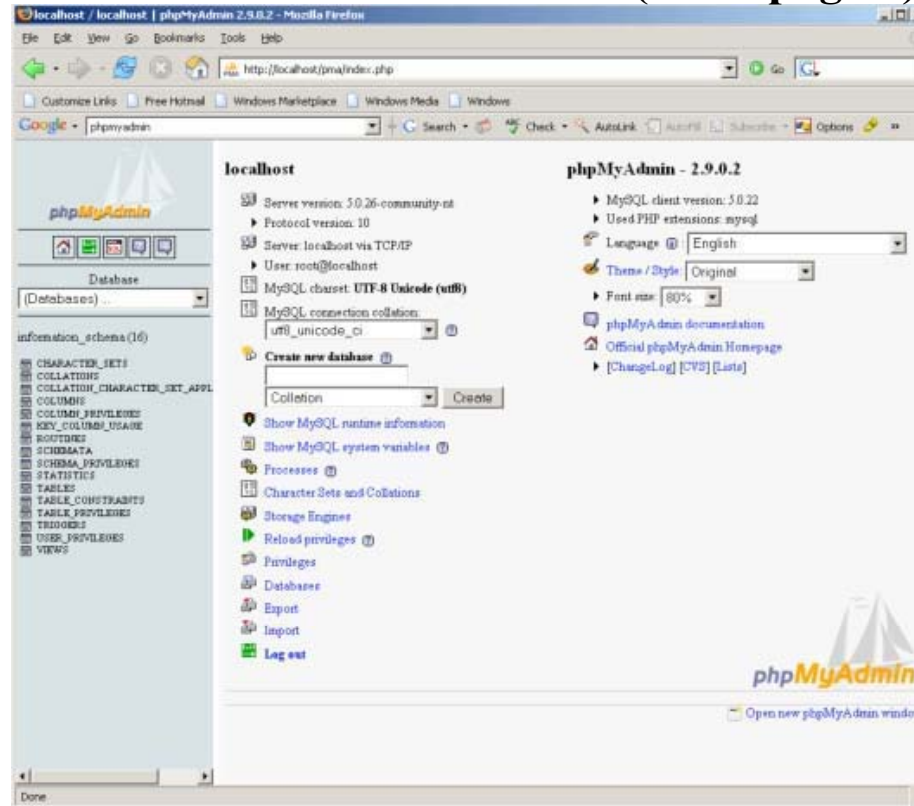
3.5.4. Functii – utilizare functii de biblioteca

1. Functii pe siruri de caractere

- ASCII (str) – intoarce codul ASCII al caracterului de pe pozitia 1 din sir;

- CONV (N, from_base, to_base) – converteste numarul N considerat in baza from_base in valoarea sa in baza to_base.

5. Care este sintaxa de conectare la serverul MySQL prin intermediul clientului linie de comanda? Precizati rolul parametrilor in comanda de conectare. (2.3. – pag.11)



2.3. Conectarea la serverul MySQL folosind clientul linie de comanda

Clientul tip linie de comanda va fi apelat din subdirectorul bin al directorului de instalare, folosind urmatoarea comanda:

```
mysql -h nume_server -u nume_utilizator -p
```

,unde:

- * h nume_server este numele masinii pe care se gaseste instalat serverul;
- * u nume_utilizator este numele utilizatorului care acceseaza serverul;
- * p activeaza citirea unei parole pentru contul de utilizator.

6. Precizati doua comenzi SQL utilizate in procesul de gestionare a datelor unui tabel. (4.1., 4.2, 4.3. – pag.23)

4. Gestionarea datelor folosind interogari SQL

4.1. Adaugarea inregistrarilor in tabele

Comanda INSERT permite inserarea (adaugarea) de noi inregistrari intr-o tabela.

Comanda INSERT din mysql are sintaxa de baza:

```
INSERT [LOW_PRIORITY| DELAYED] [IGNORE]
[INTO] nume_tabel [(nume_coloana, ...)]
VALUES (expresie, ...), (...), ...
```

LOW_PRIORITY – se foloseste pentru a intarzia scrierea efectiva a datelor in tabela pana cand alti utilizatori nu mai citesc date din tabela. Efectul este blocarea executiei pana cand se reuseste scrierea efectiva.

DELAYED – are actiune opusa parametrului precedent. Inregistrarea care trebuie adaugata este pusa intr-o coada de asteptare pe server si controlul revine la client, ca si cum scrierea ar fi fost facuta efectiv.

IGNORE – este util daca se insereaza mai multe inregistrari simultan. Prin folosirea parametrului IGNORE inregistrarile gresite sunt ignorate, dar restul sunt adaugate fara raportarea unei erori.

INTO – este optional, se foloseste pentru compatibilitatea cu alte sisteme SQL.

VALUES – specifica seturile de date (inregistrarile) care se vor adauga in tabela.

Tema 1

Adaugati o inregistrare in una din tabelele bazei de date BANCA. Vizualizati mai intai structura tabelor acestei baze de date.

