

Proiect 1 – Proiectarea filtrelor FIR prin Metoda Ferestrei

Prelucrarea semnalelor

Pirvan Teodora-Maria (ng = 5; ns = 12)

Grupa: 333AC

Universitatea Politehnica din Bucuresti



Obiective

Înțelegerea Metodei ferestrei pentru proiectarea filtrelor FIR. Rezolvarea unor probleme de proiectare de tip răspuns cu toleranțe fixate pe baza Metodei ferestrei. Studiul caracteristicilor în frecvență ale mai multor tipuri de ferestre.

Faza 1 - Răspunsurile la impuls și în frecvență ale ferestrelor uzuale

Pentru realizarea cu succes al punctului initial au fost trasate raspunsurile la impuls ale celor noua ferestre, acest lucru fiind realizat cu ajutorul comenzilor prezentate in powerpoint-ul proiectului, din pagina 14 si al functiei „stem”. A fost folosita functia $[M,r,beta,L,alfa] = PS_PRJ_1_Faza_1(ng,ns)$ pentru aflarea parametrilor de lucru.

```
[M,r,beta,L,alfa] = PS_PRJ_1_Faza_1(ng,ns) ;
```

```
% M = 33 (întreg între 16 si 45)  
% r = 91.6216 (între 80 dB si 100 dB)  
% beta = 6.6486 (între 2 dB si 10 dB)  
% L = 5 (între 2 si 7)  
% alfa = 54.8649 (între 20% si 80%)
```

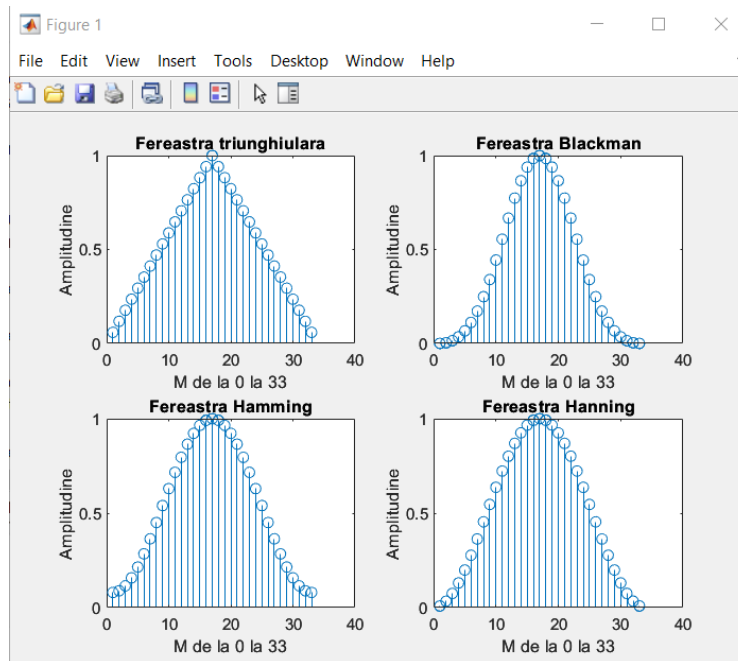


Figura 1 - cele 4 ferestre fara parametri, exceptand fereastră dreptunghiulară

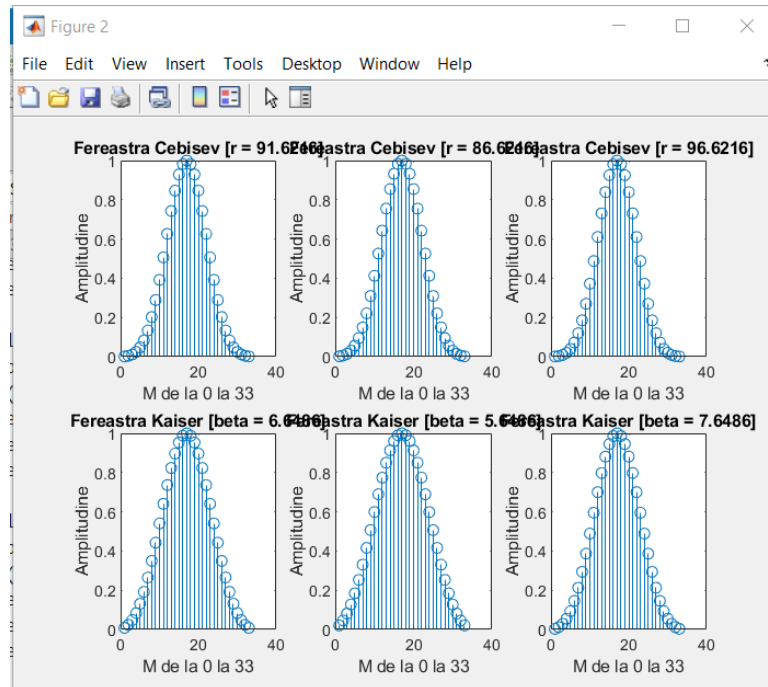


Figura 2 - cele 6 versiuni ale ferestrelor Cebisev (3 pe prima linie) si Kaiser (3 pe linia a doua)

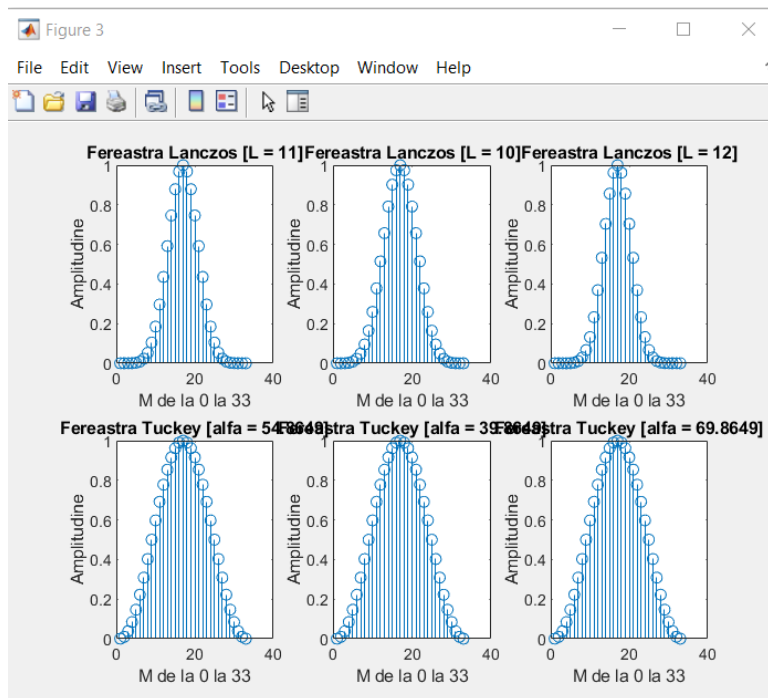


Figura 3 - 6 versiuni ale ferestrelor Lanczos (3 pe prima linie) si Tuckey (3 pe linia a doua)

Pentru obtinerea spectrelor a fost utilizata comanda „freqz”, rezolutia de reprezentare in frecventa fiind de 5000 de puncte. Pentru raspunsul la impuls a fost folosita procedura de normare.

