



**ACSE**  
automatic control and  
systems engineering



FACULTATEA DE  
**AUTOMATICĂ ȘI  
CALCULATOARE**

# Proiectarea filtrelor FIR prin optimizare

PROIECT PS

Autor: Stanciuc Vlad-Nicolae | Prelucrarea Semnalelor | Ianuarie 2022

Coordonatori: Profesor: Dan ȘTEFĂNOIU

Îndrumător de laborator: Conf. Alexandru DUMITRASCU

Anul universitar: 2021-2022

## Sumar

### Capitolul I

- Obiectivele Proiectului ..... 1

### Capitolul II

- Pașii ce trebuie efectuați pentru atingerea obiectivelor ..... 2

### Capitolul III

- Modul de abordare și rezultatele de simulare ..... 2

- **Faza 1**

- Subpunctul a) ..... 4
- Subpunctul b) ..... 6
- Subpunctul c) ..... 7
- Subpunctul d) ..... 9
- Subpunctul e) ..... 12

- **Faza 2**

- Subpunctul a) ..... 18
- Subpunctul b) ..... 21
- Subpunctul c) ..... 25

- Concluzii** ..... 33

- Bibliografie** ..... 33

## Capitolul I

- **Obiectivele proiectului**

În cadrul acestui proiect vrem să studiem proiectarea în practică a filtrelor FIR (Finite Impulse Response) prin optimizare (FTJ, FTS, FTB).

Proiectarea acestor filtre se realizează cu ajutorul a 2 metode, prima are la bază criteriul patratic, adică în sens CMMP (a normei euclidiene), iar a doua are la bază criteriul Cebîșev (min-max), adică algoritmul Parks-McClellan (în sensul normei infinit).

Vom utiliza și familiariza cu anumite funcții ce nu au restricții de fază. Aceste funcții ne vor ajuta să realizăm cele 3 tipuri de filtre asemănătoare cu metoda **firls**, prestabilită în Matlab.

În final se dorește proiectarea filtrelor FIR optime utilizând MCMMP și o mulțime finită de pulsații predefinite.

## Capitolul II

- **Pașii ce trebuie efectuați pentru atingerea obiectivelor**

1. Primul pas este înțelegerea filtrelor, mai exact ce este un FTS(Filtru Trece-Sus), un FTB(Filtru Trece-Banda) și un FTJ(Filtru Trece-Jos).Ce este mai important este intelegerea functionalitatilor si modul de utilizare al acestora.Pentru aceasta, am revenit la cursurile de SS si TSM din anul II si prin participarea la cursul de PS.
2. Al doilea pas este documentarea privind modul in care putem proiecta aceste filtre in Matlab, deci trebuie sa ne familiarizam cu functiile firls si firpm, acestea fiind cele 2 metode de proiectare.Pentru aceasta am consultat laboratorul si internetul.
3. Cel de al treilea pas este înțelegerea diferențelor între cele două metode.Acest pas a fost inteles in totalitate in timpul realizarii proiectului.
4. Al patrulea pas: Să înțelegem că pe lângă metodele de mai sus, FTS și FTB se pot realiza și prin folosirea unuia/mai multor FTJ. Trebuie înțeles la ce se referă formulele  $FTS = 1 - FTJ$ ,  $FTB = FTJ - FTJ$  și cum se pot implementa acestea.

## Capitolul III

- **Modul de abordare și rezultatele de simulare**

### Faza 1

Considerați problema de proiectare a unui FTJ cu răspunsul dorit (5.1), în care  $\omega_p$  și  $\omega_s$  au valori fixate.

Valorile generate sunt:

$\omega_p = 0.3920$ ;

$\omega_s = 0.5316$ ;

$M = [10,13,26,28,75,79,100]$

### Subpunctul a)

Pentru rezolvarea acestui subpunct am utilizat funcția **firls**, ce ne permite sa proiectam filtre FIR optime, în sens CMMP, adică al normei euclidiene. Funcția **firls** se utilizeaza astfel: **firls(M,W,A,w)** unde parametrii folosiți sunt:

„**M**” – ordinul filtrului (recomandat este să fie par),

„**W**” - vector ce conține pulsațiile normalizate reprezentând benzile de interes (coordonata x),

„**A**” - vector ce descrie valorile răspunsului dorit în benzile de interes (coordonata y), iar

„**w**” – este opțional, reprezintă ponderile și standard are valoarea [1 1].

Trebuie să proiectăm un FTJ așa că parametrii „**W**” și „**A**” au valori fixe și anume:

$W = [0 \ \omega_p \ \omega_s \ 1]$

$A = [1 \ 1 \ 0 \ 0]$ .

Am folosit o buclă for pentru a parcurge tot vectorul „**M**” și am calculat funcția pondere a filtrului folosind funcția descrisă mai sus, după care am calculat răspunsul în frecvență și vectorul frecvențelor folosind funcția **freqz(h,1,1000)**, unde **h** este funcția pondere, iar 1000 sunt numărul de linii spectrale.

După calcularea răspunsului în frecvență, am folosit funcția **plot** pentru a trasa caracteristica de frecvență, în dB. Transformarea în dB s-a realizat folosind formula:  $20 \cdot \log(\text{abs}(H))$ , unde **abs** este valoarea absolută, iar **H** este răspunsul în frecvență al filtrului.

Pentru a observa fazele fiecărui grafic am calculat faza și faza folosind funcția **phasez(h,1,1000)** ce are aceeași parametri ca și **freqz**. Trasarea graficului fazei s-a realizat cu funcția **plot**.

Am folosit funcția **plot** pentru afișare deoarece aceasta trasează fiecare răspuns, respectiv fază folosind culori diferite și astfel se pot evidenția filtrele pentru fiecare ordin în parte.

Codul folosit este:

```
%Proiect PS   Proiectarea filtrelor FIR prin optimizare
```

```
%Nume: Stanciuc
```

```
%Prenume: Vlad-Nicolae
```

```
ng = 2;
```

```
ns = 12;
```

```
%Obținerea datelor initiale
```

```
[omega_p,omega_s,M] = PS_Prj_2_Faza_lab(ng,ns);
```

```
%Salvare omega_p si omega_s impartite la pi
```

```
omega_p = 0.3920;
```

```
omega_s = 0.5316;
```

```
%Notare automata a legendei pe grafice
```

```
legendString = 'M = ' + string(M);
```

```
%Subpunctul a.
```

```
%Declaram parametrii necesari pentru apelarea firls
```

```
W = [0 omega_p omega_s 1]; %W contine benzile de interes
```

A = [1 1 0 0]; %A contine valorile raspunsului dorit in benzile de interes

%Matricea numarul 1

figure('Name', 'Faza 1 - a - {omega\_p, omega\_s, M}');

for i = 1:1:7

    %Spectrul filtrului in coordonate liniare

    subplot(3, 1, 1);

    h = firls(M(i)-1,W,A); %functia pondere a FTJ

    plot(h);

    xlabel('Pulsatiile normalizate');

    ylabel('Raspunsul dorit in benzile de interes');

    title('Spectrul filtrului in coordonate liniare - metoda LS');

    hold on;

    %Spectrul filtrului in dB

    subplot(3, 1, 2);

    h = firls(M(i)-1,W,A);

    [H,omega] = freqz(h,1,1000); %raspunsul in frecventa cu o rezolutie de peste 1000 de puncte

    plot(omega, 20\*log(abs(H))); %trasarea caracteristicii de frecventa

    xlabel('Vectorul de frecvente');

    ylabel('Amplitudinea masurata in dB');

    title('Spectrul filtrului in dB - metoda CMMP');

    hold on;

    legend(legendString);

    %Faza filtrului

    subplot(3, 1, 3);

    [faza, omega] = phasez(h,1,1000); %faza filtrului cu o rezolutie de peste 1000 de puncte

    plot(omega, faza); %trasarea graficului fazei

    xlabel('Vectorul de frecvente');

    ylabel('Faza');

    title('Faza filtrului');

    hold on;

    legend(legendString);

end

%Matricea numarul 2

pi\_minus\_omega\_p = (pi - omega\_p) / pi;

```
pi_minus_omega_s = (pi - omega_s) / pi;
```

```
figure('Name', 'Faza 1 - a - {pi-omega_s, pi-omega_p, M}');
```

```
W1 = [0 pi_minus_omega_s pi_minus_omega_p 1];
```

```
A = [1 1 0 0];
```

```
for i = 1:1:7
```

```
    % %Spectrul filtrului in coordonate liniare
```

```
    subplot(3, 1, 1);
```

```
    h = firls(M(i)-1,W1,A);
```

```
    plot(h);
```

```
    xlabel('Pulsatiile normalizate');
```

```
    ylabel('Raspunsul dorit in benzile de interes');
```

```
    title('Spectrul filtrului in coordonate liniare - metoda  
CMMF');
```

```
    hold on;
```

```
    %Spectrul filtrului in dB
```

```
    subplot(3, 1, 2);
```

```
    h = firls(M(i)-1,W1,A); %functia pondere a FTJ
```

```
    [H,omega] = freqz(h,1,1000); %raspunsul in frecventa cu o  
rezolutie de peste 1000 de puncte
```

```
    plot(omega, 20*log(abs(H))); %trasarea caracteristicii de  
frecventa
```

```
    xlabel('Vectorul de frecvente');
```

```
    ylabel('Amplitudinea masurata in dB');
```

```
    title('Spectrul filtrului in dB - metoda LS');
```

```
    hold on;
```

```
    legend(legendString);
```

```
    %Faza filtrului
```

```
    subplot(3, 1, 3);
```

```
    [faza, omega] = phasez(h,1,1000); %faza filtrului cu o  
rezolutie de peste 1000 de puncte
```

```
    plot(omega, faza); %trasarea graficului fazei
```

```
    xlabel('Vectorul de frecvente');
```

```
    ylabel('Faza');
```

```
    title('Faza filtrului - metoda LS');
```

```
    hold on;
```

```
    legend(legendString);
```

```
end
```

```
saveas(gcf, 'Faza1A{PI-omega_p,PI-omega_s,M}.png');
```

```
%Matricea numarul 3
```

```
figure('Name', 'Faza 1 - a - Secventele pondere suprapuse -  
metoda LS');
```

```
for i = 1:1:7
```

```
subplot(7, 1, i);
h = firls(M(i)-1,W,A);

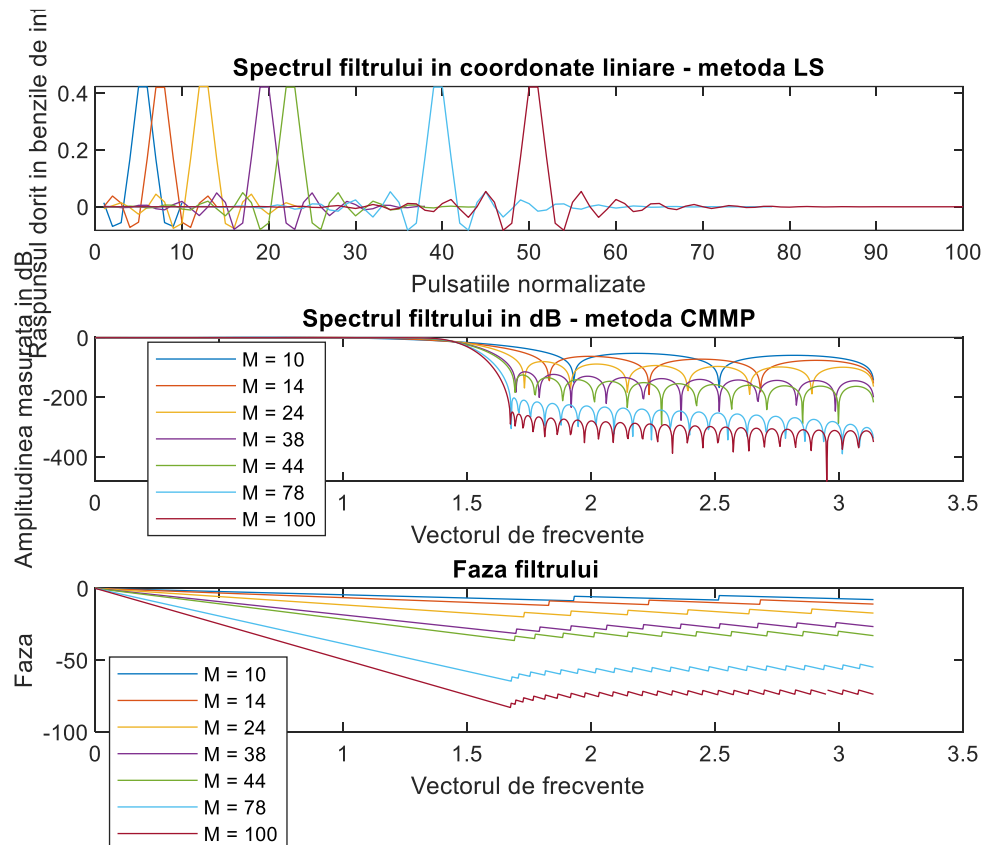
t = 0:1:(M(i)-1);
stem(t, h);

hold on;
h = firls(M(i)-1,W1,A);

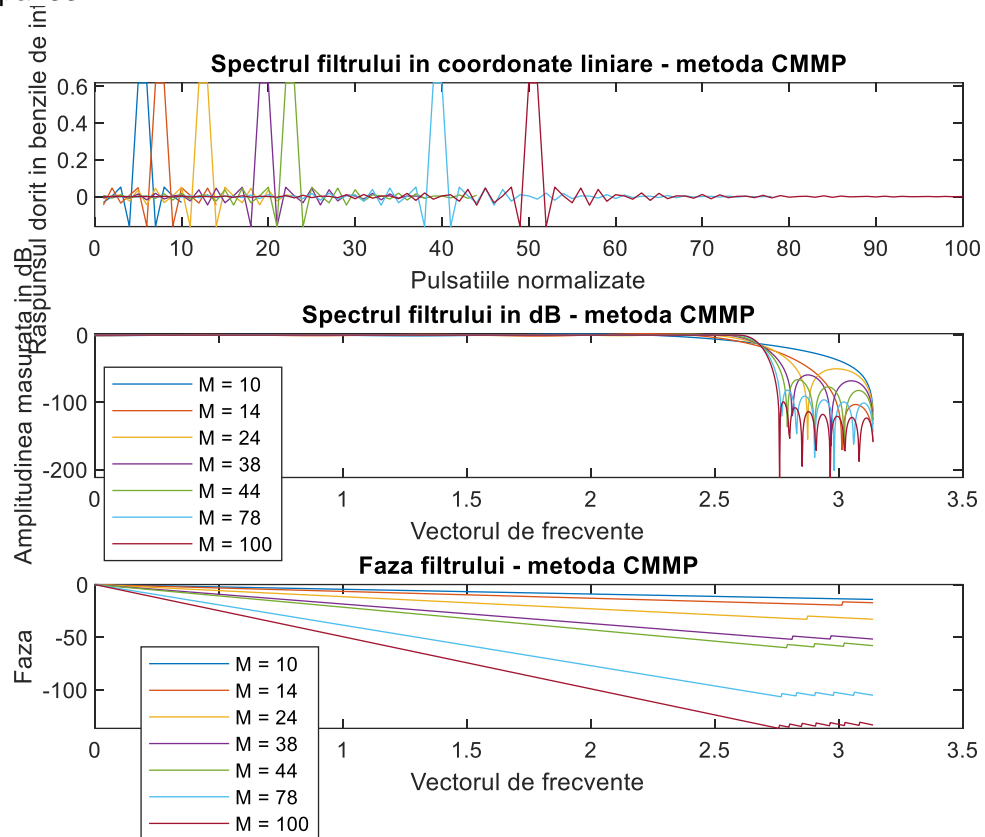
stem(t, h);

hold on;
end
```