Folositi instructiunea INSERT pentru a adauga simultan mai multe inregistrari in una din tabelele bazei de date BANCA.

4.2. Modificarea inregistrarilor din tabele

Comanda UPDATE permite actualizarea (modificarea) valorilor dintr-o tabela.

```
UPDATE [LOW_PRIORITY] [IGNORE] nume_tabel
SET nume_col1 = expresie1, nume_col2 = expresie2, ....
[WHERE conditie_actualizare]
```

WHERE conditie_actualizare – indica acele inregistrari care sunt actualizate.

Tema 2

Modificati valorile pentru doua coloane ale unei tabele din baza de date BANCA.

Modificarea se va realiza pentru o singura inregistrare folosind o conditie de actualizare.

4.3. Stergerea inregistrarilor din tabele

Pentru stergerea datelor din tabele se foloseste comanda DELETE.

Datele o data sterse nu mai pot fi recuperate.

Sintaxa comenzii DELETE este:

```
DELETE [LOW_PRIORITY] FROM nume_tabel
[WHERE conditie_stergere]
```

Tema 3

Stergeti o inregistrare din una din tabelele bazei de date BANCA folosind conditia de stergere.

4.4. Interogarea datelor

Selectia simpla

Comanda SELECT este cea mai utilizata comanda SQL. Ea permite atat regasirea si vizualizarea datelor din tabelele bazei de date cat si calcularea unor expresii care nu au legatura cu datele din tabele.

Sintaxa comenzii SELECT este:

7. Precizati variantele posibile in cazul operatiei de JOIN. (4.4. – pag.26)

Extragerea informatiei de sumarizare

Informatiile de sumarizare reprezinta informatii globale despre datele din tabele. Informatiile de sumarizare se extrag pe grupuri de inregistrari.

Gruparea se realizeaza folosind urmatorii parametrii in sintaxa comenzii SELECT:

GROUP BY {coloana | expresie} – specifica dupa ce valori se vaface gruparea. In majoritatea cazurilor expresia de grupare este reprezentata de o singura coloana. Toate inregistrarile cu aceeasi valoare pentru expresia de grupare vor fi considerate ca facand parte din acelasi grup.

Functii de sumarizare: count(),avg(), min(), max(), sum()

JOIN

Limbajul SQL foloseste comanda SELECT pentru implementarea operatorului JOIN. Acest operator permite colectarea datelor din tabele aflate in legaturi relationale. Folosirea comenzii SELECT pentru a face JOIN presupune specificarea in clauza FROM a tabelor de unde se preiau datele si, in lista de proiectie, a campurilor care vor face parte din rezultat.

SELECT lista_campuri FROM lista_tabele

WHERE conditii_join AND conditii_selectie ORDER BY ...

Exista 2 categorii de JOIN:

INNER JOIN – se includ in rezultat doar campurile care au corespondent la ambele capete ale relatiei

OUTER JOIN – va completa automat campurile care lipsesc din tabela corespondenta cu valoarea NULL (LEFT OUTER JOIN ; RIGHT OUTER JOIN)

Sintaxa comenzii SELECT se modifica in felul urmator:

SELECT lista_campuri

FROM tabela1 [INNER JOIN | LEFT OUTER JOIN | RIGHT OUTER JOIN] tabela2

ON conditie_join

WHERE conditie_selectie

8. Explicati modalitatile de introducere a unui script PHP intr-o pagina Web. (5.2. –pag.29)

5. Limbajul de scripting PHP

5.1. Notiuni introductive

Limbajul PHP este un limbaj de tip script creat special pentru Web. Este un limbaj interpretat de server (server-side scripting language). Permite generarea flexibila a unor pagini dinamice care ofera informatii in timp real.

Sa se afiseze prin intermediul unui script mesajul Hello World!.

```
<?php  
    echo 'Hello World!';  
?>
```

5.2. Inserarea codului PHP intr-o pagina HTML

Scripturile PHP sunt destinate generarii dinamice de continut in paginile Web. Un astfel de script trebuie inclus intr-o pagina HTML. Se pot utiliza urmatoarele tipuri de tag-uri HTML:

-stilul scurt (nu este valabil pentru php5):

```
<? echo "text generat dinamic"; ?>
```

-stilul XML:

```
<?php echo "text generat dinamic"; ?>
```

-stilul SCRIPT:

```
<SCRIPT LANGUAGE='php'> echo "text generat dinamic"; </SCRIPT>
```

-stilul ASP:

```
<% echo "text generat dinamic"; %>
```

5.3. Generarea dinamica a continutului

Codul PHP nu va fi vizibil in sursa trimisa de serverul Web catre navigatorul clientului. Acest text va fi interpretat de catre modulul PHP din server si doar textul generat de acest script va fi trimis mai departe de catre server.

La scrierea codului, instructiunile PHP din script vor fi terminate printr-un separator ; .