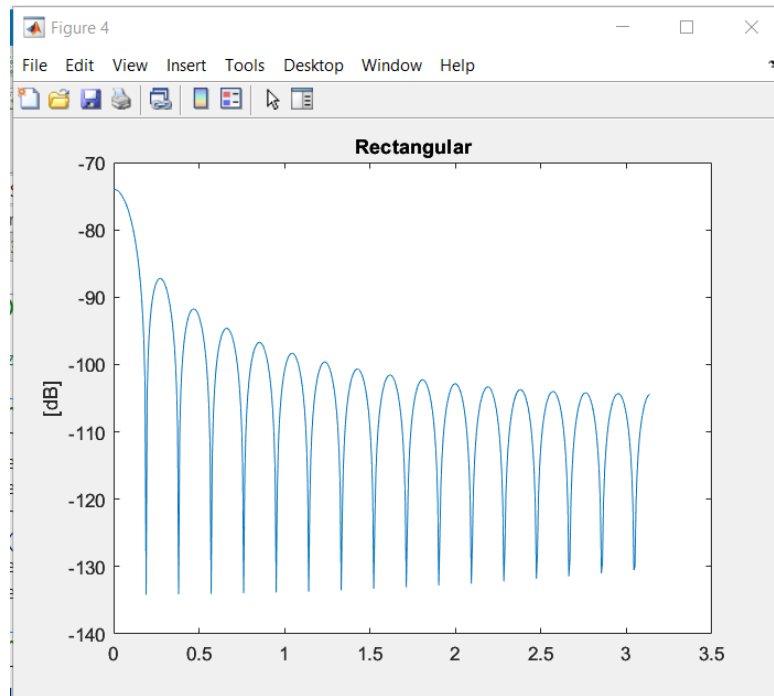


Figura 4 – spectrul ferestrei dreptunghiulare

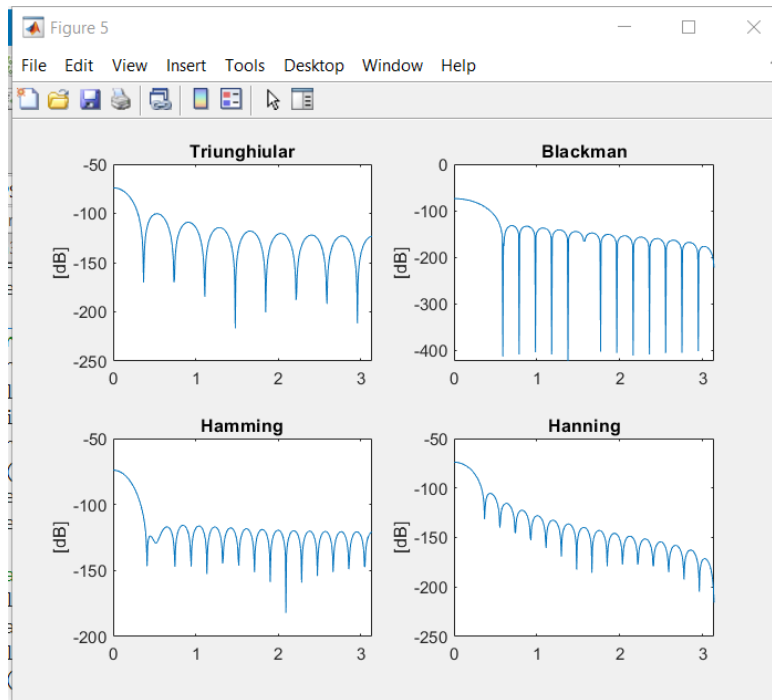


Figura 5 - spectrele celor 4 ferestre fara parametri

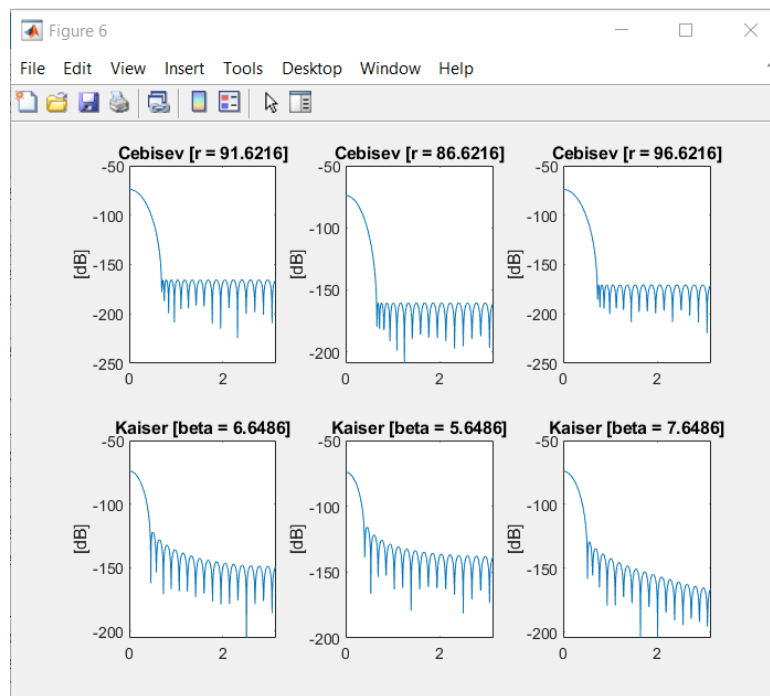


Figura 6 – spectrele celor 6 versiuni ale ferestrelor Cebisev si Kaiser

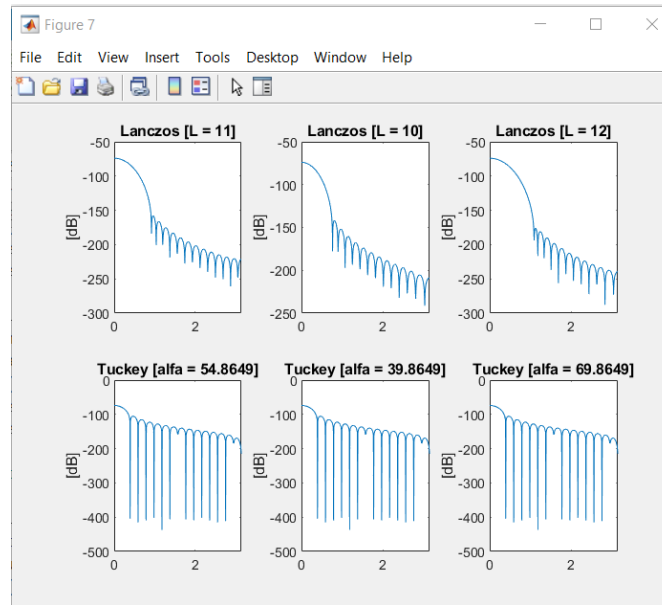


Figura 7 - spectrele celor 6 versiuni ale ferestrelor Lanczos si Tuckey

Faza 2 – Filtre proiectate cu diverse ferestre

Utilizand Metoda ferestrei, au fost proiectate FTJ de tip FIR de ordin $M = 33$, pulsatie de taiere fiind returnata de o functie proprie: $\omega_c = \text{PS_PRJ_1_Faza_2a}(\omega, \omega_s)$. Frecventa normalizata de taiere a fost aflata prin impartirea pulsatiei de taiere de 1.7236 (intre 0 si π) la π . Se observa, intr-adevar, faptul ca raspunsurile cu atenuare mare in banda de trecere au benzi de tranzitie largi, in timp ce cele cu benzi de tranzitie inguste au atenuari mici.