După rulare, graficele obținute sunt:

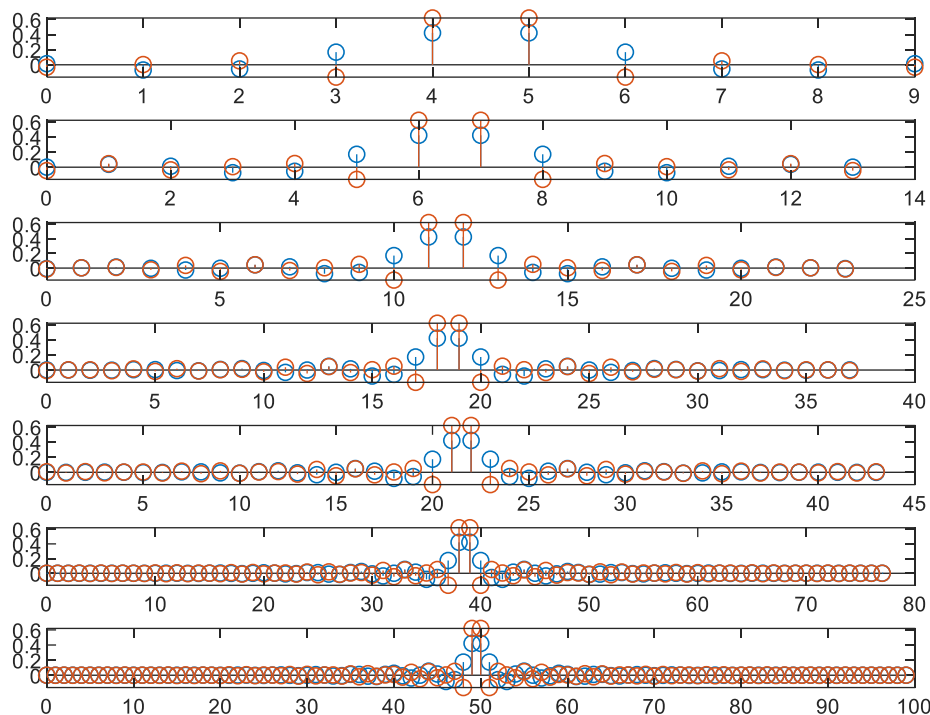


Figură 1: Răspunsurile în frecvență ale unor filtre de tip Filtru Trece-Jos construite în sens CMMP (norma 2) de ordine diferite cu  $\omega_p$  și  $\omega_s$  date



Figură 2: Răspunsurile în frecvență ale unor filtre de tip Filtru Trece-Jos construite în sens CMMP (norma 2) de ordine diferite cu  $\omega_p$  și  $\omega_s$  date





Figură 3: Secvențele pondere suprapuse cu metoda CMMP

Din graficul fazei se poate observa că aceasta este liniară până în pulsația de stopare( $\omega_s$ ) indiferent de ordin, iar din graficul caracteristici de frecvență observăm că odată cu creșterea ordinului, filtrul devine mai performant, iar amplitudinea punctelor de minim scade odată cu creșterea frecvenței, ceea ce produce apariția mai multor lobi apropiați.

### Subpunctul b)

Pentru rezolvarea acestui subpunct se folosește același cod ca cel precedent, însă pentru a proiecta filtre în sensul normei infinit se va înlocui funcția **firls** cu funcția **firpm** ce se utilizează identic, cu mențiunea că ordinul **M** poate fi și impar.

Și la acest punct se trasează caracteristica de frecvență și faza în interiorul unei bucle repetitive **for**, utilizând funcția **plot** pentru diferențierea fiecărui filtru de ordin diferit.

Codul utilizat este:

```
omega_p = 0.3920;
```

```
omega_s = 0.5316;
```

```
W = [0 omega_p omega_s 1];
```

```
A = [1 1 0 0];
```

```
w = [1 1];
```

```
for i = 1:1:7
```

```
    %Spectrul filtrului in coordonate liniare
```

```
    subplot(3, 1, 1);
```

```
    h = firpm(M(i)-1,W1,A,w); %functia pondere a FTJ
```

```
    plot(h);
```

```
    xlabel('Pulsatiile normalizate');
```

```
    ylabel('Raspunsul dorit in benzile de interes');
```

```
    title('Spectrul filtrului in coordonate liniare');
```

```
    hold on;
```

```
    %Spectrul filtrului in dB
```

```
    subplot(3, 1, 2);
```

```
    h = firpm(M(i)-1,W1,A,w); %functia pondere a FTJ
```

```
    [H,omega] = freqz(h,1,1000); %raspunsul in frecventa cu o  
rezolutie de peste 1000 de puncte
```

```
    plot(omega, 20*log(abs(H))); %trasarea caracteristicii de  
frecventa
```

```
    xlabel('Vectorul de frecvente');
```

```
    ylabel('Amplitudinea (dB)');
```

```
    title('Caracteristica de frecventa a FTJ - Norma infinit  
(Algoritmul Parks-McClellan)');
```

```
    hold on;
```

```
    legend(legendString);
```

```
    %Faza filtrului
```

```
    subplot(3, 1, 3);
```

```
    [faza, omega] = phasez(h,1,1000); %faza filtrului cu o  
rezolutie de peste 1000 de puncte
```

```
    plot(omega, faza); %trasarea graficului fazei
```

```
    xlabel('Frecventa');
```

```
    ylabel('Faza');
```

```
    title('Faza FTJ - Norma infinit (Algoritmul Parks-  
McClellan)');
```

```
    hold on;
```

```
    legend(legendString);
```

```
end
```

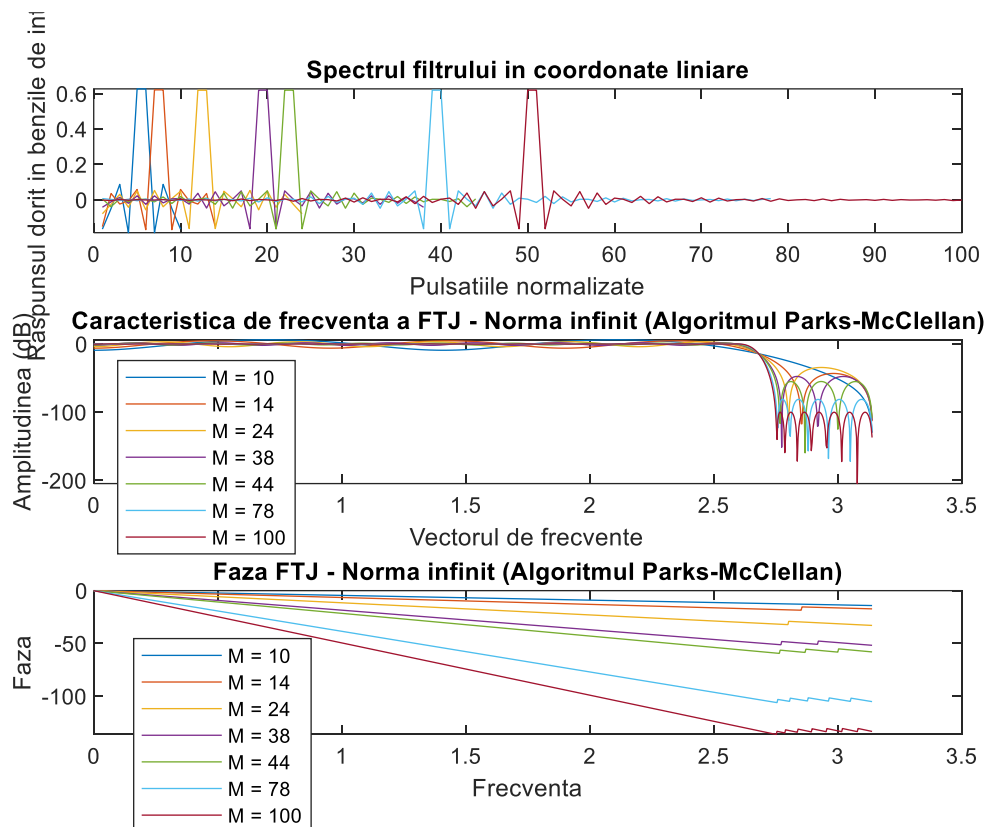
```
saveas(gcf, 'Faza1-B.png');
```

```
%Matricea numarul 3
figure('Name', 'Faza 1 - a - Secventele pondere suprapuse -
metoda PM');
for i = 1:1:7

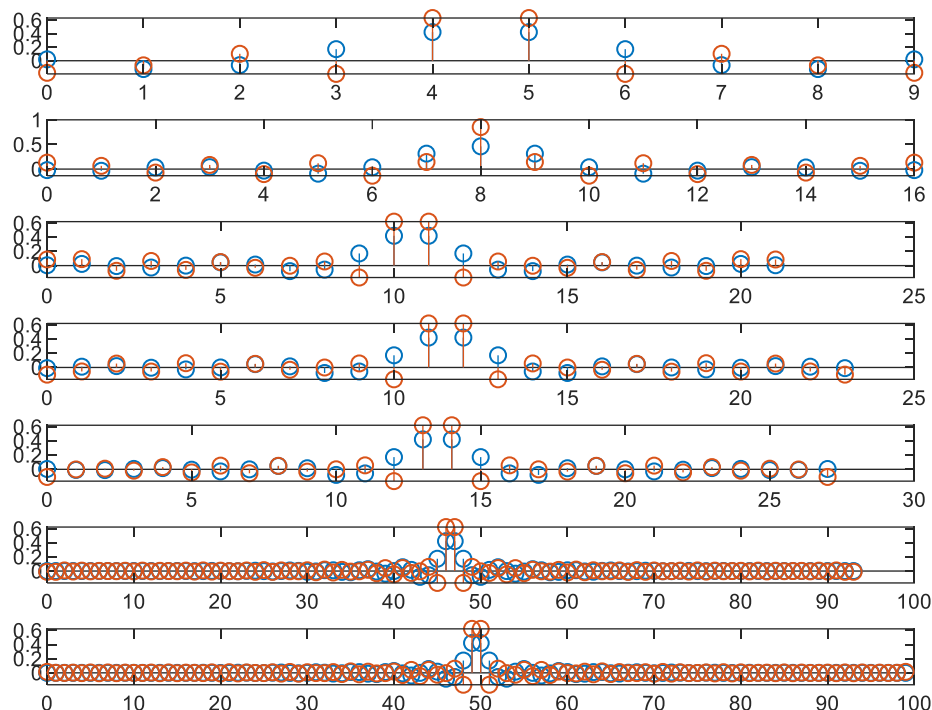
    subplot(7, 1, i);
    h = firpm(M(i)-1,W,A,w);
    t = 0:1:(M(i)-1);
    stem(t, h);
    hold on;
    h = firpm(M(i)-1,W1,A,w);
    stem(t, h);
    hold on;
end

saveas(gcf, 'Faza1-B.png');
```

Graficele obținute sunt:



Figură 4: Răspunsurile în frecvență ale unor filtre de tip Filtru Trece-Jos construite în sens CMMP (norma 2) de ordine diferite cu  $\omega_p$  și  $\omega_s$  date



Figură 5: Secvențele pondere suprapuse cu metoda CMMP

La fel ca anterior observăm că mărirea ordinului face ca filtrul să devină mai performant, iar faza este liniară până în pulsația de stopare ( $\omega_s$ ).

### Subpunctul c)

Pentru acest subpunct  $M$  a fost generat și are valoarea fixă 38;

Se calculează secvența pondere  $h$  folosind atât **firls**, cât și **firpm**, după care se calculează răspunsul în frecvență  $H$  pentru fiecare sens în parte (sensul normei euclidiene, respectiv sensul normei infinite) folosind funcția **freqz**. Faza s-a realizat cu ajutorul funcției din Matlab **phasez**.

La acest subpunct folosim funcția „find” pentru a afla indicii corespunzători valorilor din banda de stopare a răspunsului în frecvență. Cu acești indici, într-o buclă **for**, construim vectorul ce va conține frecvențele **doar** din banda de stopare, utilizat ulterior pentru a calcula atenuarea minimă. Valorile vectorului obținut se transformă în dB, iar maximul este atenuarea minimă.

