5Ministerul Educației și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

**RAPORT**

**Lucrarea de laborator nr.6**

**la Prelucrarea Semnalelor**

*Tema: Sisteme discrete liniare în timp cintinuu cercetate în domeniul de frecvență.*

**Grupa academică:**  TI-211

**A efectuat:**  Popa Cătălin

**A verificat:**  Potlog Mihail

Chișinău 2024

**Exercițiul 1**

În acest program se calculează aproximarea de sus. Se folosește funcția sinc de sistemul MATLAB.

%Programul P4\_1

% raspunsul impuls filtrului ideal

clf;

fc=0.25;

n=[-6.5:1:6.5];

y=2\*fc\*sinc(2\*fc\*n);

k=n+6.5;

stem(k,y); title('N=20'); axis([0 20 -0.2 0.6]);

16

xlabel('Timpul n'); ylabel('Amplituda'); grid

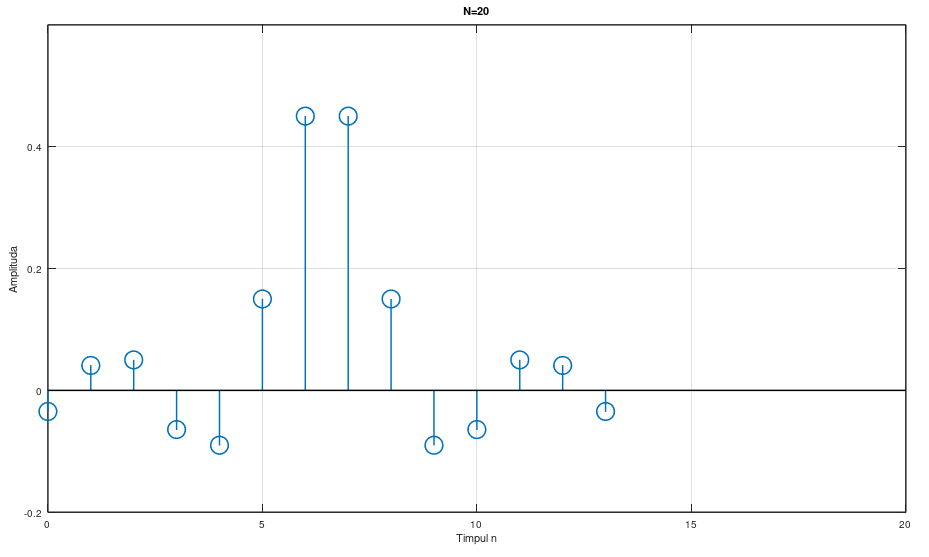


Figura 1 – Caracteristicile frecvențiale ale filtrului ideal.

Programul P4\_1 generează și afișează răspunsul impulsului unui filtru ideal, folosind frecvența de tăiere fc. Prin funcția sinc, se calculează forma răspunsului impulsului, iar rezultatul este afișat cu stem. Astfel, se obține o reprezentare grafică a caracteristicilor frecvențiale ale filtrului ideal.

**Exercițiu 2**

Programul calculează răspunsul propriu al filtrului de frecvenţă joasă.

% Programul P4\_2

% Rspunsul propriu al filtrului de frecvenţă joasă.

clf;

M=2;

num=ones(1,M)/M;

w=0:pi/255:pi;

h=freqz(num,1,w);

g=20\*log10(abs(h));

plot(w/pi,g); grid

axis([0 1 -50 0.5]);

xlabel('\omega /\pi'); ylabel('Adaugare');

title(['M = ',num2str(M)]);