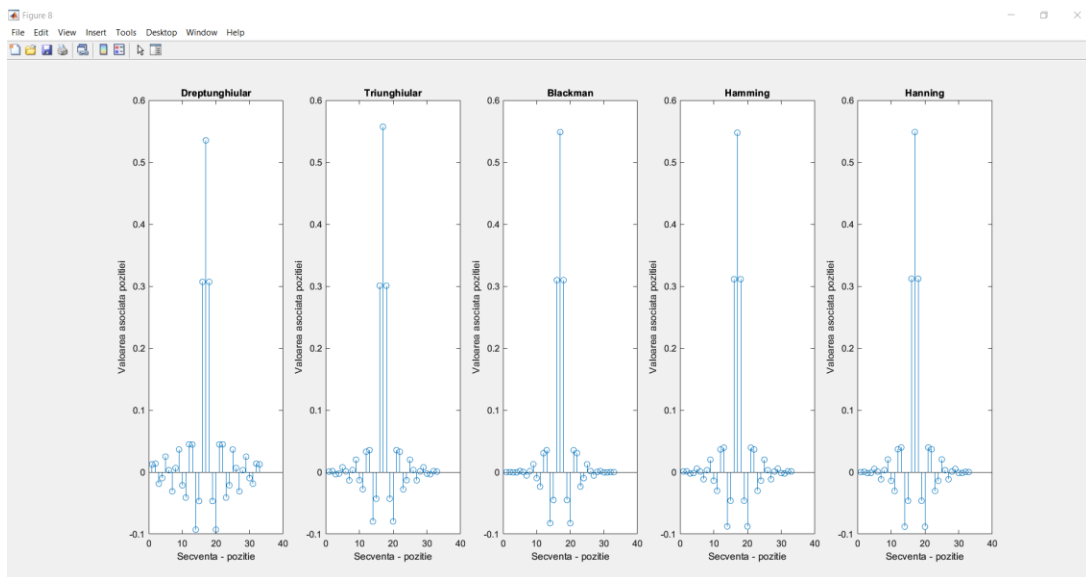


Figura 8 - secventele pondere ale filtrelor obtinute pentru fiecare fereastră neparametrică

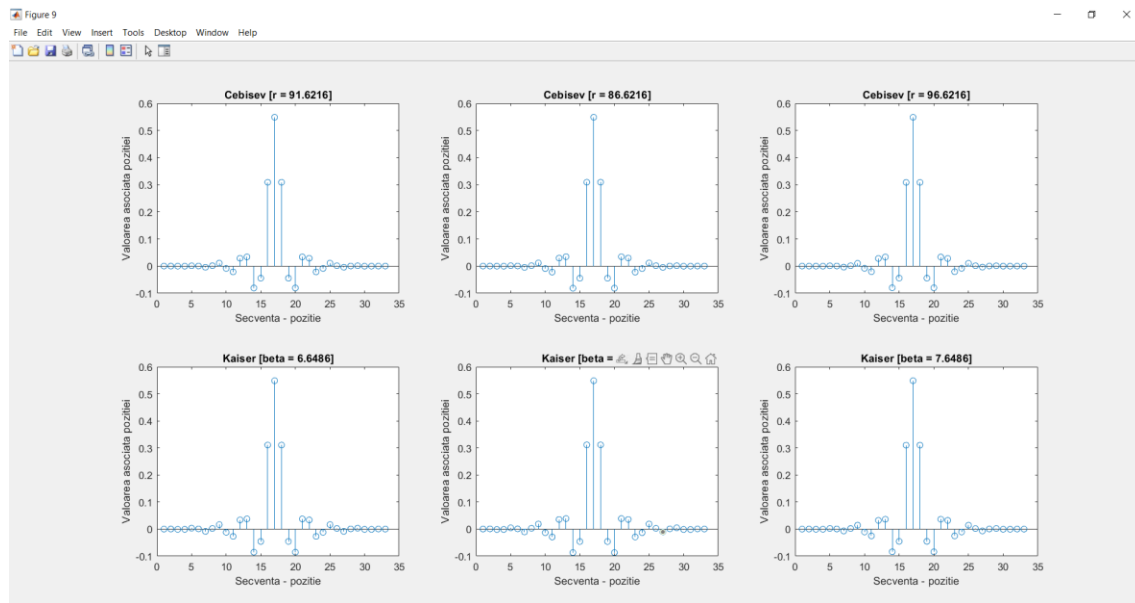


Figura 9 - secventele pondere ale filtrelor obtinute pentru ferestrele Cebisev (pe prima linie) si Kaiser (pe a doua linie)

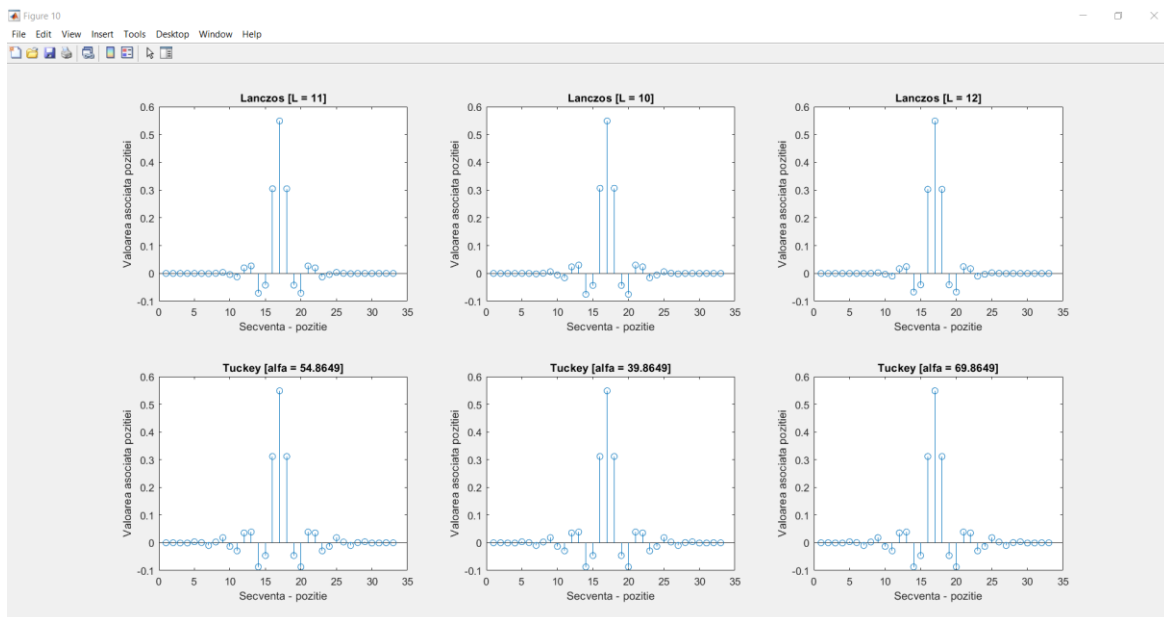


Figura 10 - secventele pondere ale filtrelor obtinute pentru ferestrele Lanczos (pe prima linie) si Tuckey (pe a doua linie)