9. Care sunt tipurile de conexiuni catre un server MySQL ce pot fi stabilite dintr-o pagina PHP?

(6.1. – pag.38)

6. Accesarea serverului MySQL utilizand PHP

6.1. Conectarea la serverul de baza de date

Pentru a putea interoga o baza de date , o pagina PHP trebuie sa stabileasca in prealabil o conexiune cu serverul MySQL. Conexiunea poate fi privita ca un canal de comunicatie prin care programul transmite cereri SQL iar serverul returneaza raspunsurile corespunzatoare.

Funcția care realizeaza conectarea persistenta la serverul de baza de date este *mysql_pconnect()*.

In general trebuie sa introduci host-ul server-ului MySQL, numele cu care utilizatorul se conecteaza si parola folosita de acesta. Toate acestea sunt optionale, iar daca nu sunt specificate functia foloseste setarile default – localhost pentru host, numele utilizatorului cu care procesul PHP ruleaza si o parola vida.

Funcția returneaza o legatura catre baza de date sau eroare in caz de nereusita.

O alta functie care realizeaza conectarea – conexiune nepersistenta – la serverul de baza de date este *mysql_connect()*.

O conexiune va fi inchisa cand se incheie executia unui script – terminare pagina – sau cand se apeleaza functia *mysql_close()*. O conexiune persistenta ramane deschisa si dupa ce executia scriptului s-a terminat si nu poate fi inchisa folosind functia *mysql_close()*.

```
<?php
$link = mysql_connect("localhost","stud19","stud");
```

```
if ($link) echo 'conexiunea s-a realizat cu succes';
else echo 'nu se poate realiza o conexiune la serverul de baza de date';
?>
```

Cand se realizeaza conectarea la serverul de baza de date trebuie stabilita si baza de date cu care se va lucra. Acest lucru se face din PHP utilizand functia *mysql_select_db()*.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
        or die('nu se poate realiza o conexiune la serverul de baza de date');
```

```
$db = mysql_select_db("banca");
```

```
if ($db) echo 'baza de date a fost selectata';
else echo 'baza de date nu poate fi selectata';
?>
```

6.2. Rularea interogarilor

Pentru rularea unei interogari pe server se utilizeaza functia *mysql_query()*.

Funcția presupune existenta unei conexiuni deschise spre server, selectia prealabila a bazei de date si existenta unor drepturi suficiente pentru rularea interogarii. Funcția returneaza false daca interogarea nu a putut fi executata pe server.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
        or die('nu se poate realiza o conexiune la serverul de baza de date');
$db = mysql_select_db("banca") or die('baza de date nu poate fi selectata');
```

```
$result = mysql_query("select * from deponent");
```

```
if ($result) echo 'interogarea a fost rulata cu succes';
else echo 'interogarea nu poate fi rulata';
?>
```

Obs: ! die() – afiseaza un mesaj si termina executia scriptului;

6.3. Obținerea si afisarea datelor

Pentru a afla numarul inregistrarilor afectate de o interogare actiune (insert , delete , update) se poate folosi functia *mysql_affected_rows()*.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
        or die('nu se poate realiza o conexiune la serverul de baza de date');
$db = mysql_select_db("banca") or die('baza de date nu poate fi selectata');
```

```
$result = mysql_query("insert into deponent(nr_buletin,cnp,nume,prenume)
        values('342674','73463726','pop','sebastian)");
```

10. Explicati doua metode care permit interpretarea si prelucrarea rezultatului unei comenzi SELECT intr-un script PHP. (6.3. – pag.39)

```
if ($bd) echo 'baza de date a fost selectata';
else echo 'baza de date nu poate fi selectata';
?>
```

6.2. Rularea interogarilor

Pentru rularea unei interogari pe server se utilizeaza functia *mysql_query()*.
Functia presupune existenta unei conexiuni deschise spre server, selectia prealabila a bazei de date si existenta unor drepturi suficiente pentru rularea interogarii. Functia returneaza false daca interogarea nu a putut fi executata pe server.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$db = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent");

if ($result) echo 'interogarea a fost rulata cu succes';
else echo 'interogarea nu poate fi rulata';
?>
```

Obs: ! die() – afiseaza un mesaj si termina executia scriptului;

6.3. Obtinerea si afisarea datelor

Pentru a afla numarul inregistrarilor afectate de o interogare actiune (insert , delete , update) se poate folosi functia *mysql_affected_rows()*.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$db = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("insert into deponent(nr_buletin,cnp,nume,prenume)
    values('342674','73463726','pop','sebastian)");
```

```
or die('interogarea nu poate fi rulata');
```

```
echo 'inregistrari afectate: ' .mysql_affected_rows();
?>
```

In urma rularii unei interogari select se poate folosi functia *mysql_num_rows()* pentru a afla numarul inregistrarilor din rezultat.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$db = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');
```

```
echo 'inregistrari afectate: ' .mysql_num_rows($result);
?>
```

Exista trei metode diferite de a interpreta si prelucra rezultatul unei interogari select:

- a) rezultatul este preluat prin functia *mysql_fetch_array* si interpretat ca un tablou asociativ; acest tablou are ca si chei numele coloanelor selectate iar ca si valori, valorile corespondente din rezultat; fiecare apel succesiv al functiei va returna urmatoarea inregistrare din rezultat.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$db = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');
```

```
while($row = mysql_fetch_array($result)) {
    echo $row["nume"].' '.$row["prenume"].'<br>';
}
?>
```

- b) rezultatul este preluat prin functia *mysql_fetch_rows* si interpretat ca un tabel indexat numeric; acest tablou are ca si indexi numerele 0,1,2 etc. iar ca si valori, valorile

10. Explicati doua metode care permit interpretarea si prelucrarea rezultatului unei comenzi SELECT intr-un script PHP. (6.3. – pag.39)

2/2

corespondente coloanelor de pe pozitiile respective din rezultat; fiecare apel succesiv al functiei va returna urmatoarea inregistrare din rezultat.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');

while($row = mysql_fetch_row($result)) {
    echo $row[2]. ' ' . $row[3]. '<br>';
}
?>
```

c) rezultatul este preluat prin functia mysql_fetch_object si interpretat ca un obiect.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');

while($row = mysql_fetch_object($result)) {
    echo $row->nume.' ' . $row->prenume.' <br>';
}
?>
```

Fiecare apel succesiv al functiei mysql_fetch_object va returna urmatoarea interogare din rezultat.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');
```

```
for($i=0; $i<mysql_num_rows($result); $i++) {
    $row = mysql_fetch_object($result);
    echo $row->nume.' ' . $row->prenume.' <br>';
}
?>
```

6.4. Adaugarea datelor din pagina Web in baza de date

Pentru a adauga date dintr-o pagina web se realizeaza o interogare INSERT pe server utilizand functia mysql_query().

Aceasta functie presupune existenta unei conexiuni deschise spre server, selectia prealabila a bazei de date si existenta unor drepturi suficiente pentru rularea interogarii.

Urmatorul script implementeaza un mecanism de introducere (adaugare) de noi inregistrari in tabela deponent din baza de date banca.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

if ($_GET["action"] == 'insert' && $_POST["nr_buletin"] <> "")
    mysql_query("insert into deponent(nr_buletin,cnp,nume,prenume,oras,adresa)
        values('".$_POST["nr_buletin"]."',".$_POST["cnp"]."',".$_POST["nume"]."',
            '".$_POST["prenume"]."',".$_POST["oras"]."',".$_POST["adresa"]."')")
    or die('interogarea nu poate fi rulata');

$result = mysql_query("select * from deponent")
    or die('interogarea 2 nu poate fi rulata');
echo '[Nr. buletin] - [CNP] - [Nume] - [Prenume] - [Oras] <br />';
while($row = mysql_fetch_object($result)) {
    echo $row->nr_buletin.' ' . $row->cnp.' ' . $row->nume.' ' . $row->prenume.' ' . $row->oras.' <br />';
}
?>
```