Codul folosit este:

```
M = PS_Prj_2_Faza_1c(2,12) ;
omega_p = 0.3920;
omega_s = 0.5316;
```

```
W = [0 omega_p omega_s 1]; %vector ce contine pulsatiile
normalizate
A = [1 1 0 0]; %vector ce descrie valorile raspunsului dorit
in benzile de interes (coordonata y)
figure('Name','Faza 1-{omega_s,omega_p} - C');

subplot(2,1,1);
h_norma2 = firls(M-1,W,A); %functia pondere a FTJ
[H_norma2,omega_norma2] = freqz(h_norma2,1,120); %raspunsul
in frecventa; rezolutia 120 linii spectrale
plot(omega_norma2,20*log(abs(H_norma2))); %trasarea
caracteristicii de frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTJ');
hold on;

ind_norma2 = find(omega_norma2>(omega_s*pi)); %subvector
corespunzator benzii de stopare

for i = 1 : length(ind_norma2)

    n = ind_norma2(i); %indicii corespunzatori pentru a crea
vectorul corespunzator benzii de stopare
    Hstop_norma2(i) = H_norma2(n); %frecvente din vectorul
frecventelor in banda de stopare

end

h_normainf = firpm(M-1,W,A);
[H_normainf,omega_normainf] = freqz(h_normainf,1,120);
%raspunsul in frecventa cu rezolutie de 120 linii spectrale
plot(omega_normainf,20*log(abs(H_normainf))); %trasarea
caracteristicii de frecventa
legend('Norma 2','Norma infinit'); %legenda
hold off;

subplot(2,1,2);
[faza_norma2, omega_norma2] = phasez(h_norma2,1,120); %faza
filtrului; rezolutia 120 de linii spectrale
plot(omega_norma2, faza_norma2); %trasarea graficului fazei
hold on;

[faza_normainf, omega_normainf] = phasez(h_normainf,1,120);
%faza filtrului
plot(omega_normainf, faza_normainf); %afisarea graficului
fazei
xlabel('Frecventa');
```

```
ylabel('Faza');
```

```
title('Faza FTJ');
```

```
legend('Norma 2','Norma infinit');
```

```
hold on;
```

```
saveas(gcf,'Faza1-C.png'); %salvarea imaginii
```

```
ind_normainf = find(omega_normainf > (omega_s*pi)); %subvector  
corespunzator benzii de stopare
```

```
for i = 1 : length(ind_normainf)
```

```
    n = ind_normainf(i); %indicii corespunzatori pentru a  
    crea vectorul corespunzator benzii de stopare
```

```
    Hstop_normainf(i) = H_normainf(n); %frecvente din  
    vectorul frecventelor in banda de stopare
```

```
end
```

```
    APM = sprintf("RPM %5.5g",  
max(20*log(abs(Hstop_normainf))))
```

```
    ALS = sprintf("RPM %5.5g", max(20*log(abs(Hstop_norma2))))
```

```
    %pentru {pi-omega_s,pi-omega_p}.
```

```
    pi_minus_omega_p = (pi - omega_p)/pi;
```

```
    pi_minus_omega_s = (pi - omega_s)/pi ;
```

```
W1 = [0 pi_minus_omega_s pi_minus_omega_p 1]; %vector ce  
contine pulsatiile normalizate
```

```
A = [1 1 0 0]; %vector ce descrie valorile raspunsului dorit  
in benzile de interes
```

```
figure('Name','Faza 1-{pi-omega_s,pi-omega_p} - C');
```

```
subplot(2,1,1);
```

```
h_norma2 = firls(M-1,W1,A); %functia pondere a FTJ
```

```
[H_norma2,omega_norma2] = freqz(h_norma2,1,120); %raspunsul  
in frecventa
```

```
plot(omega_norma2,20*log(abs(H_norma2))); %afisarea  
caracteristicii de frecventa
```

```
xlabel('Vectorul frecventelor'); %denumirea axelor
```

```
ylabel('Amplitudinea (dB)');
```

```
title('Caracteristica de frecventa a FTJ');
```

```
hold on;
```

```
ind_norma2 = find(omega_norma2 > (pi_minus_omega_s*pi));  
%subvector corespunzator benzii de stopare
```

```
for i = 1 : length(ind_norma2)

    n = ind_norma2(i); %indicii corespunzatori pentru a crea
    vectorul corespunzator benzii de stopare
    Hstop_norma2(i) = H_norma2(n); %frecvente din vectorul
    frecventelor in banda de stopare

end

h_normainf = firpm(M-1,W1,A); %functia pondere a FTJ
[H_normainf,omega_normainf] = freqz(h_normainf,1,120);
%raspunsul in frecventa
plot(omega_normainf,20*log(abs(H_normainf))); %afisarea
caracteristicii de frecventa
legend('Norma 2','Norma infinit');
hold off;

subplot(2,1,2);
[faza_norma2, omega_norma2] = phasez(h_norma2,1,120); %faza
filtrului
plot(omega_norma2, faza_norma2); %afisarea graficului fazei
hold on;

[faza_normainf, omega_normainf] = phasez(h_normainf,1,120);
%faza filtrului
plot(omega_normainf, faza_normainf); %afisarea graficului
fazei
xlabel('Frecventa');
ylabel('Faza');

title('Faza FTJ');
legend('Norma 2','Norma infinit');
hold on;

saveas(gcf,'Faza1-C.png');

ind_normainf = find(omega_normainf>(pi_minus_omega_s*pi));
%subvector corespunzator benzii de stopare

for i = 1 : length(ind_normainf)

    n = ind_normainf(i); %indicii corespunzatori pentru a
    crea vectorul corespunzator benzii de stopare
    Hstop_normainf(i) = H_normainf(n); %frecvente din
    vectorul frecventelor in banda de stopare

end
```

Student: Stanciuc Vlad

IS

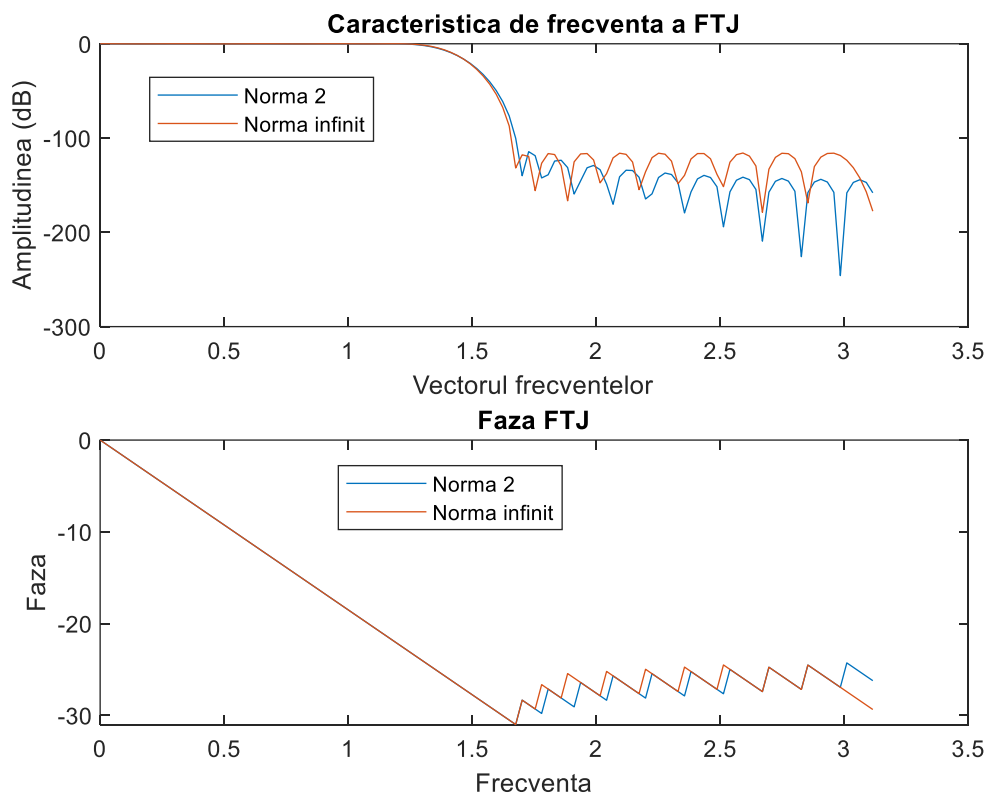
Grupa: 334 AB

```
APM = sprintf("RPM %5.5g%",
```

```
max(20*log(abs(Hstop_normainf))))
```

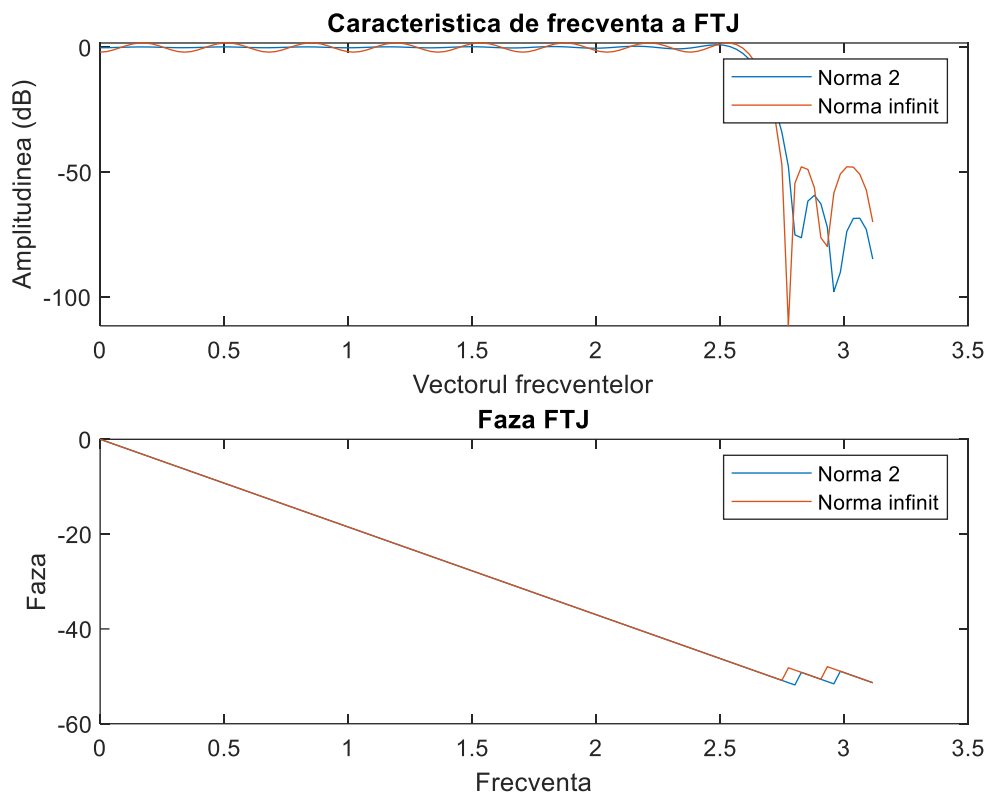
```
ALS = sprintf("RPM %5.5g%", max(20*log(abs(Hstop_norma2))))
```

Graficele obținute sunt:



Figură 6: Raspunsurile in frecventa ale unor filtre de tip Filtru Trece-Jos cu pulsatiile  $\omega_p$  si  $\omega_s$ . Unul construit in sens CMMP, iar celalalt in sensul normei infinit, ambele cu ordinul 38





Figură 7: Raspunsurile in frecventa ale unor filtre de tip Filtru Trece-Jos cu pulsatiile  $\pi\omega_p$  si  $\pi\omega_s$ . Unul construit in sens CMMP, iar celalalt in sensul normei infinit, ambele cu ordinul 38

## Subpunctul d)

Rezolvare: Pentru rezolvarea acestui subpunct se va defini funcția pondere

$\mathbf{w} = [1 \ 5]$  și se vor construi filtre de ordin 38.

Pentru a construi vectorii benzii de trecere, respectiv benzii de stopare am procedat similar ca la subpunctul precedent, folosind funcția **find** pentru a găsi indicii, apoi am folosit o buclă **for** pentru a construi vectorii de interes.

După aflarea vectorilor benzii de trecere și benzii de stop am calculat dispersia erorii atât în banda de stop, cât și în banda de trecere folosind funcția **std**, după care am realizat raportul acestora.

Codul folosit este:

```
%subpunctul d

%definirea parametrilor functiei firls, respectiv firpm
ws = PS_Prj_2_Faza_1d(2,12) ;
M = 55;
omega_p = 0.3920;
omega_s = 0.5316;

W = [0 omega_p omega_s 1]; %vector ce contine pulsatiile normalizate
reprezentand benzile de interes
A = [1 1 0 0]; %vector ce descrie valorile raspunsului dorit in benzile de
interes
w = [1 5]; %ponderile, definesc amplitudinile pentru banda de trecere, respectiv
de stopare

figure('Name','Faza 1 - D - Sens CMMP');

% cu functia pondere [1 5]
subplot(2,1,1);
h_norma2 = firls(M-1,W,A,w); %functia pondere a FTJ
[H_norma2,omega_norma2] = freqz(h_norma2,1,120); %raspunsul in frecventa
plot(omega_norma2,20*log(abs(H_norma2))); %trasareacaracteristicii de frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTJ - Sens CMMP');
hold on;

%fara functia pondere
h_norma2w = firls(M-1,W,A); %functia pondere a FTJ
[H_norma2w,omega_norma2w] = freqz(h_norma2w,1,120); %raspunsul in frecventa
plot(omega_norma2w,20*log(abs(H_norma2w))); %trasarea caracteristicii de
frecventa
legend('Functia pondere [1 5]','Functia pondere [1 1]');
hold off;

subplot(2,1,2);
[faza_norma2, omega_norma2] = phasez(h_norma2,1,120); %faza filtrului
plot(omega_norma2, faza_norma2); %afisarea graficului fazei
hold on;

[faza_norma2w, omega_norma2w] = phasez(h_norma2w,1,120); %faza filtrului
plot(omega_norma2w, faza_norma2w); %afisarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza FTJ - Sens CMMP');
legend('Functia pondere [1 5]','Functia pondere [1 1]');
hold off;

saveas(gcf,'Faza1-D-Norma2.png'); %salvarea figurii

indstop_norma2 = find(omega_norma2>(ws*pi)); %subvector corespunzator benzii de
stopare [omega_s,pi]

for i = 1 : length(indstop_norma2)

    n = indstop_norma2(i); %indicii corespunzatori pentru a crea vectorul
corespunzator benzii de stopare
```

```
Hstop_norma2(i) = H_norma2(n); %frecvente din vectorul frecventelor in banda
de stopare

end

indtrecere_norma2 = find(omega_norma2 < (omega_p*pi)); %subvector corespunzator
benzii de trecere [0,omega_p]

for i = 1 : length(indtrecere_norma2)

    n = indtrecere_norma2(i); %indicii corespunzatori pentru a crea vectorul
    corespunzator benzii de trecere
    Htrecere_norma2(i) = H_norma2(n); %frecvente din vectorul frecventelor in
    banda de trecere

end

figure('Name','Faza 1 - D - Sens Norma Infinit');

subplot(2,1,1);
%cu functia pondere [1 5]
h_normainf = firpm(M-1,W,A,w); %functia pondere a FTJ
[H_normainf,omega_normainf] = freqz(h_normainf,1,120); %raspunsul in frecventa
plot(omega_normainf,20*log(abs(H_normainf))); %afisarea caracteristicii de
frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTJ - Sensul Normei Infinit');
hold on;

%fara functia pondere
h_normainfw = firpm(M-1,W,A); %functia pondere a FTJ
[H_normainfw,omega_normainfw] = freqz(h_normainfw,1,120); %raspunsul in
frecventa
plot(omega_normainfw,20*log(abs(H_normainfw))); %afisarea caracteristicii de
frecventa
legend('Functia pondere [1 5]','Functia pondere [1 1]');
hold off;

subplot(2,1,2);
[faza_normainf, omega_normainf] = phasez(h_normainf,1,120); %faza filtrului
plot(omega_normainf, faza_normainf); %trasarea graficului fazei
hold on;

[faza_normainfw, omega_normainfw] = phasez(h_normainfw,1,120); %faza filtrulu
plot(omega_normainfw, faza_normainfw); %afisarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza FTJ - Sensul Normei Infinit');
legend('Functia pondere [1 5]','Functia pondere [1 1]');
hold off;

saveas(gcf,'Fazal-D-Normainf.png');

indstop_normainf = find(omega_normainf > (omega_s*pi)); %subvector corespunzator
benzii de stopare [omega_s,pi]

for i = 1 : length(indstop_normainf)
```

```

n = indstop_normainf(i); %indicii corespunzatori pentru a crea vectorul
corespunzator benzii de stopare
Hstop_normainf(i) = H_normainf(n); %frecvente din vectorul frecventelor in
banda de stopare

end

indtrecere_normainf = find(omega_normainf < (omega_p*pi)); %subvector
corespunzator benzii de trecere [0,omega_p]

for i = 1 : length(indtrecere_normainf)

    n = indtrecere_normainf(i); %indicii corespunzatori pentru a crea vectorul
corespunzator benzii de trecere
    Htrecere_normainf(i) = H_normainf(n); %frecvente din vectorul frecventelor
in banda de trecere

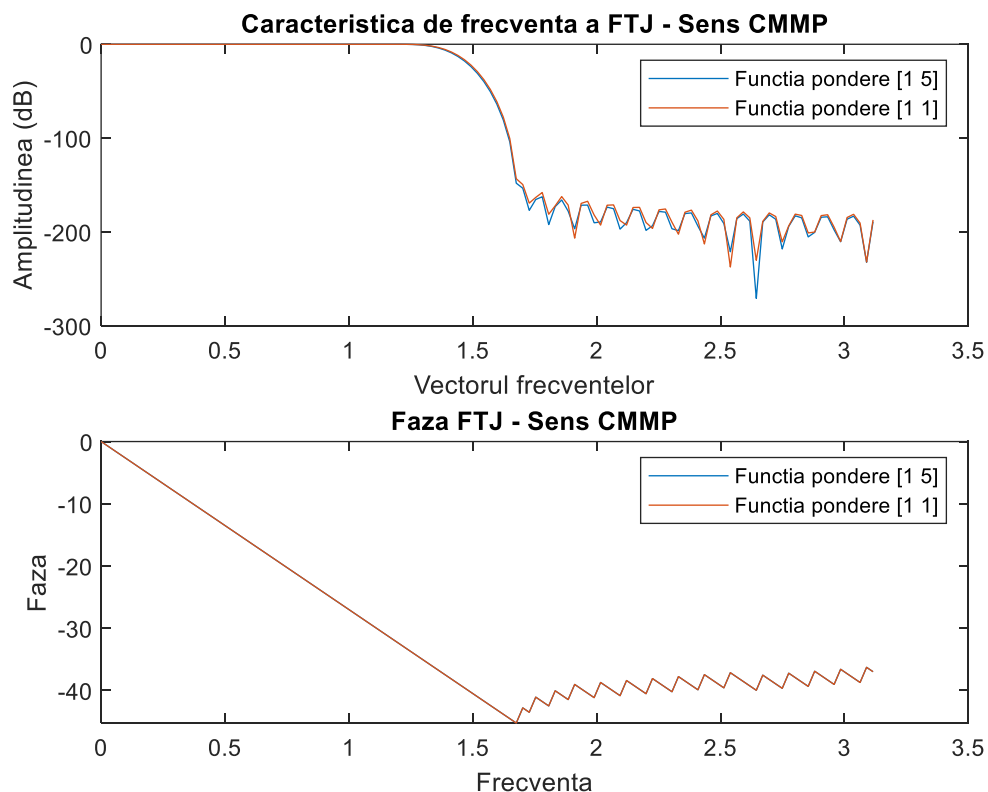
end

Err_banda_stop_normainf = abs(Hstop_normainf); %eroarea din banda de stopare
Err_banda_trecere_normainf = abs(1-Htrecere_normainf); %eroarea din banda de
trecere
Dispersia_PMC = std(abs(Hstop_normainf))/std(abs(1-Htrecere_normainf))

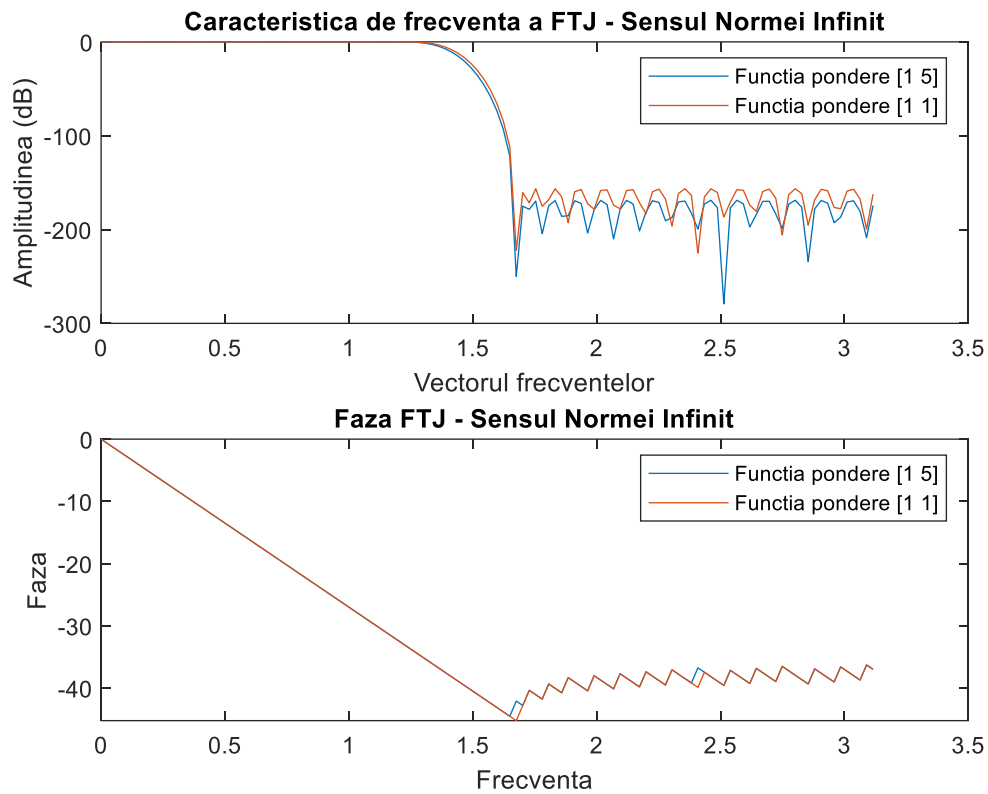
Err_banda_stop_norma2 = abs(Hstop_norma2); %eroarea din banda de stopare
Err_banda_trecere_norma2 = abs(1-Htrecere_norma2); %eroarea din banda de trecere
Dispersia_CMMP = std(abs(Hstop_norma2))/std(abs(1-Htrecere_norma2))