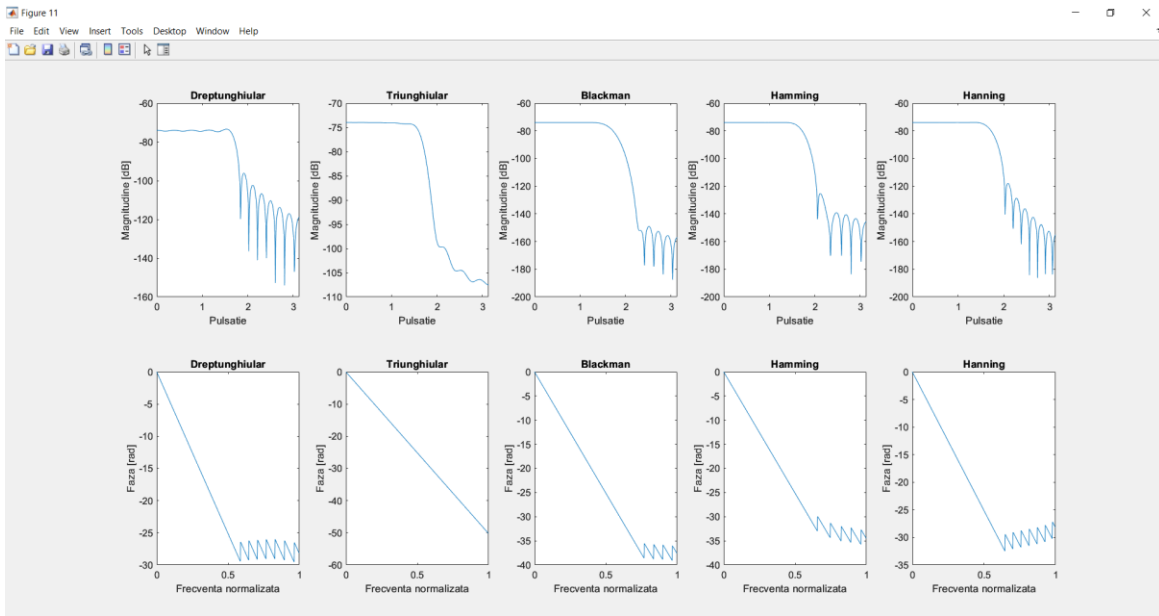


Figura 11 - spectrele filtrelor obtinute pentru fiecare fereastră ne-parametrică (inclusiv cea dreptunghiulară) pe prima linie și fazele aferente pe linia a doua

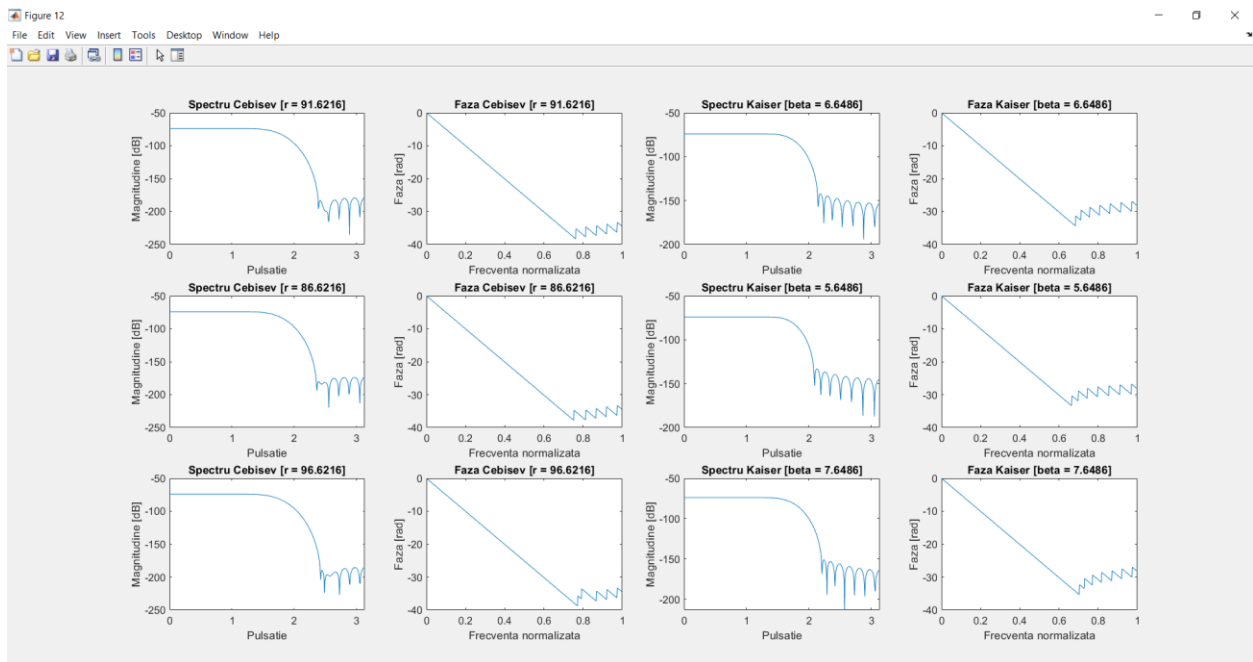


Figura 12 - spectrele filtrelor obtinute pentru ferestrele Cebisev și Kaiser pe coloanele 1, 3 (respectiv), cu fazele aferente pe coloanele 2, 4.