```
<form action="<?php echo $PHP_SELF;?>?action=insert" method="post">
```

TEHNOLOGII MULTIMEDIA

ANUL 4, SEMESTRUL 7

1. Ce este o aplicație multimedia în viziunea modernă a lumii tehnologiei informațiilor?

Curs TMM - pag 47-48

În viziune modernă o aplicație multimedia conține: text, imagini statice, sunete, imagini video, animație, grafică strans legate între ele și care determină prin diferite metode, diferite abilitați de interactivitate cu utilizatorul.

O aplicație multimedia este completă atunci când conține o combinație a cel puțin 4 dintre elementele constitutive (cele 6 de mai sus) cu condiția ca ele să interacționeze între ele și să permită interacțiunea cu utilizatorul.

2. Ce înseamnă hypertext și HTML?

Curs TMM – pag 49

Din punct de vedere matematic, hypertextul se definește ca și mediu „n-dimensiuni”, adică ca “text / mediu cu n dimensiuni” sau un “mediu la dimensiune n”. Hypertext: textul are mai multe dimensiuni; bucăți din text pot fi introduse la momente diferite de timp, în zone diferite, în funcție de modul de navigare (dimensiunea după care se navighează) are altă structură. Hypertextul reprezintă o structură de text care permite saltul în interiorul aceluiași text pentru a se căuta o altă informație. Ex: în Internet: link-urile. HTML = HyperText Mark-Up Language

3. Designul informațional reprezintă...

Curs TMM – pag 76

Designul informațional

După ce informația a fost culeasă trebuie selectata astfel încât în dosarul aplicației să intre informația importantă și care are legătură cu subiectul. Există 3 nivele de eșalonarea a informației:

- principal - este cel care definește subiectul și pe care se va baza aplicația. Este prima informație perceptibilă de toți cei din publicul țintă, de aceea ea trebuie să fie clară și concisă.
- secundar – este informația care explică subiectul; informația secundară poate fi axată pe mai multe nivele, în funcție de designul informațional stabilit. Este informația specializată și care în funcție de nivelul de informație se va adresa unui număr mai restrâns din publicul țintă.
- senzorială – este informația care caracterizează subiectul și-l particularizează. Ea se va adresa senzorial utilizatorului, creându-i starea necesară înțelegerii aplicației (culoare, muzică).

4. Care sunt tipurile de link-uri (legaturi) pentru WWW? Enumerati si definiti (intr-o propozitie).

Curs TMM – pag 123-124

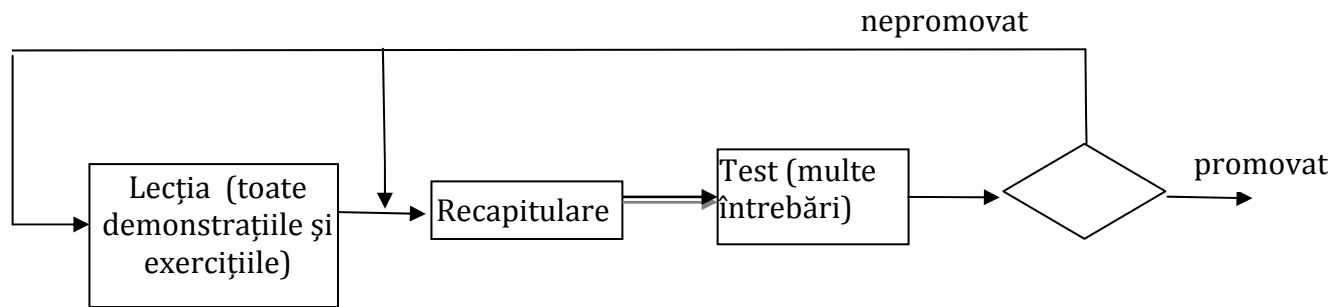
(se considera raspuns corect enumerarea corecta cu definitia din prima propozitie)

. Categorii de linkuri (legaturi)

- **Legaturi intrapagina:** cele care determina legaturi in pagina de web intre 2 pozitii diferite din pagina. Legaturile pastreaza aceeasi adresa de web si acelasi URL. Acest tip de legaturi se folosesc pentru a simplifica accesul la informatie mare sau din paginile web lungi. Pentru a putea realiza o legatura intrapagina infomatia trebuie impartita pe categorii dupa diversi indici si aceste categorii determina legatura in pagina. Este obligatoriu ca in pozitia inferioara a destinatiei legaturii linkului sa existe un alt link care sa ne readuca in pozitia initiala (top/ sus/ inapoi).
- **Legaturi intrasite-** legaturi intre 2 pagini diferite din acelasi site web, in acest scop nu se modifica domeniul ci doar extensia adresei. Sunt utilizate in trei scopuri:1)pt a compune o imagine informationala completa intre diferite obiecte informationale; 2)pt a putea crea o alternativa la legatura intrapagina atunci cand informatia este prea mare, prea lunga, prea voluminoasa; 3) pt a oferi legaturi in scop informational si de a crea relatii intre informatii diferite.