```

Graficele obtinute sunt :



Figură 7: Raspunsurile in frecventa a doua filtre de tip Filtru Trece-Jos de ponderi diferite construite in sens CMMP



Figură 7: Raspunsurile in frecventa a doua filtre de tip Filtru Trece-Jos de ponderi diferite construite in sens normei infinit

```
Dispersia_PMC =
    1.0798e-04

Dispersia_CMMP =
    0.2598

>> fazal_d

Dispersia_PMC =
    1.0798e-04

Dispersia_CMMP =
    0.2598
```

Figură 8: Raporturile obtinute pentru filtrul PMC si pentru filtrul CMMP

Raportul obținut este: 1.0798e-04 pentru filtrul în sens PMC și 0.2598 pentru filtrul în sens CMMP, așa cum se poate observa și în figura de mai sus. Acesta este destul de mare deoarece ordinul ales este relativ mic, iar filtrul este departe de a fi ideal.

## Subpunctul e)

Rezolvare: La acest subpunct a trebuit să proiectăm un FTS și un FTB prin 2 metode diferite. Am proiectat filtrele atât în sens CMMP, cât și în sensul normei infinite, astfel că pentru prima metodă atât la FTS, cât și la FTB am modificat parametrii funcției **firls**, respectiv **firpm**, mai exact **W** și **A** pentru a proiecta filtrele cerute. Pentru a doua metodă am folosit un FTJ realizat întâi cu **firls**, după care cu **firpm** exact ca la subpunctele precedente.

Pentru prima metodă procedăm ca la subpunctele precedente însă avem câteva modificări:

Pentru FTS:  $\omega_p = 0.3920$ ;  $\omega_s = 0.5316$ ;  
astfel că  $W = [0 \ \omega_p \ \omega_s]$ , iar  $A = [0 \ 0 \ 1 \ 1]$ .

Pentru FTB: acesta are câte 2 pulsații de trecere, respectiv 2 de stop, câte una la stânga și câte una la dreapta.

Le-am definit pe fiecare astfel:

$\omega_p = 0.3920$ ;

$\omega_s = 0.5316$ ;

$\pi - \omega_p = 0.8752$ ;

$\pi - \omega_s = 0.8307$ ;

iar  $W = [0 \ \omega_p \ \omega_s \ \pi - \omega_s \ \pi - \omega_p \ 1]$ ;

$A = [0 \ 0 \ 1 \ 1 \ 0 \ 0]$ .

Calcularea secvenței pondere, a răspunsului în frecvență, respectiv a fazei este identică cu subpunctele precedente, **firls/firpm**, **freqz** și **phasez**. Apoi folosim **plot** pentru diferențierea graficelor.

Pentru cea de a doua metodă am folosit un FTJ(secvența pondere) realizat cu **firls/firpm**. Am utilizat câte o buclă **for** pentru calcularea secvenței pondere pentru fiecare filtrul în parte astfel:

Pentru FTS: știind că secvența pondere a FTS este egală cu impulsul unitar – secvența pondere a FTJ realizat și impulsul unitar este 1 doar 0 și 0 în rest, am realizat o secvență **for** unde calculăm secvența pondere ca fiind 0 – secvența pondere a FTJ în toate punctele (până în  $M+1$ ), fără  $h[0]$  unde impulsul este 1, deci secvența pondere a FTS este 1 – secvența pondere a FTJ. Acest  $h[0]$  se află la  $M/2+1$ , unde  $M$  este ordinul FTJ.

Pentru FTB: calculăm 2 FTJ, unul cu  $W = [0 \ \pi - \omega_s \ \pi - \omega_p \ 1]$  și unul cu  $W = [0 \ \omega_p \ \omega_s \ 1]$ , astfel că un filtru va avea banda de trecere în jos unde o are și FTB, iar unul va avea banda de trecere în jos unde FTB are banda de trecere în sus. Secvența pondere se va calcula pe toată lungimea intervalului  $M+1$  ca fiind  $FTJ_r - FTJ_l$ , astfel obținem un FTB.

Codul folosit este:

```
%subpunctul e
%metoda 1 - firls modificat

%FTS

omega_p = 0.3920;
omega_s = 0.5316;

M = 20; %ordinul filtrului
W = [0 omega_p omega_s 1]; %vector ce contine pulsatiile normalizate
    reprezentand benzile de interes
A = [0 0 1 1]; %vector ce descrie valorile raspunsului dorit in benzile de
    interes

h_FTS1_norma2 = firls(M,W,A); %functia pondere a FTS

%raspunsul in frecventa(metoda 1); rezolutia 110 linii spectrale

[H_FTS1_norma2,omega_FTS1_norma2] = freqz(h_FTS1_norma2,1,110);
%FTB

pi_minus_omega_p = 0.8752;
pi_minus_omega_s = 0.8307;

M = 20; %ordinul filtrului
W = [0 omega_p omega_s pi_minus_omega_s pi_minus_omega_p 1];
A = [0 0 1 1 0 0];

h_FTB1_norma2 = firls(M,W,A); %functia pondere a FTB
%raspunsul in frecventa(metoda 1); rezolutia 110 linii spectrale

[H_FTB1_norma2,omega_FTB1_norma2] = freqz(h_FTB1_norma2,1,110);
%metoda 2 - folosind un FTJ
%FTS
%h_FTS = imp_unitar - h_FTJ

W = [0 omega_p omega_s 1];
A = [1 1 0 0];

h_FTJ = firls(M,W,A); %functia pondere a FTJ

%FTS = 1-FTJ

for i = 1 : (M+1)

    h_FTS2_norma2(i) = 0 - h_FTJ(i); %functia pondere a FTS
```

```

    if(i == M/2+1) %mijlocul corespunzator, unde impulsul e 1(punctul 0); h are
    un singur maxim deoarece M este par
        h_FTS2_norma2(i) = 1 - h_FTJ(i); %functia pondere a FTS
    end

end

%FTB = FTJ_right - FTJ_left

W = [0 omega_p omega_s 1];
A = [0 0 1 1];

h_FTJ_r = firls(M,W,A); %functia pondere a FTJ mai departat de 0

W = [0 pi_minus_omega_s pi_minus_omega_p 1];

h_FTJ_l = firls(M,W,A); %functia pondere a FTJ mai apropiat de 0

for i = 1 : (M+1)

    h_FTB2_norma2(i) = h_FTJ_r(i) - h_FTJ_l(i);

end

%raspunsul in frecventa pentru FTS(metoda 2) cu rezolutia 110 linii spectrale
[H_FTS2_norma2,omega_FTS2_norma2] = freqz(h_FTS2_norma2,1,110);
%raspunsul in frecventa pentru FTB(metoda 2) cu rezolutia 110 linii spectrale
[H_FTB2_norma2,omega_FTB2_norma2] = freqz(h_FTB2_norma2,1,110);

%Calculam erorile care exista intre cele 2 metode:

Err_h_FTS_norma2 = norm(h_FTS1_norma2 - h_FTS2_norma2)
Err_H_FTS_norma2 = norm(H_FTS1_norma2 - H_FTS2_norma2)

Err_h_FTB_norma2 = norm(h_FTB1_norma2 - h_FTB2_norma2)
Err_H_FTB_norma2 = norm(H_FTB1_norma2 - H_FTB2_norma2)

%metoda 1 - firpm modificat

%FTS
%benzile la FTS sunt inversate

omega_p = 0.3920;
omega_s = 0.5316;

%definirea parametrilor functiei firls, respectiv firpm
M = 20; %ordinul filtrului
W = [0 omega_p omega_s 1]; %vector ce contine pulsatiile normalizate
reprezentand benzile de interes
A = [0 0 1 1]; %vector ce descrie valorile raspunsului dorit in benzile de
interes

h_FTS1_inf = firpm(M,W,A); %functia pondere a FTS
[H_FTS1_inf,omega_FTS1_inf] = freqz(h_FTS1_inf,1,110); %raspunsul in frecventa
cu rezolutia 110 linii spectrale

```



%FTB

```
omega_p = 0.3920;  
omega_s = 0.5316;
```

```
pi_minus_omega_p = 0.8752;  
pi_minus_omega_s = 0.8307;
```

%definirea parametrilor functiei firls, respectiv firpm

```
M = 20; %ordinul filtrului
```

```
W = [0 omega_p omega_s pi_minus_omega_s pi_minus_omega_p 1];
```

```
A = [0 0 1 1 0 0];
```

```
h_FTBl_inf = firpm(M,W,A); %functia pondere a FTB
```

```
[H_FTBl_inf,omega_FTBl_inf] = freqz(h_FTBl_inf,1,110); %raspunsul in frecventa  
cu rezolutia 110 linii spectrale
```