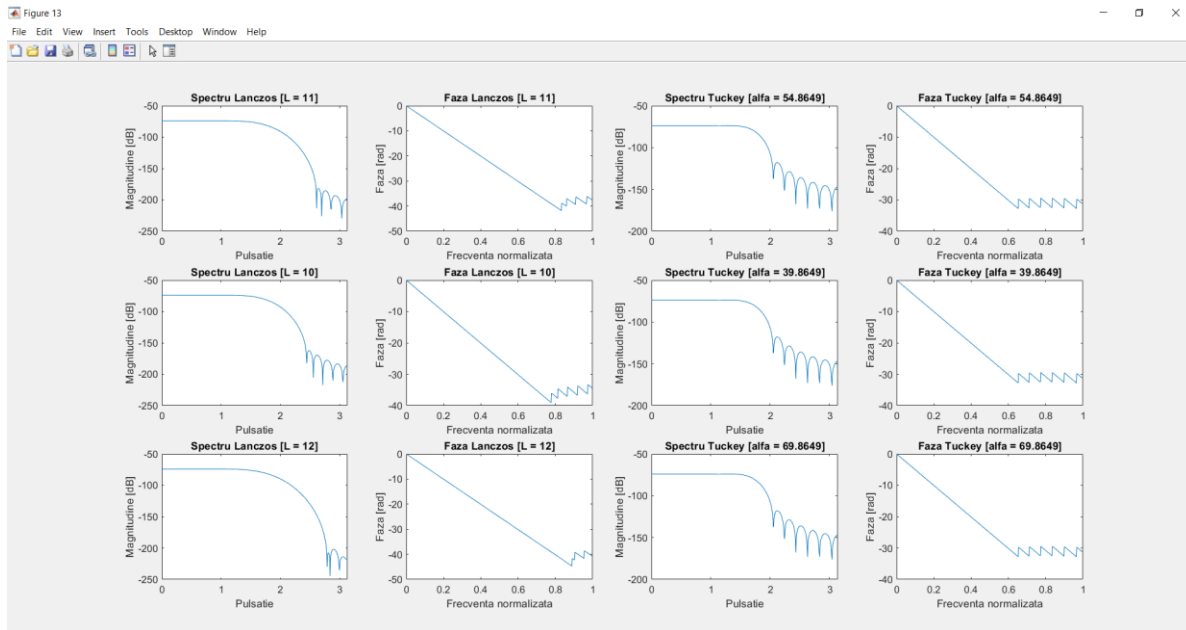


Figura 13 - spectrele filtrelor obtinute pentru ferestrele Lanczos

si Tuckey pe coloanele 1, 3 (respectiv), cu fazele aferente pe coloanele 2,4.

Clasificarea subiectiva a filtrelor

Pentru a se putea realiza o clasificare precisa a ferestrelor, este necesara ficarea unor caracteristici ce trebuiesc luate in calcul. Astfel, cu ajutorul powerpoint-ului pus la dispozitie (Pagina 12), principalele proprietati in studiul acestei clasificari au fost urmatoarele:

1. O deschidere cât mai mică a lobului principal din spectru (lobul trebuie sa fie cat mai îngust)
2. O atenuare accentuată a lobilor paraziți

Ulterior analizarii spectrelor ferestrelor, a fost realizat clasamentul celor 17 ferestre:

#	Fereastra
1.	Kaiser [beta = 5.6486]
2.	Kaiser [beta = 6.6486]
3.	Kaiser [beta = 6.6486]
4.	Cebisev [r = 86.6216]
5.	Cebisev [r = 91.6216]
6.	Cebisev [r = 96.6216]
7.	Lanczos [L = 10]
8.	Lanczos [L = 11]
9.	Lanczos [L = 12]

10.	Hanning
11.	Tuckey [alfa = 39.8649]
12.	Tuckey [alfa = 54.8649]
13.	Tuckey [alfa = 69.8649]
14.	Blackman
15.	Hamming
16.	Rectangular
17.	Triunghiular

Din clasament se observa ca primele trei pozitii sunt ocupate de ferestrele Kaiser, care par a fi cele mai eficiente, indeplinind cele doua proprietati cu succes (lobul principal este foarte ingust, iar atenuarea lobilor paraziti este foarte vizibila). In ceea ce priveste urmatoarele pozitii ocupate, Cebisev se dovedeste a fi destul de apropiat de Kaiser, marea diferenta fiind totusi observata asupra latimii lobului principal. Pe urmatoarele locuri din clasament se plaseaza fereastra Lanczos, datorita atenuarii accentuate a lobilor, nefiind atat de favorabila deschiderea lobului principal (este destul de mare fata de ferestrele precedente). Urmeaza Hanning, unde se observa o vizibila atenuare a lobilor, insa nu la fel de puternic precum la Lanczos, de asemenea evidentiindu-se si latimea lobului principal, care este destul de mica. Fereastra Tuckey are o deschidere scurta a lobului principal, insa este evidentiata cea de-a doua proprietate. De asemenea, se observa ca cele trei ferestre Tuckey sunt destul de asemanatoare! Blackman se aseamana destul de mult cu Tuckey, fapt pentru care este plasat imediat sub el. Toate cele trei ferestre ramase reprezinta variante foarte slabe calitativ, fiecare avand dejavantaje masive fata de celelalte ferestre prezentate, fapt pentru care sunt plasate pe ultimele locuri in urmatoarea ordine: Hamming, Rectangular, Triunghiular.

Clasamentul celor 9 ferestre:

#	Fereastra
1.	Kaiser [beta = 5.6486]
2.	Cebisev [r = 86.6216]
3.	Lanczos [L = 12]
4.	Hanning
5.	Tuckey [alfa = 39.8649]
6.	Tuckey [alfa = 54.8649]
7.	Hamming
8.	Rectangular
9.	Triunghiular

Pentru rezolvarea si demonstrarea punctului b. al Fazei 2 au fost alese ferestrele din fruntea clasamentului (Kaiser [beta = 5.6486]), respectiv cea plasata pe cea de-a noua pozitie (Lanczos [L = 12]). Ulterior realizarii modificarilor necesare asupra acestora, se vor compara cel mai bun filtru al ferestrei #9 cu cel mai slab filtru al ferestrei #1.

Cel mai slab filtru al ferestrei #1: Kaiser [$\beta = 5.6486$], $M = 33$ – Latimea lobului principal foarte mare, de asemenea atenuarea prea putin vizibila a lobilor paraziti

Cel mai bun filtru al ferestrei #9: Lanczos [$L = 12$], $M = 33$ – Latimea lobului principal este cea mai mica in comparatie cu celelalte doua spectre de acelasi tip, insa cu ult mai mare decat cel mai slab filtru al ferestrei #1. In ceea ce priveste atenuarea lobilor paraziti, se poate spune ca este destul de vizibila.

Faza 3 - Utilizarea Metodei ferestrei pentru rezolvarea PPFTI

Punctul a. propune realizarea unei functii MATLAB care primește ca argumente răspunsul la impuls al unui filtru FIR și frecvențele ω_p , ω_s , care mai apoi va întoarce abaterile maxime realizate Δ_{pr} și Δ_{sr} în benzile de trecere, respectiv stopare.

```
function [Delta_pr,Delta_sr] = faza3_puncta(h,omega_p,omega_s)

banda_trecere = 0:omega_p/1000:omega_p; % realizare banda trecere [0,omega_p];
banda_stopare = omega_s: omega_s/1000:pi; % realizare banda stopare
H1 = freqz(h,1,banda_trecere);
H2 = freqz(h,1,banda_stopare);
Delta_pr = max(abs(1-abs(H1))); % maximul distantei dintre |H|_p si 1 (pentru a determina Delta_pr)
Delta_sr = max(abs(H2)); % maximul lui |H|_s (pentru a determina Delta_sr).

end
```

Pentru demonstrarea punctului b. se vor proiecta 9 filtre de tip FIR, cate unul pentru fiecare combinatie (M, ω_c), unde ordinul M variaza in multimea $\{M \text{ (de referinta), } M+[M/2], 2M\}$, iar pulsatia de taiere, ω_c , se calculeaza ca $(1-w)*\omega_p + w*\omega_s$, unde w ia valori in multimea $\{0.25, 0.5, 0.75\}$. De asemenea, pentru fiecare filtru proiectat, se vor calcula tolerantele realizate, Δ_{pr} si Δ_{sr} , cu ajutorul rutinei de la punctul precedent. Ulterior, se calculeaza raspunsurile in frecventa ale celor 9 filtre, cu ajutorul functiei `FREQZ`, la rezolutia $K=5000$ si se afiseaza caracteristicile de frecventa ale filtrelor in ordinea descrescatoare a performantei.