 - **Legaturi intersite** – sunt legaturi care creeaza salturi intre site-uri web diferite prin schimbarea adresei (URL-ului). Motivatia este de a oferi un mesaj informational cat mai complet utilizat, de a crea posibilitatea de a gasi informatie suplimentara, cat si de a mari traficul pe site. Daca aceste linkuri intersite sunt cu informatie valoroasa, interesanta, relevanta utilizatorul va aprecia faptul ca a fost oferita si va mai reveni pe site. Dar intodeauna aceste legaturi trebuie sa fie functionale (revizuirea continua a paginii de linkuri).

5. Care este organigrama de bază pentru aplicații multimedia de tip instruire interactivă (e-learning)?

Curs TMM – pag 82-83



6. Ce reprezinta World Wide Web, cine si unde l-a inventat ?

Curs TMM – pag 24-26, 41-46

World Wide Web este o structura globala informationala de tip hypermedia, bazata pe rețeaua Internet (similara cu notiunea de software).

WWW se bazeaza pe protocoalele de tip URL, HTTP, HTML.

World Wide Web a fost inventat de cercetătorul englez Tim Berners-Lee în 1989 în încercarea de a stoca eficient date provenite din cercetări la CERN, Elvetia. Berners-Lee, consultant specializat în programe de prelucrarea textului, și-a dorit un sistem care să pună la dispoziția cercetătorilor o modalitate mai ușoară de a realiza secțiuni separate de informații legându-le apoi electronic între ele. Sistemul său s-a bazat pe conceptul hypertext-ului, sau altfel spus pe textul cu legături care pot conduce la alte documente, fișiere, sunete, imagini sau chiar programe. Sistemul WWW permite hypertext-ului să facă legături cu fișiere de pe diferite platforme. El a scris primul protocol client si server web in 1990 si a definit notiunile de URI, HTTP si HTML.

7. Ce reprezinta Internetul si cum a fost format?

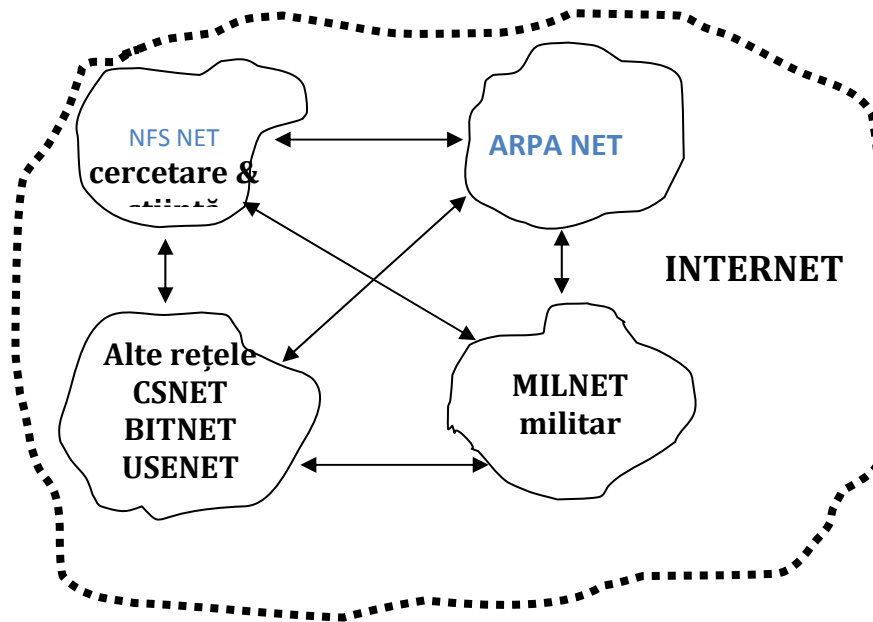
Curs TMM - pag 27 - 40

Dezvoltarea Internetului se leagă de înființarea în 1958 a unei agenții pentru proiecte de cercetare a tehnicii avansate, numită ARPA (Advanced Research Project Agency), aflată sub conducerea Pentagonului, cu scopul de a promova implementarea celor mai avansate tehnologii în toate domeniile.

În **1969** se lansează oficial prima rețea numită ARPANET. În 1979 ARPANET (numită acum DARPA) decide să se separe în două rețele: o rețea cu caracter educațional și comercial cu aceeași denumire de ARPANET și o rețea cu caracter militar, MILNET. Cele două rețele păstrează puncte comune și posibilități de schimb de informații, fapt care duce la perfecționarea sistemului de securitate a lor.

În paralel, în toți acești ani se dezvoltă și alte rețele cu caracter național în SUA în lumea universitară: CSNET (Computer and Science Network) și BITNET (Because it's time Network). În 1983 se realizează tranziția de la protocolul NCP la cel TCP/IP. În **1985** National Sciences Foundations (NSF) înțelege importanța unor rețele globale și uneste rețelele în rețeaua NSF NET destinată oamenilor de știință și cercetătorilor, rețea care lega cinci supercalculatoare. În 1984 ia ființă și rețeaua JANET localizată în Marea Britanie.

Unirea tuturor acestor rețele (între anii 1980 – 1990) a condus la realizarea INTERNET-ului, nume care provine din prescurtarea: **Inter Network Sistem** (Sistem de interconectare a rețelelor).