%metoda 2 - folosind un FTJ

%FTS

```
%h_FTS = imp_unitar - h_FTJ
```

%benzile de trecere pentru FTJ

```
W = [0 omega_p omega_s 1];  
A = [1 1 0 0];
```

```
h_FTJ_inf = firpm(M,W,A); %functia pondere a FTJ
```

```
%FTS = 1-FTJ
```

```
for i = 1 : (M+1)
```

```
    h_FTS2_inf(i) = 0 - h_FTJ_inf(i); %functia pondere a FTS
```

```
    if(i == M/2+1) %mijlocul corespunzator, unde impulsul e 1(punctul 0); h are  
un singur maxim deoarece M este par
```

```
        h_FTS2_inf(i) = 1 - h_FTJ_inf(i); %functia pondere a FTS
```

```
    end
```

```
end
```

```
%FTB = FTJ_right - FTJ_left
```

```
%FTB se obtine scazand un FTJ ce are omega_p si omega_s mai departe de 0 si un  
FTJ ce are wp si ws mai aproape de 0
```

```
pi_minus_omega_p = 0.8752;  
pi_minus_omega_s = 0.8307;
```

```
W = [0 pi_minus_omega_s pi_minus_omega_p 1]; %vector ce contine pulsatiile  
normalizate reprezentand benzile de interes (coordonata x)
```

```
A = [1 1 0 0]; %vector ce descrie valorile raspunsului dorit in benzile de  
interes (coordonata y)
```

```
h_FTJ_r_inf = firpm(M,W,A); %functia pondere a FTJ mai departat de 0
```

```
W = [0 omega_p omega_s 1];
```

```
h_FTJ_l_inf = firpm(M,W,A); %functia pondere a FTJ mai apropiat de 0
```

```
for i = 1 : (M+1)
```

```
    h_FTB2_inf(i) = h_FTJ_r_inf(i) - h_FTJ_l_inf(i);
```

```
end
```

```
%raspunsul in frecventa pentru FTS cu rezolutia 110 linii spectrale
```

```
[H_FTS2_inf,omega_FTS2_inf] = freqz(h_FTS2_inf,1,110);
```

```
%raspunsul in frecventa pentru FTB cu rezolutia 110 linii spectrale
```

```
[H_FTB2_inf,omega_FTB2_inf] = freqz(h_FTB2_inf,1,110);
```

```
%Calculam erorile care exista intre cele 2 metode:
```

```
Err_h_FTS_normainf = norm(h_FTS1_inf - h_FTS2_inf)
```

```
Err_H_FTS_normainf = norm(H_FTS1_inf - H_FTS2_inf)
```

```
Err_h_FTB_normainf = norm(h_FTB1_inf - h_FTB2_inf)
```

```
Err_H_FTB_normainf = norm(H_FTB1_inf - H_FTB2_inf)
```

```
%trasarea graficelor
```

```
%Norma Euclidiană - CMMP
```

```
figure('Name','Faza 1 - E - FTS');
```

```
subplot(2,1,1);
```

```
plot(omega_FTS1_norma2,20*log(abs(H_FTS1_norma2))); %trasarea caracteristicii de  
frecventa (metoda 1)
```

```
hold on;
```

```
plot(omega_FTS1_inf,20*log(abs(H_FTS1_inf))); %trasarea caracteristicii de  
frecventa (metoda 1)
```

```
hold on;
```

```
plot(omega_FTS2_norma2,20*log(abs(H_FTS2_norma2))); %trasarea caracteristicii de  
frecventa (metoda 2)
```

```
hold on;
```

```
plot(omega_FTS2_inf,20*log(abs(H_FTS2_inf))); %trasarea caracteristicii de  
frecventa (metoda 2)
```

```
xlabel('Vectorul frecventelor');
```

```
ylabel('Amplitudinea (dB)');
```

```
title('Caracteristica de frecventa a FTS');
```

```
legend('CMMP-metoda 1','Alg. PMC-metoda 1','CMMP-metoda 2','Alg. PMC-metoda 2');
```

```
hold off;
```

```
subplot(2,1,2);
```

```
[faza_FTS1_norma2, omega_FTS1_norma2] = phasez(h_FTS1_norma2,1,110); %faza  
filtrului cu rezolutia 110 de linii spectrale
```

```
plot(omega_FTS1_norma2, faza_FTS1_norma2); %trasarea graficului fazei
```

```
hold on;
```

```
[faza_FTS1_inf, omega_FTS1_inf] = phasez(h_FTS1_inf,1,110); %faza filtrului cu  
rezolutia 110 de linii spectrale
```

```
plot(omega_FTS1_inf, faza_FTS1_inf); %trasarea graficului fazei
```

```
hold on;
```

```
[faza_FTS2_norma2, omega_FTS2_norma2] = phasez(h_FTS2_norma2,1,110); %faza  
filtrului cu rezolutia 110 de linii spectrale
```

```
plot(omega_FTS2_norma2, faza_FTS2_norma2); %trasarea graficului fazei
```

```

hold on;
[faza_FTS2_inf, omega_FTS2_inf] = phasez(h_FTS2_inf,1,110); %faza filtrului cu
rezolutia 110 de linii spectrale
plot(omega_FTS2_inf, faza_FTS2_inf); %trasarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza FTS');
legend('CMMP-metoda 1','Alg. PMC-metoda 1','CMMP-metoda 2','Alg. PMC-metoda 2');
hold on;

saveas(gcf,'Faza1-E-FTS.png');

figure('Name','Faza 1 - E - FTB');

subplot(2,1,1);
plot(omega_FTB1_norma2,20*log(abs(H_FTB1_norma2))); %afisarea caracteristicii de
frecventa
hold on;
plot(omega_FTB1_inf,20*log(abs(H_FTB1_inf))); %afisarea caracteristicii de
frecventa
plot(omega_FTB2_norma2,20*log(abs(H_FTB2_norma2))); %afisarea caracteristicii de
frecventa
hold on;
plot(omega_FTB2_inf,20*log(abs(H_FTB2_inf))); %afisarea caracteristicii de
frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTB');
legend('CMMP-metoda 1','Alg. PMC-metoda 1','CMMP-metoda 2','Alg. PMC-metoda 2');
hold off;

subplot(2,1,2);
[faza_FTB1_norma2, omega_FTB1_norma2] = phasez(h_FTB1_norma2,1,110); %faza
filtrului cu rezolutia 110 de linii spectrale
plot(omega_FTB1_norma2, faza_FTB1_norma2); %trasarea graficului fazei
hold on;

[faza_FTB1_inf, omega_FTB1_inf] = phasez(h_FTB1_inf,1,110); %faza filtrului cu
rezolutia 110 de linii spectrale
plot(omega_FTB1_inf, faza_FTB1_inf); %trasarea graficului fazei
hold on;

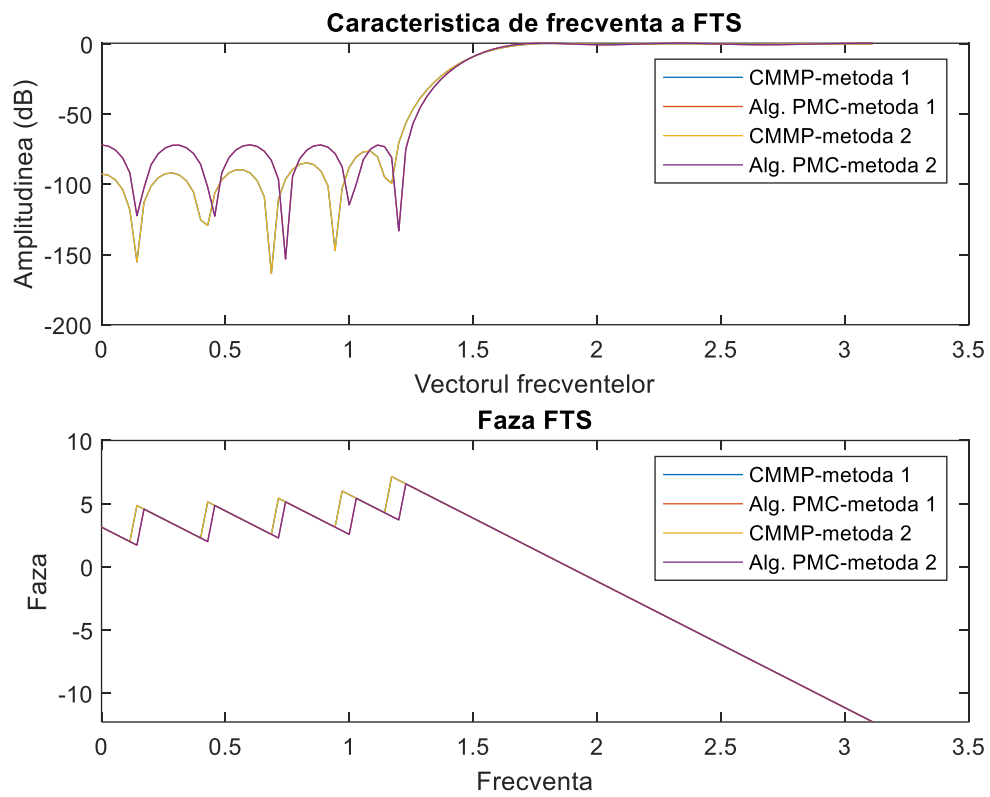
[faza_FTB2_norma2, omega_FTB2_norma2] = phasez(h_FTB2_norma2,1,110); %faza
filtrului cu rezolutia 110 de linii spectrale
plot(omega_FTB2_norma2, faza_FTB2_norma2); %trasarea graficului fazei
hold on;

[faza_FTB2_inf, omega_FTB2_inf] = phasez(h_FTB2_inf,1,110); %faza filtrului cu
rezolutia 110 de linii spectrale
plot(omega_FTB2_inf, faza_FTB2_inf); %trasarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza FTB');
legend('CMMP-metoda 1','Alg. PMC-metoda 1','CMMP-metoda 2','Alg. PMC-metoda 2');
hold on;

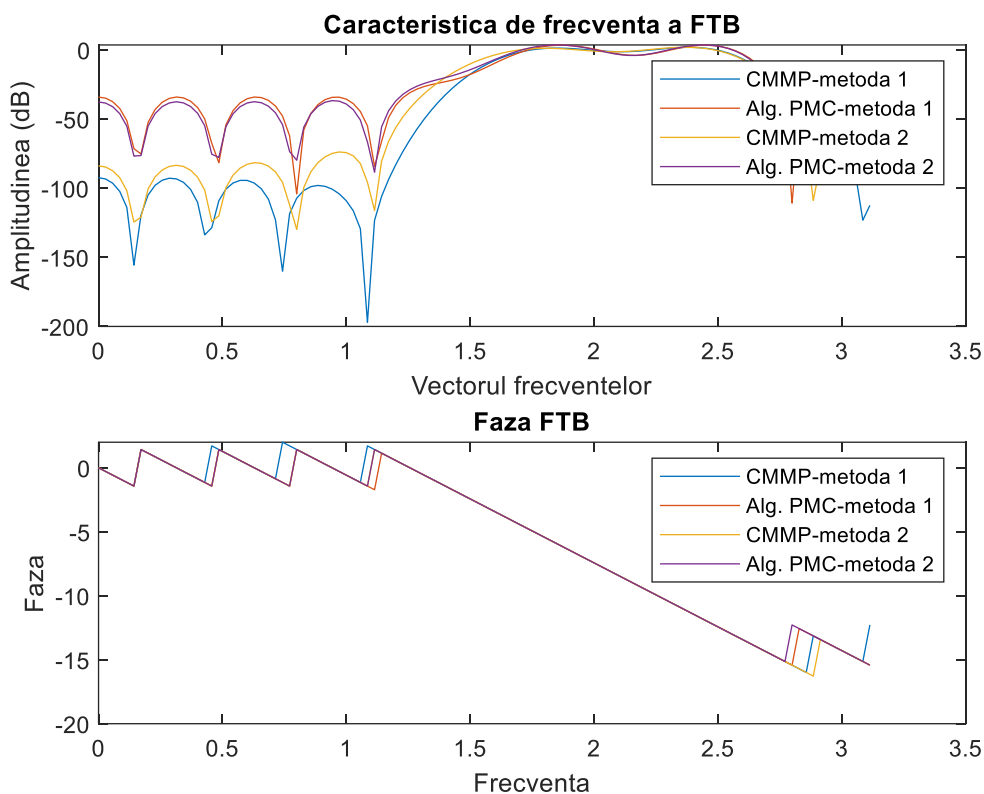
saveas(gcf,'Faza1-E-FTB.png');

```

Graficele obținute sunt:



Figură 9: Raspunsurile in frecventa a patru filtre de tip Filtru Trece -Sus construite in sens CMMP si in sensul  
normei



Figură 10: Răspunsurile în frecvență a patru filtre de tip Filtru Trece-Bandă construite în sens CMMP și în sensul normei infinit

Err_h_FTS_norma2 =	Err_h_FTS_normainf =
5.1442e-16	7.2034e-16
Err_H_FTS_norma2 =	Err_H_FTS_normainf =
5.4963e-15	7.6836e-15
Err_h_FTB_norma2 =	Err_h_FTB_normainf =
0.0598	0.0332
Err_H_FTB_norma2 =	Err_H_FTB_normainf =
0.6271	0.3471

Figură 11: Erorile pentru cele două tipuri de filtre

Erorile obținute la FTS sunt foarte mici, de aprox  $7 \cdot 10^{-16}$ . La FTB se poate observa și pe grafic faptul că avem niște erori destul de mari, cu o diferență de aprox 0.03 și 0.3 între cele două metode în benzile de stopare.

## Faza 2

Proiectarea în sens CMMP a filtrelor FIR cu răspuns ideal complex, fără restricții de fază.

### Subpunctul a)

Am folosit funcția **firls\_FTJ\_c.m**, încărcată pe platforma Moodle, pentru a proiecta diferite filtre fără restricții de fază. Pulsațiile de trecere și de stopare le-am păstrat pe aceleași de la FAZA 1:

$\omega_p = 0.3920$ ;

$\omega_s = 0.5316$ ;

Funcția dată se apelează astfel:

$h = \text{firls\_FTJ\_c}(M, \omega_p, \omega_s, K)$

unde **M** este ordinul, iar **K** este întârzierea de grup.

Ordinul dat este  $M = 20$ , iar întârzierea de grup între  $M/4=5$  și  $3*M/4=15$ . Am trasat graficele secvenței pondere, răspunsului în frecvență, dar și al deviației de la liniaritate atât pentru valorile **K**.