A fost realizata special un program care sa proiecteze un singur filtru, dandu-se M , ω_c și tipul ferestrei. Se va lucra cu fereastra #1 din clasificarea anterioara (cea de 17) – Kaiser.

Parametri obtinuti ulterior rularii apelului `[omega_p,omega_s,Delta_p] = PS_PRJ_1_Faza_3b(ng,ns):`

```
%[omega_p,omega_s,Delta_p] = PS_PRJ_1_Faza_3b(ng,ns) ;
```

```
omega_p = 1.7236; % asemanatoare cu omega_c
omega_s = 2.1829;
Delta_p = 5.4865;
Delta_s = Delta_p;
beta = 7.6486;
```

```
%% Secvente pondere - ordinea descrescatoare a performantei
```

```
% Performanta = M cat mai mic si suma tolerantelor cat mai mica
```

```
% Se realizeaza un vector care ordoneaza crescator suma tolerantelor
```

```
% se ordoneaza intai suma, apoi sumele se ordoneaza in functie de ordinul M
```

```
vector = [suma1,suma2,suma3,suma4,suma5,suma6,suma7,suma8,suma9];
```

```
vector_ordonat = sort(vector); % ordonare in functie de suma:
```

```
% suma1,suma3,suma2,suma4,suma6,suma7,suma9,suma5,suma8
```

```
% Daca ordonam in functie de filtrul de mai sus, ordinea descrescatoare a
```

```
% performantei filtrelor va fi:
```

```
% filtru1,filtru3,filtru2,filtru4,filtru6,filtru7,filtru9,filtru5,filtru8
```

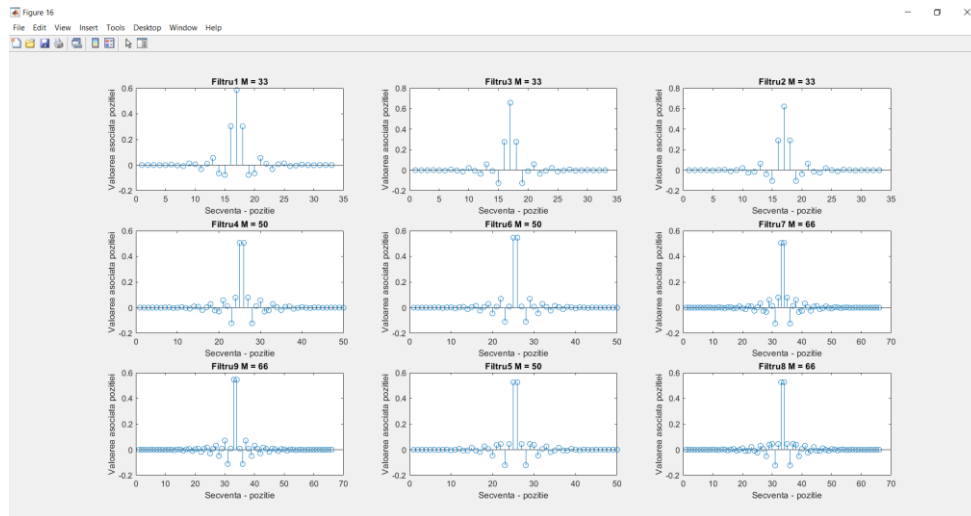


Figura 16 - secvențele pondere ale filtrelor, enumerate pe linii, în ordinea descrescătoare a performanței

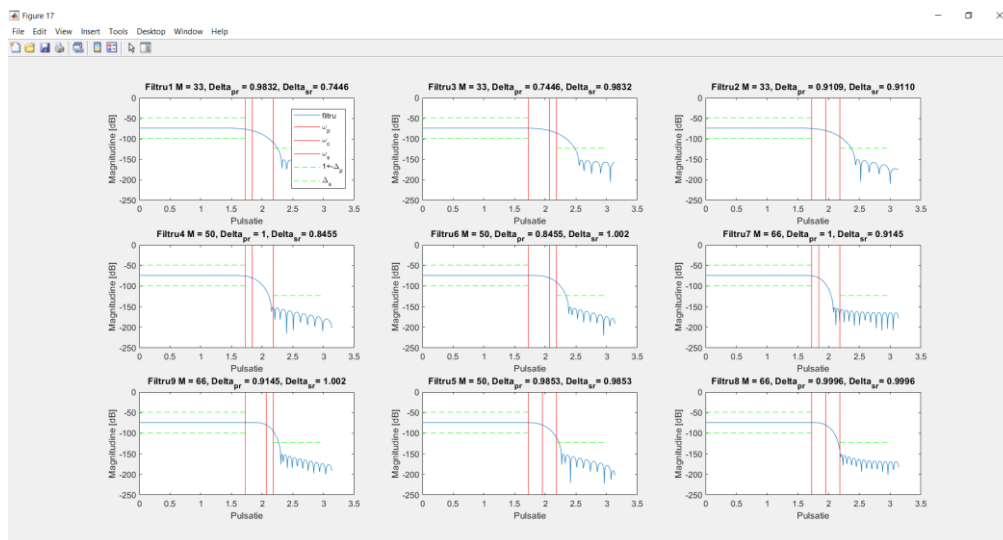


Figura 17 - spectrele filtrelor enumerate pe linii, in ordinea descrescatoare a performantei