Conform definiției FNC (Federal Networking Council) din 1995 **INTERNET** se refera la un sistem integrat de resurse informationale globale care sunt (1) legate / inter-conectate logic printr-o adresa unica globala bazata pe IP Internet Protocol sau derivatii sai, (2) suporta comunicatii care folosesc suita de protocoale TCP/IP, (3) funizeaza, utilizeaza sau creaza accesul, public sau privat, la servicii de nivel inalt bazate pe tehnologii de comunicare. Mai simplu, Internetul este o retea de retele, reprezinta o retea globala de sisteme de calcul inter-conectate informational si comunicational pe baza protocoalelor de tip TCP/IP.

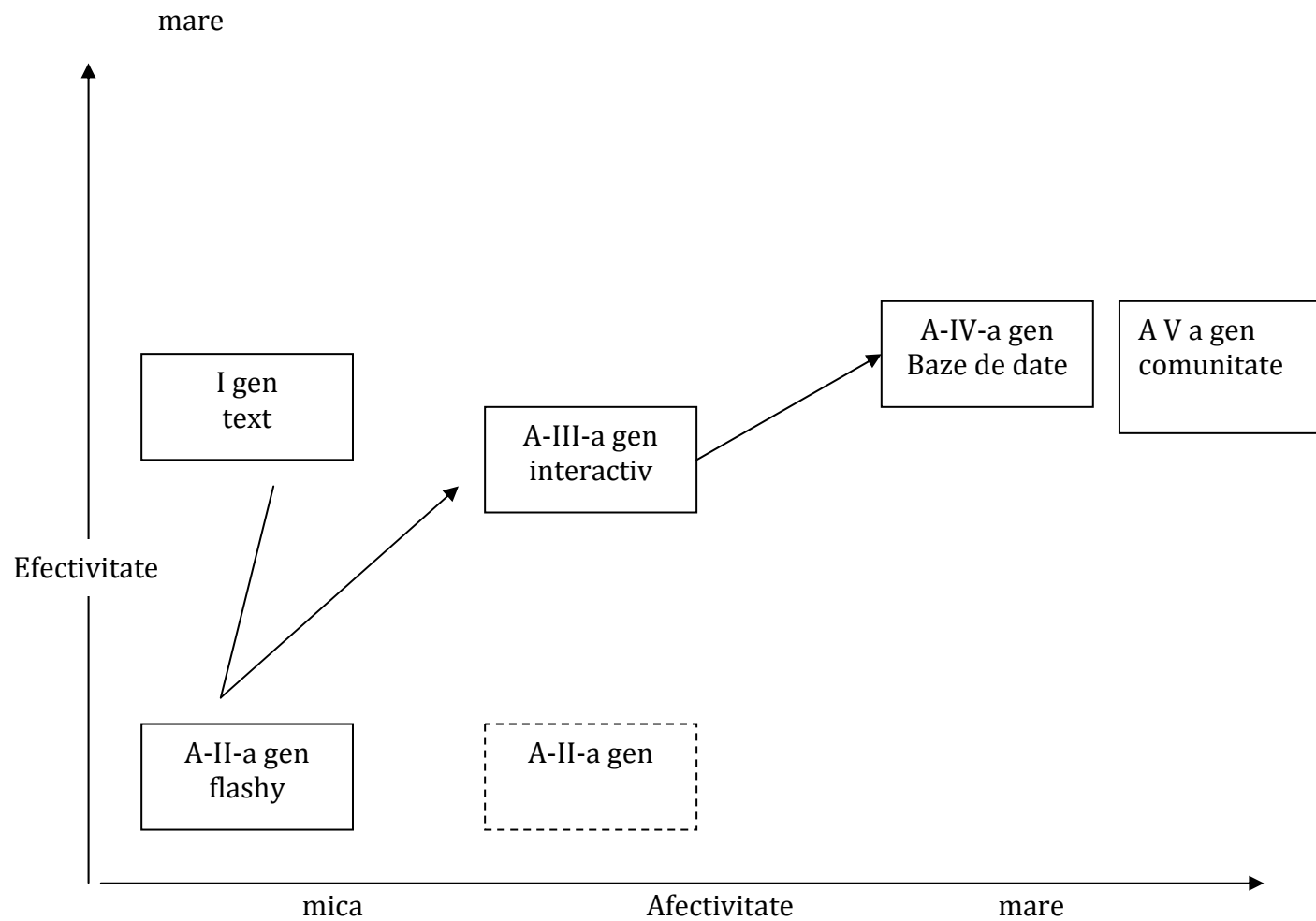
8. Care sunt clasificările cunoscute de site web și exemplificați grafic aceste categorii?

Curs TMM pag 120 - 122

(se considera raspuns corect enumerarea corecta cu realizarea graficului, chiar fara explicatii)

Clasificările de site web se bazează pe Aceste categorii se definesc prin metodele utilizate, dar și ca dezvoltare istorică.

- Prima generație de site-uri web este caracterizată printr-o apariție în stilul paginilor tiparite (de marketing sau de informare) prin conținut și prin efectivitate. În această generație link-urile nu sunt gândite printr-un sistem de navigare care să te ajute să găsești ușor informații, ci ele sunt doar accidentale în urma unei coincidențe multiple de text. Sunt caracterizate printr-un număr redus de vizitatori și un număr crescut de utilizatori.
- A-II-a generație (mijlocul anilor '90) este caracterizată prin exces grafic, elemente decorative și mai puțin text. Există multe link-uri în pagină, dar lipsește interactivitatea cu utilizatorul. Ca și dezavantaj: prin creșterea atractivității, a scăzut eficiența site-ului deoarece textul este foarte puțin.
- A-III-a generație reunește toate cele trei criterii de design ale site-ului web prin îmbinarea informației despre: audiență, scopul site-ului, criteriile design-ului utilizarea altor medii (audio, video, bază de date) și o interactivitate ridicată cu utilizatorul, cât și existența unor link-uri inter-site și extra-site. Se caracterizează prin mărirea numărului de utilizatori și începutul, istoric, în perioada de 'dot.com' de la începutul anilor 2000.
- A-IV-a generație de site-uri se caracterizează prin introducerea unui design informațional mai interactiv, reactualizarea zilnică a informației, cât și prin introducerea altor elemente media (elemente audio și video) de mare întindere.
- A-V-a generație de site-uri se caracterizează prin introducerea tehnologiilor web 2.0, a elementelor de semantic web și prin apariția social media. Tehnologiile au interactivitate mare și pun accentul pe utilizator, ele devenind creatorul de informație de 'siet web'.



9. Care sunt tipurile de motoare de cautare web? Dati exemple.

Curs TMM pag 139 – 143 , documentul Motoare de Cautare - links si Tehnologii Semantic Web.

Motoarele de căutare web se împart în trei categorii mari: cele care indexează informația automat, cele care se bazează pe informațiile introduse manual si cele mixte. La acestea se adauga motoarele de cautare cu indexare semantica. Motoarele de cautare web moderne ale acestui an inglobeaza toate elementele fiecărei dintre categorii: sunt si tematice, cautarea se bazeaza si pe roboti dar si in mod semantic (Google, Bing).

Tipurile de motoare de cautare web sunt (cu cateva exemple):

- Directoare tematice: ALIWEB, AltaVista, Yahoo, Lycos, MSN
- Motoare de căutare bazate pe roboți : AllTheWeb, HotBot, Google, Bing
- Motoare de metacăutare : Dogpile, Mamma, Kartoo, Metacrawler
- Motoare de cautare semantic: Wolfram Alpha, Hakia, SenseBot, DeepDyve, Cognition

10. Ce reprezinta tehnologiile web 2.0?

Documentul Tehnologii web 2.0 definitii, slide 9-17