Codul utilizat este:

%subpunctul a

M = 20; %ordinul filtrului

omega\_p = 0.3920;

omega\_s = 0.5316;

K = [5 7 10 13 15];

for i = 1:length(K)

h = firls\_FTJ\_c(M, omega\_p, omega\_s, i); %functia pondere a FTJ;

%secventa pondere

subplot(3,1,1);

stem(h); %trasarea secvenței pondere a FTJ

xlabel('Timp');

ylabel('Amplitudinea liniara');

title('Secventa pondere');

hold on;

legend('K=5', 'K=7', 'K=10', 'K=13', 'K=15'); %legenda

%raspunsul in frecventa

subplot(3,1,2);

[H, omega] = freqz(h, 1, 120); %raspunsul in frecventa cu rezolutia 120 de linii spectrale

plot(omega, 20\*log(abs(H))); %trasarea caracteristicii de frecventa

xlabel('Vectorul frecventelor');

ylabel('Amplitudinea (dB)');

title('Caracteristica de frecventa');

hold on;

legend('K=5', 'K=7', 'K=10', 'K=13', 'K=15');

```
%faza
subplot(3,1,3);
[faza, omega] = phasez(h,1,120); %faza filtrului cu rezolutia 120 de linii
spectrale
plot(omega, faza); %trasarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza'); %titlul figurii
legend('K=5','K=7','K=10','K=13','K=15');
hold on;

end
saveas(gcf, 'Faza2-A1.png');

figure();
for i = 1:1:length(K)

    h = firls_FTJ_c(M,omega_p,omega_s,i); %functia pondere a FTJ;

    %deviatia de la liniaritate
    grpdelay(h,1000);
    title('Deviatia de la liniaritate');
    hold on;
    legend('K=5','K=7','K=10','K=13','K=15');

end
saveas(gcf, 'Faza2-A2.png');

figure();
%intarzierea = M/2
h = firls_FTJ_c(M,omega_p,omega_s,M/2);

%secventa pondere
subplot(4,1,1);
stem(h); %trasarea secventei pondere a FTJ
xlabel('Timp');
ylabel('Amplitudinea liniara');
title('Secventa pondere');
legend('K=10');

%raspunsul in frecventa
subplot(4,1,2);
[H,omega] = freqz(h,1,120); %raspunsul in frecventa cu rezolutia 120 de
linii spectrale
plot(omega,20*log(abs(H))); %trasarea caracteristicii de frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa ');
hold on;
legend('K=10'); %legenda

%faza
subplot(4,1,3);
[faza, omega] = phasez(h,1,120); %faza filtrului cu rezolutia 120 de linii
spectrale
plot(omega, faza); %trasarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza');
legend('K=10');
```

```
subplot(4,1,4)
grpdelay(h,1000);
title('Deviatia de la liniaritate'); %titlul figurii
hold on;
legend('K=10');
saveas(gcf, 'Faza2-A-K=10.png');
```

Graficele obținute sunt:

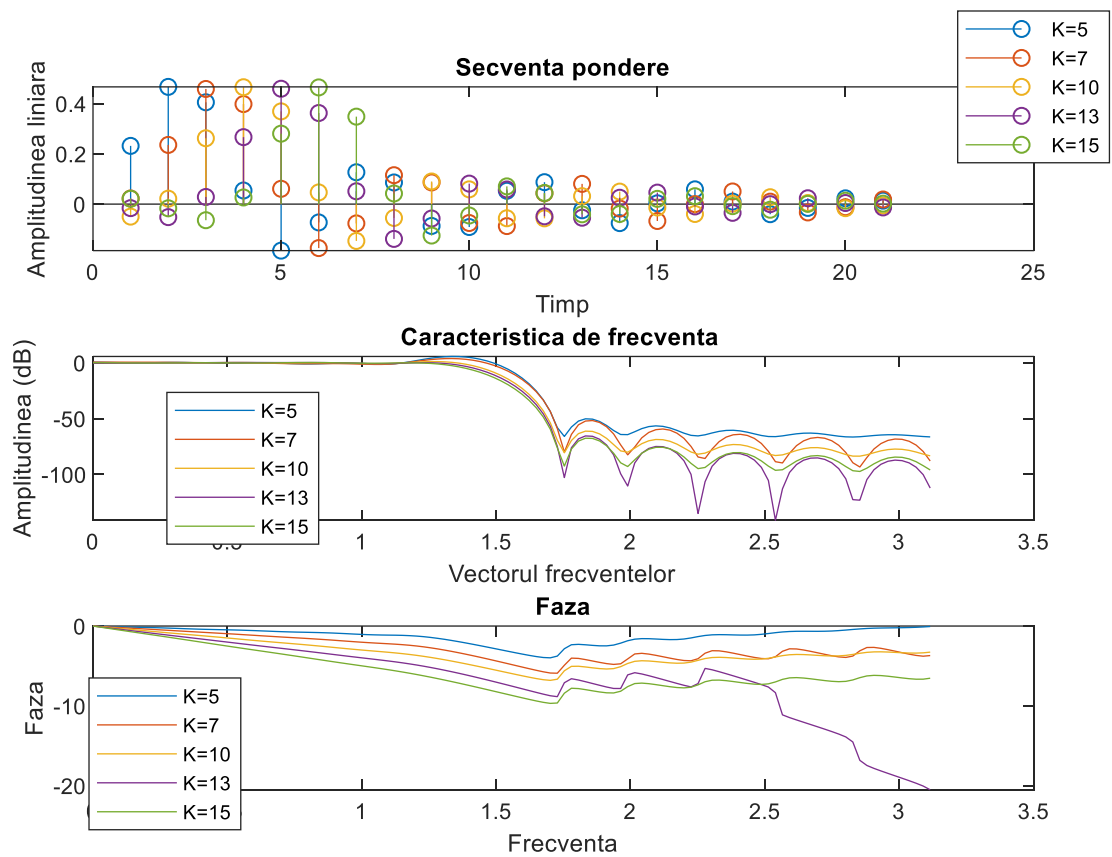
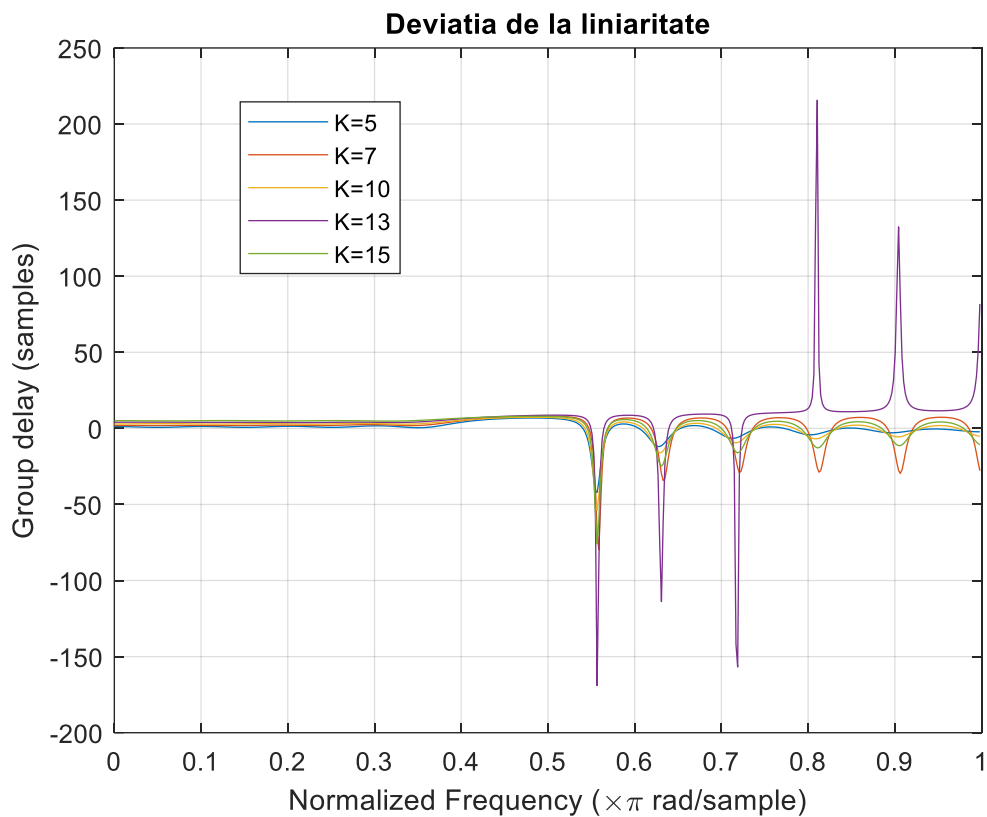
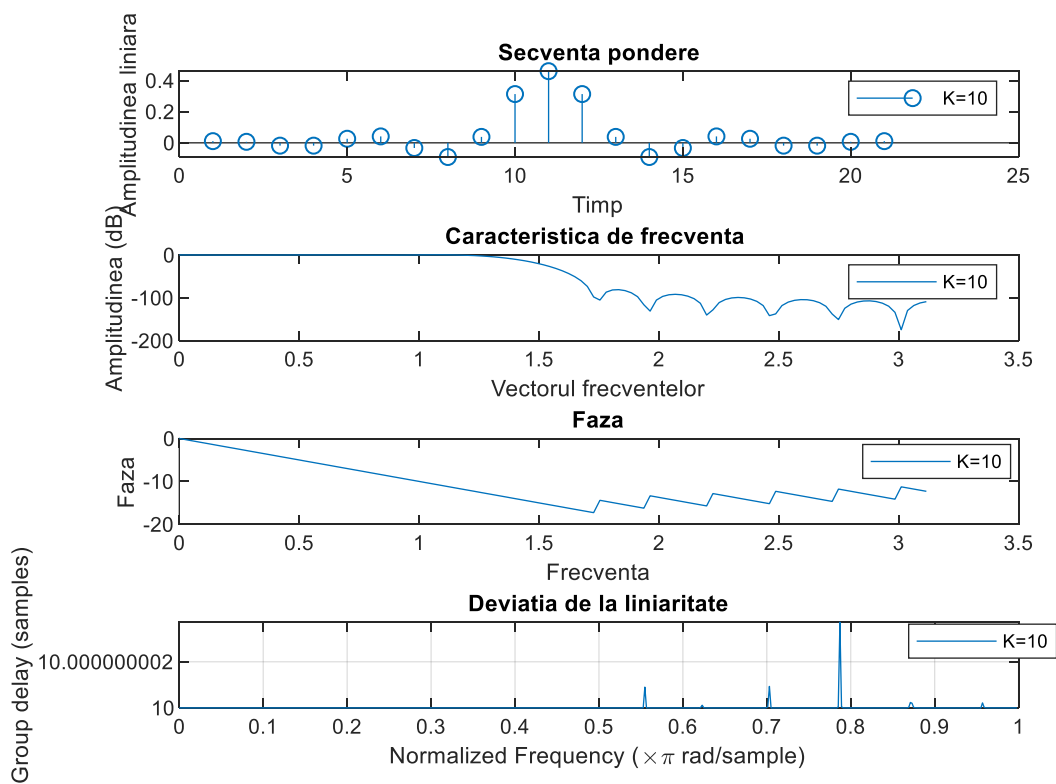


Figura 11: Secvențele pondere și răspunsurile în frecvență a mai multor filtre de tip Filtru Trece-Jos, cu întârzieri de grup diferite





Figură 12: Deviația de la liniaritate a fiecărui filtru de tip Filtru Trece-Jos, cu întârzieri de grup diferite



Figură 13: Secvența pondere, răspunsul în frecvență și deviația de la liniaritate a unui filtru de tip Filtru Trece-Jos cu întârzierea de grup fiind jumătate din ordin

De asemenea, odată cu creșterea întârzierii, observăm că punctul de maxim al secvenței de pondere este la un moment de timp mai îndepărtat.

Am realizat și graficele deviației de la liniaritate unde am observat ca odată cu creșterea frecvenței, întârzierea este foarte mare în doar câteva puncte.

### Subpunctul b)

Pentru rezolvarea acestui subpunct am modificat funcția dată **firls\_FTJ\_c.m**, adăugând un nou parametru de ieșire pentru a vedea performanța relativă a fiecărui filtru proiectat. Modificarea s-a realizat astfel:

Funcția are un nou parametru de ieșire, denumit **pr**. Acest parametru se calculează astfel:

$$pr = 100 \cdot (\omega_p - r^T h) / \omega_p.$$

Am păstrat aceleași pulsații de trecere, respectiv de stop și am trasat secvența pondere și caracteristica de frecvență pentru filtre de ordin diferit, de la  $M = 10$ , până la  $M = 100$ , cu pasul 20.

Codul folosit este:

```
%subpunctul b

%Parametrii utilizati pentru apelarea functiei firls_FTJ_c.m sunt:
omega_p = 0.3920;
omega_s = 0.5316;

%trasarea secventei pondere si a raspunsului in frecventa pentru valori diferite
ale intarzierii de grup si ale ordinului filtrului
for M = 10 : 20 : 100

    figure();
    [h pr] = firls_FTJ_c(M,omega_p,omega_s,M/2); %functia pondere a FTJ;

    %secventa pondere
    subplot(3,1,1);
    stem(h); %trasarea secventei pondere a FTJ
    xlabel('Timp');
    ylabel('Amplitudine');
    title(['Secventa pondere pentru M=',num2str(M), ' si K=',num2str(M/2)]);
    legend(['Performanta Relativa = ',num2str(pr)]); %performanta relativa

    %raspunsul in frecventa
    subplot(3,1,2);
    [H,omega] = freqz(h,1,130); %raspunsul in frecventa cu 130 de linii
spectrale
    plot(omega,20*log(abs(H))); %trasarea caracteristicii de frecventa
    xlabel('Vectorul frecventelor');
    ylabel('Amplitudinea (dB)');
    title(['Caracteristica de frecventa pentru M=',num2str(M), ' si
K=',num2str(M/2)]);

    %faza
    subplot(3,1,3);
    [faza, omega] = phasez(h,1,130); %faza filtrului cu rezolutia 130 de linii
spectrale
    plot(omega, faza); %trasarea graficului fazei
    xlabel('Frecventa');
    ylabel('Faza');
    title(['Faza pentru M=',num2str(M), ' si K=',num2str(M/2)]);
```

```
end

%Folosim perechea de pulsatii {pi-omega_s, pi-omega_p};

%%
pi_minus_omega_p = 0.8752;
pi_minus_omega_s = 0.8307;

for M = 10 : 20 : 100

    figure();
    [h pr] = firls_FTJ_c(M,pi_minus_omega_p,pi_minus_omega_s,M/2); %functia
    pondere a FTJ;

    %secventa pondere
    subplot(3,1,1);
    stem(h); %trasarea secventei pondere a FTJ
    xlabel('Timp');
    ylabel('Amplitudine');
    title(['Secventa pondere pentru M=',num2str(M), ' si K=',num2str(M/2)]);
    legend(['Performanta Relativa = ',num2str(pr)]; %performanta relativa

    %raspunsul in frecventa
    subplot(3,1,2);
    [H,omega] = freqz(h,1,130); %raspunsul in frecventa cu 130 de linii
spectrale
    plot(omega,20*log(abs(H))); %trasarea caracteristicii de frecventa
    xlabel('Vectorul frecventelor');
    ylabel('Amplitudinea (dB)');
    title(['Caracteristica de frecventa pentru M=',num2str(M), ' si
K=',num2str(M/2)]);

    %faza
    subplot(3,1,3);
    [faza, omega] = phasez(h,1,130); %faza filtrului cu rezolutia 130 de linii
spectrale
    plot(omega, faza); %trasarea graficului fazei
    xlabel('Frecventa');
    ylabel('Faza');
    title(['Faza pentru M=',num2str(M), ' si K=',num2str(M/2)]);

end
```

Graficele obținute sunt:

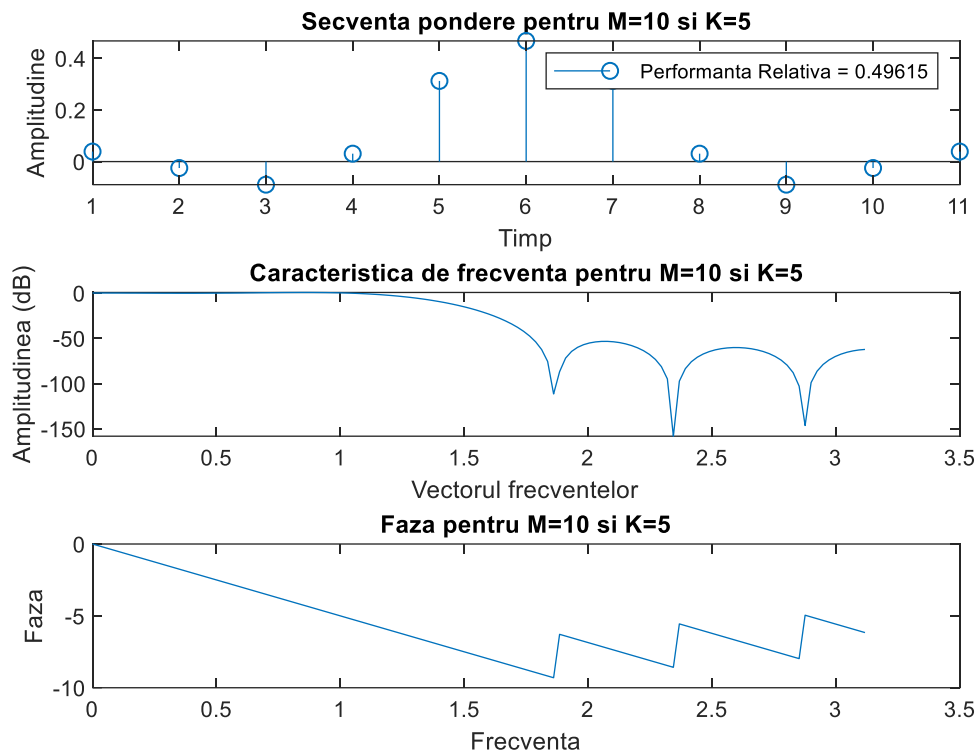


Figura 14: Secvența pondere și răspunsul în frecvență a unui filtru de tip Filtru Trece-Jos cu ordinul 10 și întârzierea de grup 5