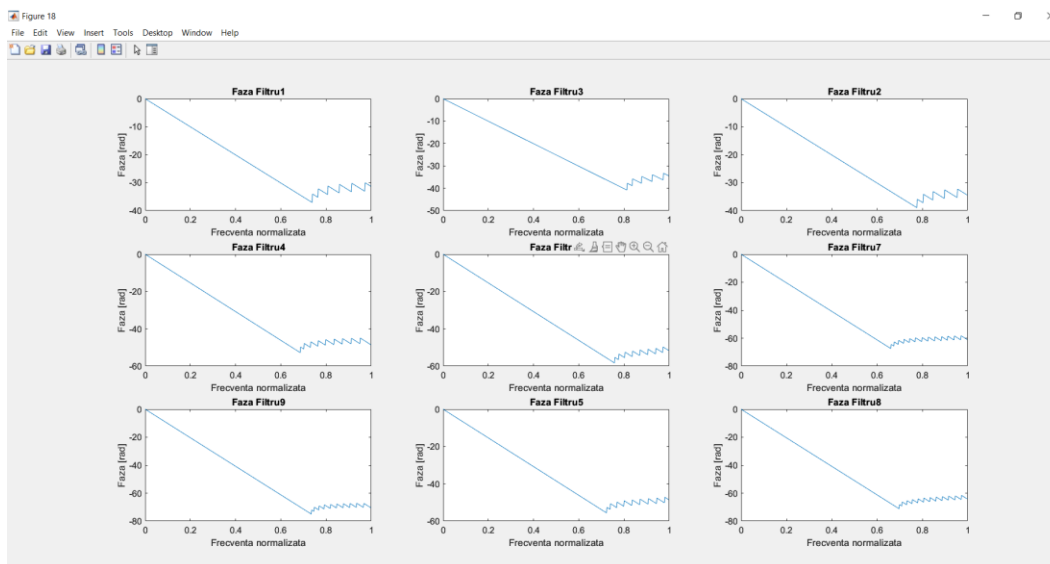


Figura 18 - fazele corespunzatoare ale filtrelor

Faza 4 - Concurs de proiectare

Pentru realizarea cu succes a acestei faze, au fost realizate calcule minutioase cu ajutorul programului MATLAB, de ajutor fiind, de asemenea, o functie asemanatoare cu cea precedenta, insa putin modificata.

```

function [filtru,M_aux,omega_aux,verificare_toleranta] = Faza4_concurs(M,tip,omega_c,omega_p,ome
% aux este folosit pentru cazurile in care poate exista un parametru aditional
% daca aux = 0 => nu exista parametru
% aux!= 0 => exista parametru

% Se verifica tipul ferestrei => se formeaza fereastra propriu zisa
if aux == 0
    if tip == "dreptunghiulara"
        window = boxcar(M);
    elseif tip == "triunghiulara"
        window = triang(M);
    elseif tip == "blackman"
        window = blackman(M);
    elseif tip == "hamming"
        window = hamming(M);
    elseif tip == "hanning"
        window = hanning(M);
    else
        fprintf("Greseala!");
    end
end

if aux ~= 0
    if tip == "cebisev"
        window = chebwin(M,aux);
    elseif tip == "kaiser"
        window = kaiser(M,aux);
    elseif tip == "tuckey"
        window = tukeywin(M,aux);
    elseif tip == "lanczos"
        window = functie_fereastra_lanczos(M,aux);
    else
        fprintf("Greseala!");
    end
end

% Normare frecventa
freq_c=omega_c/pi;

% Se realizeaza filtrul ferestrei cu ajutorul functiei fir1
filtru = fir1(M-1,freq_c>window);

% Se afla tolerantele, respectiv suma dintre acestea
[Delta_pr,Delta_sr] = faza3_puncta(filtru,omega_p,omega_s);

% Afisari

fprintf("Suma dintre /Delta_p_r si /Delta_p_s este %f\n\n", Delta_pr + Delta_sr);

if aux == 0
    if tip == "dreptunghiulara"
        fprintf("Filtru de ordin %d realizat cu fereastra dreptunghiulara\n\n",M);
    elseif tip == "triunghiulara"
        fprintf("Filtru de ordin %d realizat cu fereastra triunghiulara\n\n",M);
    elseif tip == "blackman"
        fprintf("Filtru de ordin %d realizat cu fereastra blackman\n\n",M);
    elseif tip == "hamming"

```

```

elseif tip == "hamming"
    fprintf("Filtru de ordin %d realizat cu fereastră hamming\n\n",M);
elseif tip == "hanning"
    fprintf("Filtru de ordin %d realizat cu fereastră hanning\n\n",M);
end
end

if aux ~= 0
    if tip == "cebisev"
        fprintf("Filtru de ordin %d realizat cu fereastră Cebisev, parametrul additional avand val
elseif tip == "kaiser"
        fprintf("Filtru de ordin %d realizat cu fereastră Kaiser, parametrul additional avand val
elseif tip == "tuckey"
        fprintf("Filtru de ordin %d realizat cu fereastră Tuckey, parametrul additional avand val
elseif tip == "lanczos"
        fprintf("Filtru de ordin %d realizat cu fereastră Lanczos, parametrul additional avand va
    end
end

elseif tip == "tuckey"
    fprintf("Filtru de ordin %d realizat cu fereastră Tuckey, parametrul additional avand val
elseif tip == "lanczos"
    fprintf("Filtru de ordin %d realizat cu fereastră Lanczos, parametrul additional avand va
end
end

verificare_toleranta = Faza3_punctb2(Delta_pr,Delta_p,Delta_sr,Delta_s);
M_aux = M;
omega_aux = omega_c;
...

```

Pentru gasirea celui mai bun filtru, trebuie variate ordinul M, pulsatia de taiere omega_c si tipul de fereastră, plus parametrul de configurare a ferestrei, daca are.

```
% faza de taiere se va lua cel mai apropiat
```

```
% se va alege M cu valori in intervalul inchis [2, 101]; % nu este
% specificat nimic in legatura cu intervalul de alegere a lui M
% omega_c in intervalul (0,3.14]
```

Variatia parametrilor auxiliari a fost realizata in functie de instructiunile plasate in inceputul documentului printru studenti: r intre 80 dB si 100 dB, beta intre 2 dB si 10 dB, L intreg intre 2 si 7, alfa intre 20% si 80%.

```

% Trebuie variate: ordinul M, pulsatia de taiere omega_c si tipul de fereastră, plus param
% de configurare a ferestrei, daca are.
% Din aceasta faza se intelege => simularea filtrelor ferestrelor si
% cautarea celei mai optime situatii pentru fiecare, realizarea ulterioara
% a unui "podium", evidentierea primelor 3 optiuni

```

Mecanism de lucru:

Se vor realiza 2 for-uri (unul pentru M, altul pentru omega_c), se va seta timpul ferestrei si auxiliarul (0 daca nu exista parametru, diferit de 0 daca exista) si se vor apela 2 functii: una

pentru aflarea lui Δ_{pr} si Δ_{sr} , si cealalta pentru aflarea filtrului, valorilor M , sumei, si a lui ω_c . Acestea sunt apoi stocate in niste vectori, iar din interiorul vectorului de sume se extrage cea mai mica. Ulterior, se afla pozitia la care se afla M minim si ω_c minim. Se verifica conditia in ceea ce priveste toleranta, iar daca aceasta este respectata, se obtine filtrul optim ferestrei.