La 30 septembrie 2005, Tim O'Reilly a scris un articol ce cuprindea viziunea lui despre Web 2.0. "What is Web 2.0", in jurul principiului participarii – daca Web 1.0 a fost Comerț, atunci Web 2.0 sunt Oamenii.

Web 2.0 este un concept care reuneste un set de tehnologii si servicii create in jurul ideii ca accesul si utilizarea Web sa nu mai fie pasiva ci utilizatorii sa devina contribuitori activi de continut, medii de comunicare si tehnologii. Web 2.0 are intelesuri diferite din punctul de vedere al tehnologiei, comunicarii, stiintelor sociale sau economice.

Tehnologiile Web 2.0 reprezinta un set de principii și practici care unesc un adevărat sistem solar de site-uri ce demonstrează unele dintre sau toate acele principii, la o distanță variabilă de principiile de baza:

- Web ca Platformă, transforma Web intr-un server de aplicații gigant
- Modele de programare usoare (lightweight), Thin Client Computing
- Distributia informatiei
- Inter-operabilitate, pentru mai multe tipuri de echipamente
- design centrat pe utilizator, generația viitoare de Software, conținut generat de utilizator
- Sfârșitul ciclului de lansare de software
- Colaborare
- Utilizatorii trebuie să fie tratați ca și co-dezvoltatori
- Susținerea inteligenței colective
- bogată experiență a utilizatorilor - 'Utilizatorul conduce!'
- Serviciul se îmbunătățește automat odată cu înmulțirea utilizatorilor
- Floksonomy (sistem de clasificare bazat pe cuvinte cheie si pe grupuri)

PRODUCȚIE AUDIO-VIDEO

ANUL 4, SEMESTRUL 8

1. Cum variază câmpul de focalizare în funcție de distanța focală a obiectivului?

- cu cât distanța focală este mai mare cu atât zona de profunzime este mai mică
- cu cât distanța focală este mai mică cu atât zona de profunzime este mai mare

2. Care sunt efectele modificării diafragmei asupra imaginii?

- deschiderea irisului mărește luminozitatea imaginii
- închiderea irisului mărește profunzimea focalizării

3. Cum este distanța focală a unui obiectiv cu sistem de transfocare? Cum este unghiul de deschidere pentru un obiectiv cu distanța focală mică?

- cu transfocator este variabilă
- cu unghi de deschidere mare este mică

4. Ce se folosește pentru eliminarea distorsiunilor de perspectivă ale obiectelor din imagine?

- alegerea unui unghi potrivit de filmare
- alegerea distanței focale adecvate

5. Cum se poate modifica compoziția în plan?

- modificarea poziției obiectelor în cadru
- prin plasarea adecvată a camerei în raport cu subiectul filmat
- modificarea poziției luminilor pentru a obține efecte cu ajutorul umbrelor

6. Compoziția în adâncime a unui plan se poate modifica în mai multe moduri, dați câteva exemple.

- reglarea focalizării pe obiecte aranjate pe axa optică a camerei
- iluminarea distinctă a elementelor de decor

7. Cum se realizează mișcările de cameră și care este scopul lor?

- să înceapă cu un cadru static urmat de mișcarea aparatului
- să se termine cu un cadru static
- să pornească de pe un centru de interes și să se oprească pe unul de importanță mai mare.

8. Cum se poate modifica atitudinea telespectatorului față de subiectul filmat?

- modul de alegere al unghiului de filmare
- modificarea modului de iluminare al subiectului
- organizarea unei compoziții de obiecte care să ghideze privirea spre subiectul respectiv

9. Care sunt funcțiile creatoare ale montajului?

- modificarea timpului prin dilatări sau comprimări
- obținerea de noi entități, ființe umane sau obiecte
- obținerea de noi spații

10. Enumerați câteva dintre regulile ce trebuie respectate pentru realizarea racordului de trecere între imagini .

- respectarea continuității de mișcare
- evitarea diferențelor de lumină între imagini succesive