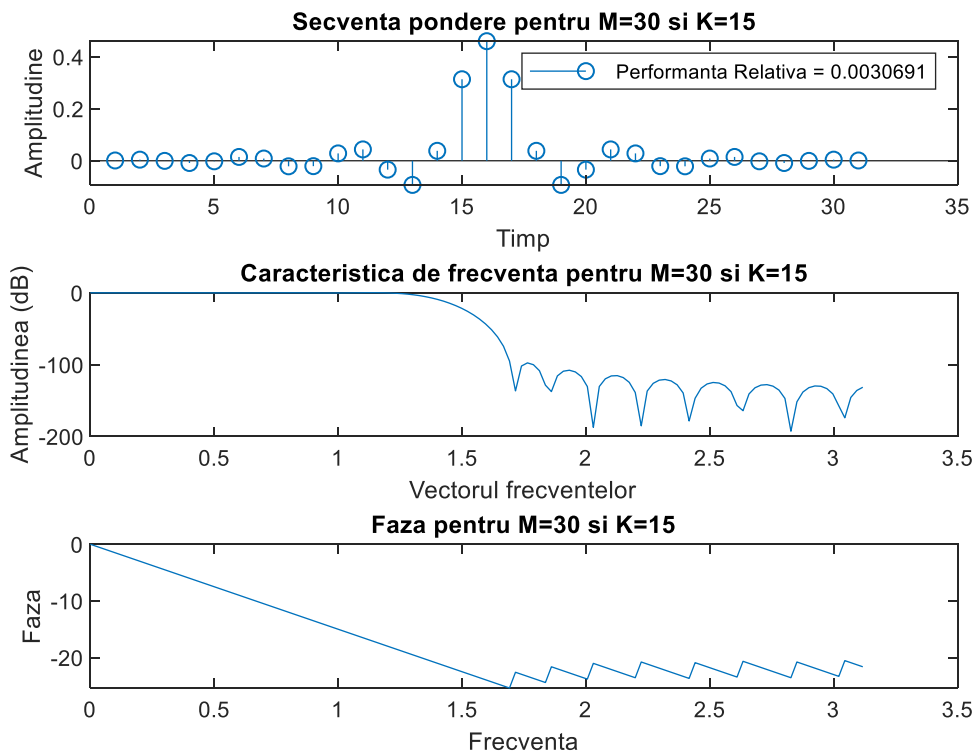


Figura 15: Secvența pondere și răspunsul în frecvență a unui filtru de tip Filtru Trece-Jos cu ordinul 30 și întârzierea de grup 15

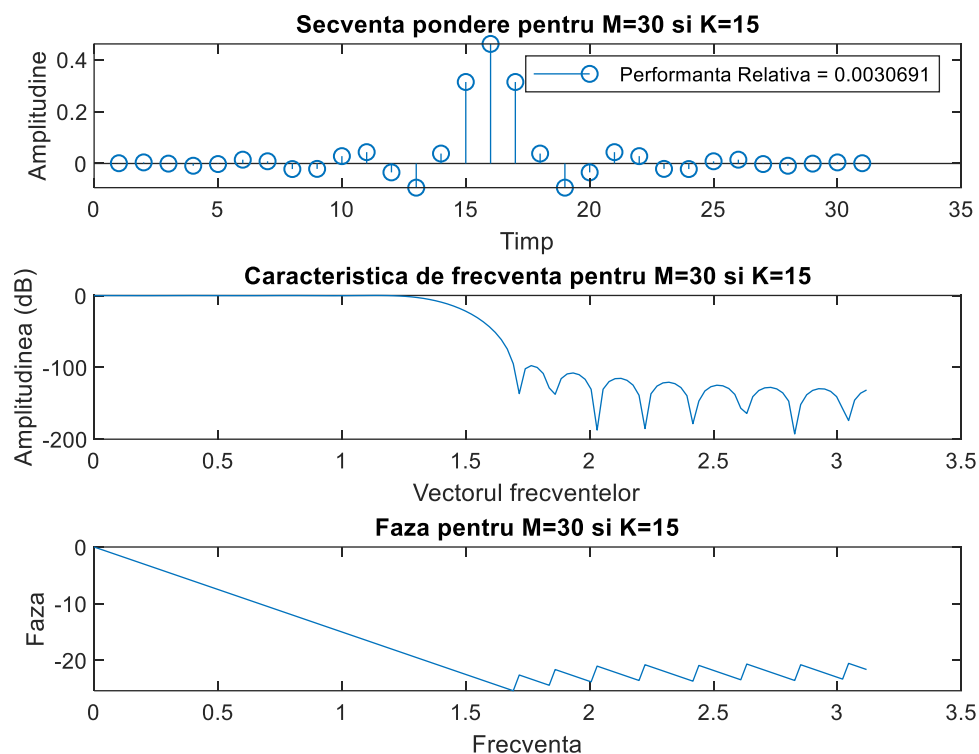


Figura 16: Secvența pondere și răspunsul în frecvență a unui filtru de tip Filtru Trece-Jos cu ordinul 30 și întârzierea de grup 15

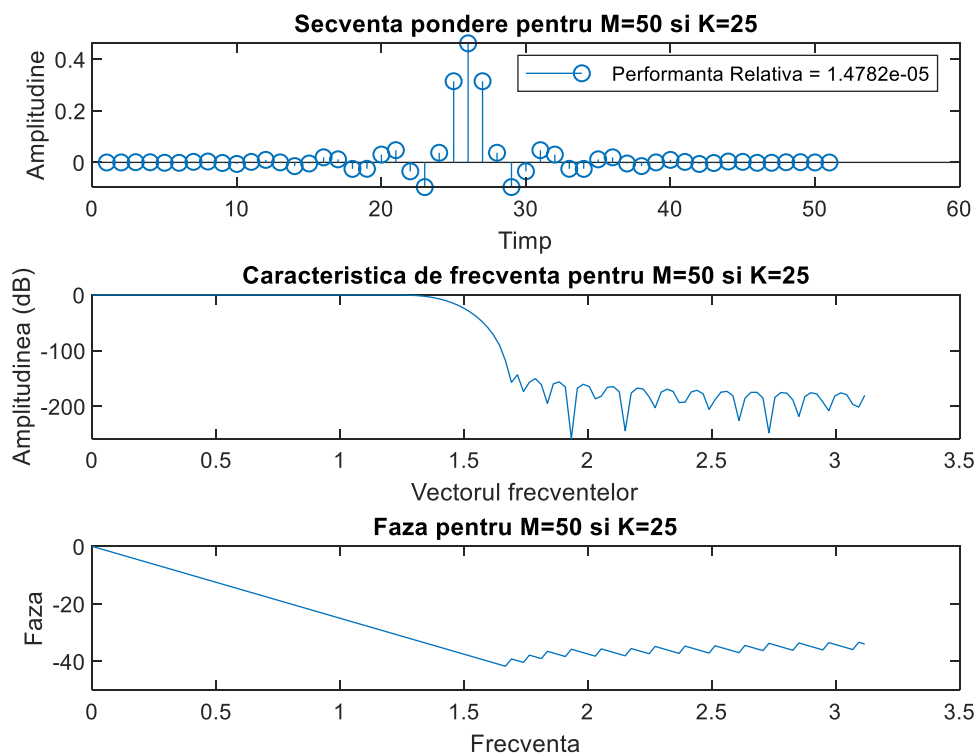


Figura 17: Secvența pondere și răspunsul în frecvență a unui filtru de tip Filtru Trece-Jos cu ordinul 50 și întârzierea de grup 25

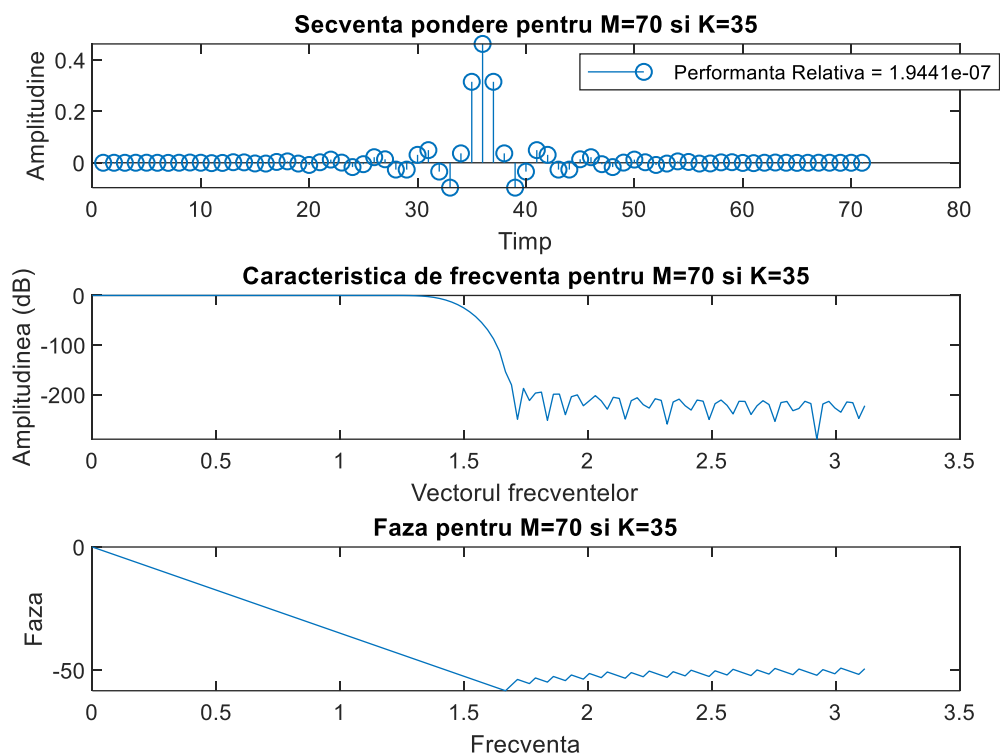


Figura 18: Secvența pondere și răspunsul în frecvență a unui filtru de tip Filtru Trece-Jos cu ordinul 70 și întârzierea de grup 35

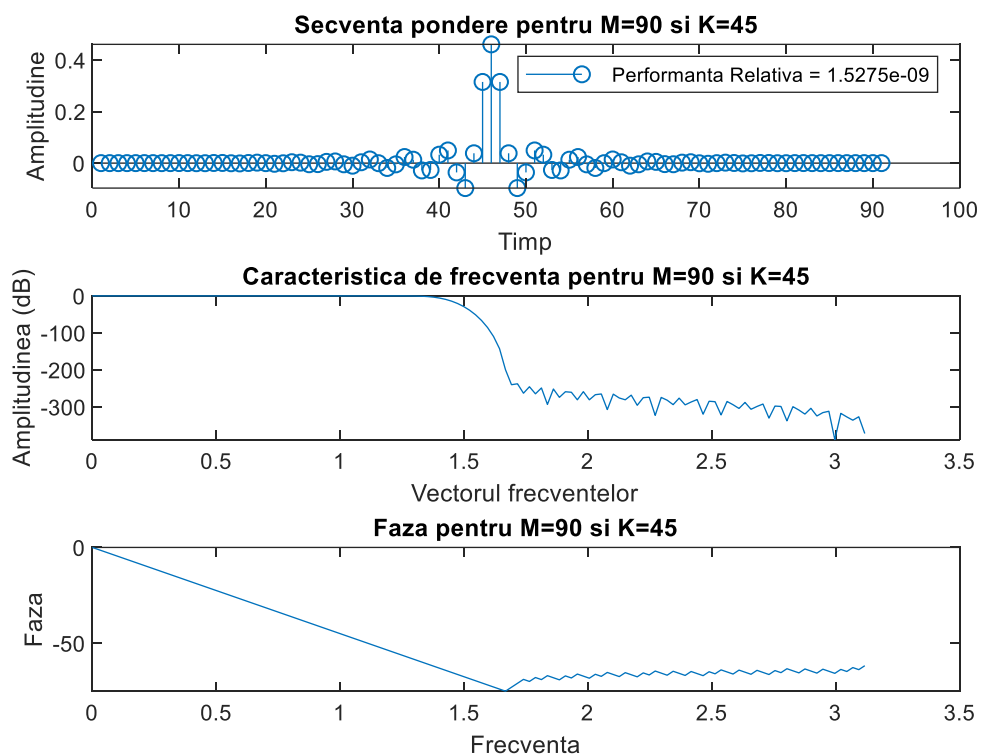


Figura 19: Secvența pondere și răspunsul în frecvență a unui filtru de tip Filtru Trece-Jos cu ordinul 90 și întârzierea de grup 45

Analizand graficele si performantele relative ale fiecarui filtru observam ca odata cu cresterea ordinului filtrului ,performanta relativa scade, altfel spus filtrele au o performanta mai ridicata.

## Subpunctul c)

Într-un script am realizat două filtre FTS și două filtre FTB, create folosind atât funcțiile noi cât și funcția prestabilită în Matlab **firls** cu parametri corespunzători, pentru a putea compara rezultatele.