In urma multelor calcule, s-au obtinut urmatoarele rezultate optime clasificate in functie de suma minima:

```
cebishev : % omega_c = 1.95, M = 98, suma = 0.000010 , r = 94
kaiser : % omega_c = 1.97, M = 100, suma = 0.000019, beta=10
lanczos : % omega_c = 1.95, M = 100, suma = 0.00012
```

$L = 4$

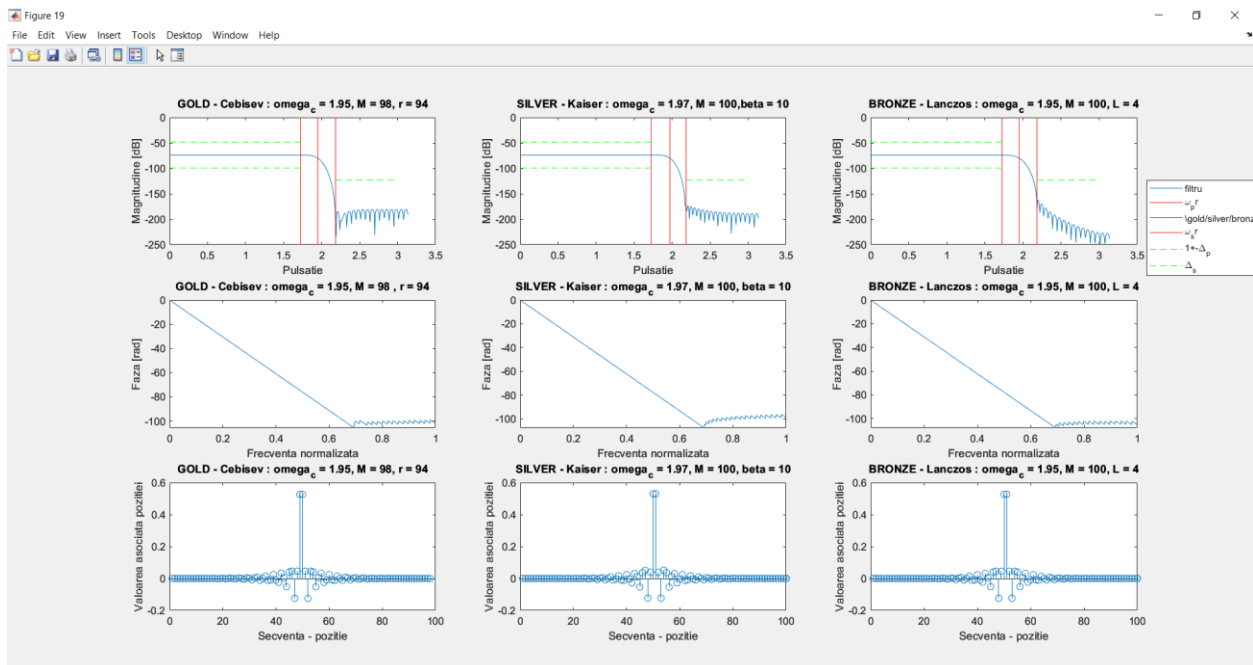


Figura 19 - spectrele filtrelor GOLD, SILVER, BRONZE pe prima linie, fazele aferente pe linia a doua si secventele pondere aferente pe linia a treia.

Faza 5 – Optionala

Asemanator Faza 4, doar ca s-au modificat anumite conditii.

```
vector_aux1 = [];  
vector_aux2 = [];  
vector_aux3 = [];  
vector_aux4 = [];  
%primul vector => stocam sumele, al 2-lea => stocam M, al 3-lea => stocam omega_c  
for M = 2:1:80 % initial  
    for omega_c = 0.1:0.1:3.14  
        for aux = 80:1:100  
            tip = "cebisev";  
  
            [filtru_faza4,M_aux,omega_c_aux,~] = Faza4_concurs(M,tip,omega_c,omega_p,omega_  
            [Delta_pr_aux,Delta_sr_aux] = faza3_puncta(filtru_faza4,omega_p,omega_s);  
            [F, ~] = freqz(filtru_faza4,5000);  
            absolut = abs(F);  
            K_p = find(absolut >= (1-Delta_p),1, 'last');  
            K_s = find(absolut <= Delta_s,1,'first');  
            omega_pr=om_window(K_p);  
            omega_sr=om_window(K_s);  
  
            omega_sr=om_window(K_s);  
            banda = omega_sr - omega_pr;  
            vector_aux1 = [vector_aux1,banda];  
            vector_aux2 = [vector_aux2,M_aux];  
            vector_aux3 = [vector_aux3,omega_c_aux];  
            vector_aux4 = [vector_aux4,aux];  
        end  
    end  
end  
  
% se afla banda minima din valorile stocate in vector_aux  
banda_minima_cebisev = min(vector_aux1);  
% se afla pozitia pe care se afla banda_minima => aflam M minim  
pozitie = find(vector_aux1 == banda_minima_cebisev);  
pozitie = pozitie(1);  
M_minim_cebisev = vector_aux2(pozitie);  
omega_minim_cebisev = vector_aux3(pozitie);  
r_minim = vector_aux4(pozitie);
```

S-a lucrat exclusiv cu fereastra Cebisev, deoarece aceasta s-a clasat pe primul loc in cadrul analizei anterioare. Ordinul M, totusi, a avut valori cuprinse in intervalul [2,100], iar omega_c a avut pasul de 0.01.

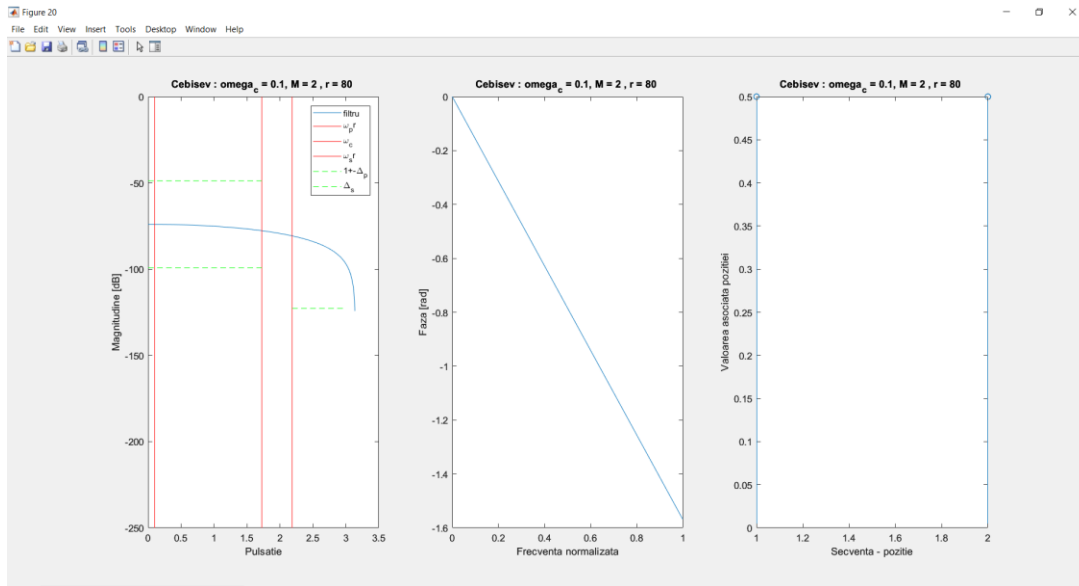


Figura 20 - spectrul filtrului optimal propus, faza aferenta si secventa pondere