Codul folosit este:

```
%subpunctul c

%FTS calculat cu functia firls_FTS_c.m

%Parametrii utilizati pentru apelarea functiei firls_FTS_c.m sunt:
M = 20;
omega_p = 0.3920;
omega_s = 0.5316;

[h_FTS1 pr_FTS]= firls_FTS_c(M,omega_s,omega_p,M/2); %functia pondere a FTS;

%raspunsul in frecventa
[H_FTS1,omega_FTS1] = freqz(h_FTS1,1,110); %raspunsul in frecventa cu 110 de
linii spectrale

figure();
subplot(2,1,1);
plot(omega_FTS1,20*log(abs(H_FTS1))); %trasarea caracteristicii de frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTS construit cu noua functie si M = 20');
legend(['Performanta Relativa = ',num2str(pr_FTS)]);

subplot(2,1,2);
[faza, omega] = phasez(h_FTS1,1,110); %faza filtrului cu rezolutia 110 de linii
spectrale
plot(omega, faza); %trasarea graficului fazei
xlabel('Frecventa'); %denumirea axelor
ylabel('Faza');
title('Faza FTS construit cu noua functie si M = 20');
saveas(gcf, 'Faza2-C-FTS1.png');

%FTS utilizand functia firls predefinita in Matlab

W = [0 omega_p omega_s 1];
A = [0 0 1 1];

h_FTS2 = firls(M,W,A); %functia pondere a FTS
```

```

[H_FTS2,omega_FTS2] = freqz(h_FTS2,1,110); %raspunsul in frecventa cu rezolutia
110 linii spectrale

figure();
subplot(2,1,1);
plot(omega_FTS2,20*log(abs(H_FTS2))); %trasarea caracteristicii de frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTS construit cu functia FIRLS si avand M =
20'); %titlul figurii

subplot(2,1,2);
[faza, omega] = phasez(h_FTS2,1,110); %faza filtrului; rezolutia 110 de linii
spectrale
plot(omega, faza); %trasarea graficului fazei
xlabel('Frecventa'); %denumirea axelor
ylabel('Faza');
title('Faza FTS construit cu functia FIRLS si M = 20'); %titlul figurii

saveas(gcf, 'Faza2-C-FTS2.png');
hold off;

%FTB utilizand firls_FTB_c.m

%Parametrii utilizati pentru apelarea functiei firls_FTB_c.m sunt:
M = 20;

omega_p = 0.3920;
omega_s = 0.5316;
pi_minus_omega_p = 0.8752;
pi_minus_omega_s = 0.8307;

figure();
[h_FTB1 pr_FTB]= firls_FTB_c(M,omega_s,omega_p,
pi_minus_omega_s,pi_minus_omega_p,M/2); %functia pondere a FTB;

%raspunsul in frecventa
[H_FTB1,omega_FTB1] = freqz(h_FTB1,1,110); %raspunsul in frecventa cu 110 de
linii spectrale

subplot(2,1,1);
plot(omega_FTB1,20*log(abs(H_FTB1))); %trasarea caracteristicii de frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTB construit cu noua functie si M = 20');
legend(['Performanta Relativa = ',num2str(pr_FTB)]); %afisarea performantei
relative

subplot(2,1,2);
[faza, omega] = phasez(h_FTB1,1,110); %faza filtrului cu rezolutia 110 de linii
spectrale
plot(omega, faza); %trasarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza FTB construit cu noua functie si M = 20');

saveas(gcf, 'Faza2-C-FTB1.png');

```



```
%FTB utilizand functia firls

W = [0 omega_p omega_s pi_minus_omega_s pi_minus_omega_p 1];
A = [0 0 1 1 0 0];

h_FTB2 = firls(M,W,A); %functia pondere a FTB
[H_FTB2,omega_FTB2] = freqz(h_FTB2,1,110);

figure();
subplot(2,1,1);
plot(omega_FTB2,20*log(abs(H_FTB2))); %afisarea caracteristicii de frecventa
xlabel('Vectorul frecventelor');
ylabel('Amplitudinea (dB)');
title('Caracteristica de frecventa a FTB construit cu functia FIRLS si avand M = 20');

subplot(2,1,2);
[faza, omega] = phasez(h_FTB2,1,110); %faza filtrului cu rezolutia 110 de linii spectrale
plot(omega, faza); %trasarea graficului fazei
xlabel('Frecventa');
ylabel('Faza');
title('Faza FTB construit cu functia FIRLS si M = 20');

saveas(gcf,'Faza2-C-FTB2.png');
hold off;
```

Graficele obținute sunt:

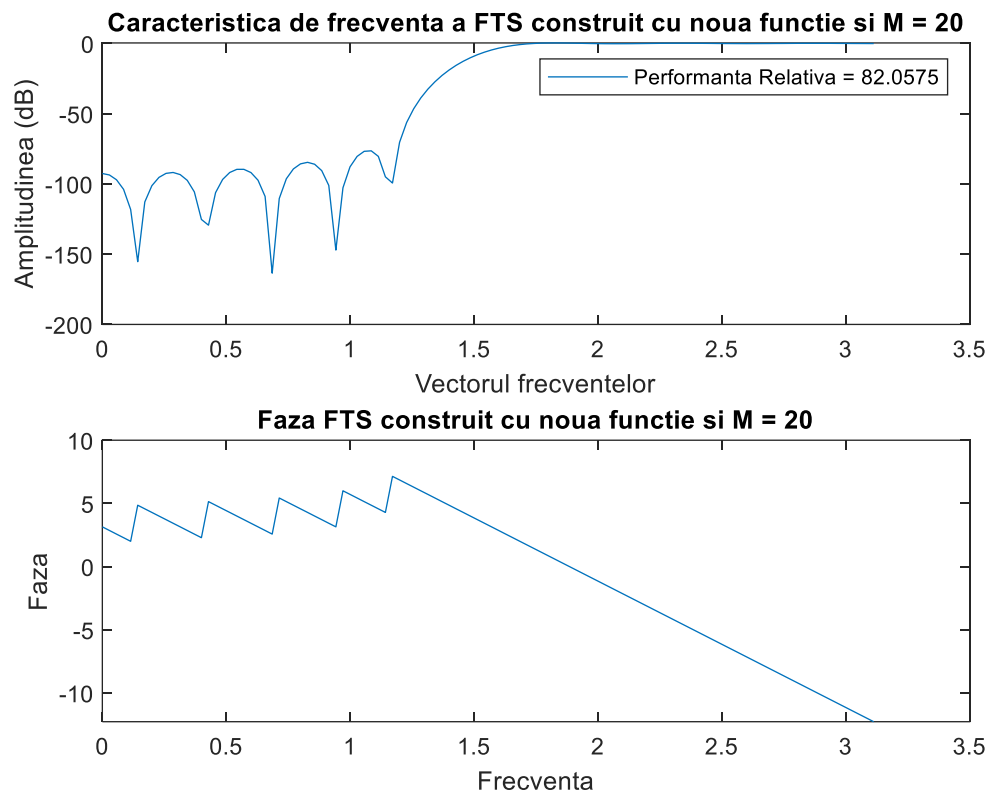


Figura 2: Răspunsul în frecvență a unui filtru de tip Filtru Trece-Sus, construit cu funcția `firls_FTS_c`

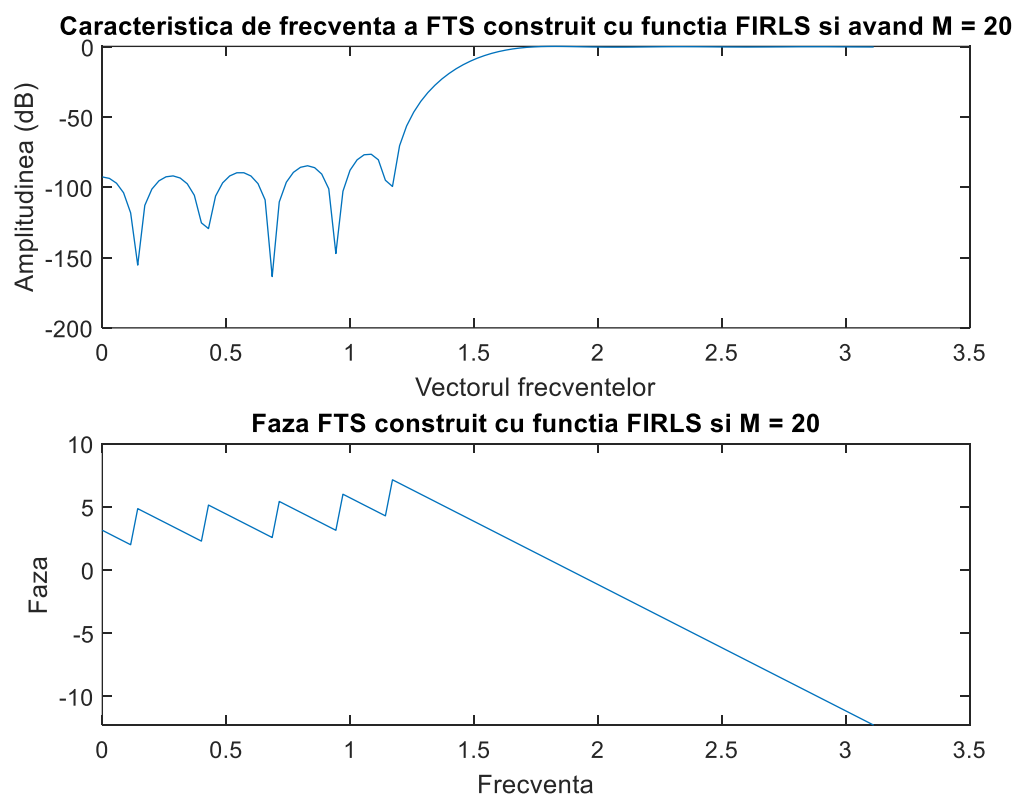


Figura 3: Răspunsul în frecvență a unui filtru de tip Filtru Trece-Sus, construit cu funcția `firls`

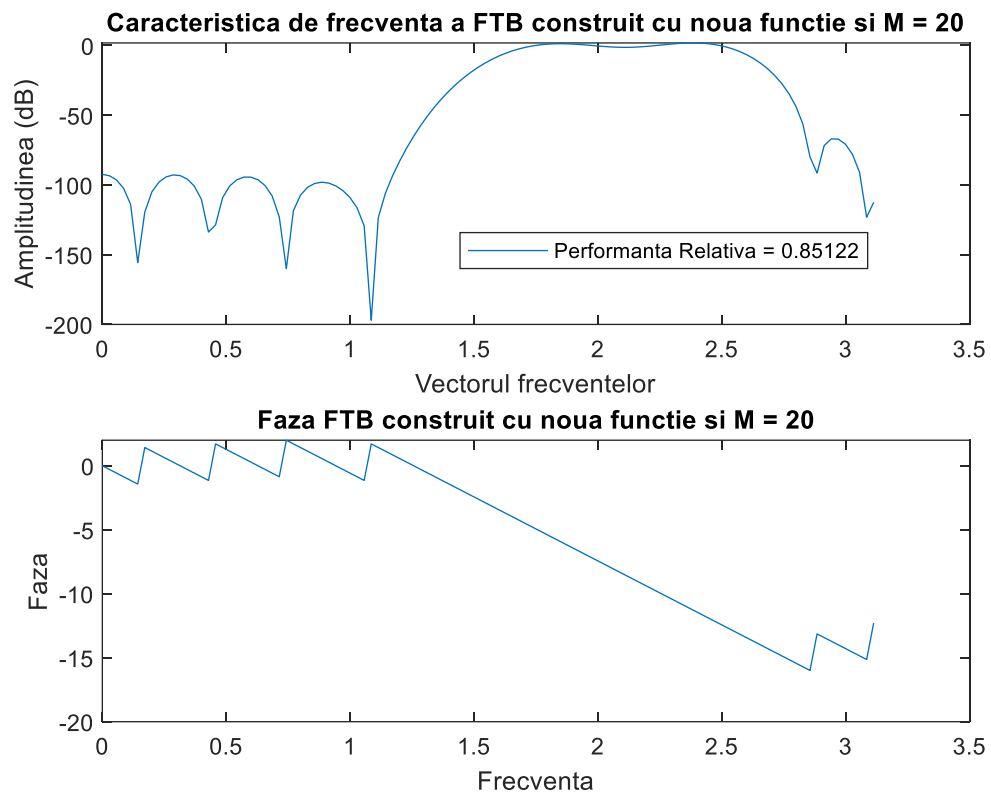


Figura 4: Răspunsul în frecvență a unui filtru de tip Filtru Trece-Sus, construit cu funcția `firls_FTB_c`

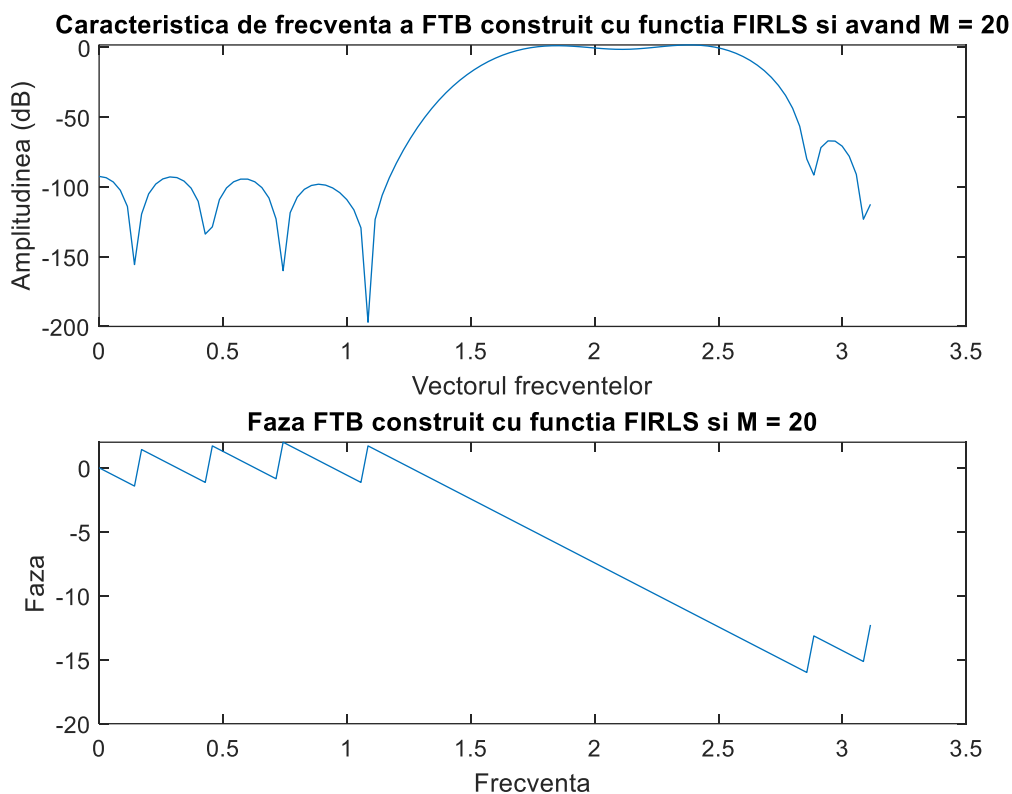


Figura 5: Răspunsul în frecvență a unui filtru de tip Filtru Trece-Sus, construit cu funcția firls

## Concluzii

1. Filtrele pot fi clasificate în funcție de performanțele relative(erori relative).
2. Creșterea ordinului duce filtrul mai aproape de un filtru ideal.
3. Ordinul nu poate fi oricât de mare. Mărirea excesivă a acestuia duce spre perturbarea matricei de inversat.
4. Știind doar un filtru de tip FTJ, putem obține filtre de tip FTS, respectiv FTB folosind formulele  $FTS = 1 - FTJ$  și  $FTB = FTJr - FTJl$ .
5. Dispersia este mai mare cu cât filtrul are un ordin mai mic.
6. Lobii din benzile de stopare/trecere au aceeași mărime pentru cazul normei infinit.

## Bibliografie

Student: Stanciuc Vlad

IS

Grupa: 334 AB

- Curs 11 Prelucrarea Semnalelor – Prof. Dan ȘTEFĂNOIU
- Suport teoretic laborator 5 - Prof. Dan ȘTEFĂNOIU si Conf. Alexandru DUMITRASCU
- Notițe laborator 5
- Matlab documentation