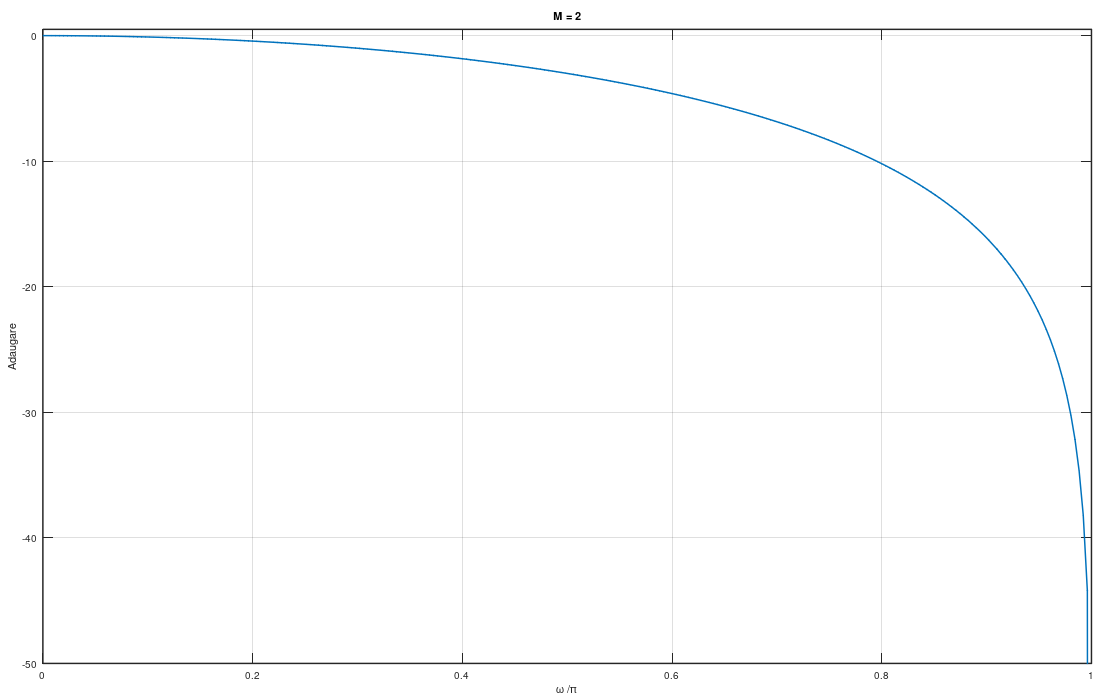


Figura 2 – Caracteristica de atenuare ale filtrului de frecvență joasă.

Programul P4\_2 calculează și afișează răspunsul în frecvență al unui filtru de frecvență joasă cu o lungime a ferestrei de filtrare dată de M. Se utilizează funcția freqz pentru a obține răspunsul în frecvență al filtrului. Apoi, amplitudinea răspunsului în frecvență este convertită în unități decibel folosind 20\*log10(abs(h)), iar rezultatul este afișat grafic folosind plot. Astfel, se obține o reprezentare vizuală a caracteristicilor de atenuare ale filtrului de frecvență joasă.

**Exercițiu 3**

Cu ajutorul programul dat putem analiza proprietăţile celor patru tipuri de funcţii de transfer care caracterizează filtrele:

Tipul 1: Răspunsul impuls simetric de lungime impară;

Tipul 2: Răspunsul impuls simetric de lungime pară;

Tipul 3: Răspunsul impuls asimetric de lungime impară;

Tiplu 4: Răspunsul impuls asimetric de lungime pară.

% Programul P4\_3

% Punctele zero FIR filtrelor in faza liniara

clf;

b=[1 -8.5 30.5 -63];

num1=[b 81 fliplr(b)];

num2=[b 81 81 fliplr(b)];

num3=[b 0 -fliplr(b)];

num4=[b 81 -81 -fliplr(b)];

n1=0:length(num1)-1;

n2=0:length(num2)-1;

subplot(2,2,1); stem(n1,num1);

xlabel('Timpul n'); ylabel('Amplituda'); grid

title('Tipul 1 FIR filtrului');

subplot(2,2,2); stem(n2,num2);

xlabel('Timpul n'); ylabel('Amplituda'); grid

title('Tip 2 FIR filtrului');

subplot(2,2,3); stem(n1,num3);

xlabel('Timpul n'); ylabel('Amplituda'); grid

title(' Tip 3 FIR filtrului ');

subplot(2,2,4); stem(n2,num4);

xlabel('Timpul n'); ylabel('Amplituda'); grid

18

title('Tip 2 FIR filtrului');

pause

subplot(2,2,1); zplane(num1,1);

title(' Tip 1 FIR filtrului ');

subplot(2,2,2); zplane(num2,1);

title(' Tip 2 FIR filtrului ');

subplot(2,2,3); zplane(num3,1);

title('Tip 3 FIR filtrului');

subplot(2,2,4); zplane(num4,1);

title('Tip 4 FIR filtrului');

disp('Zeroul FIR filtrului de Tip 1');

disp(roots(num1));

disp(' Zeroul FIR filtrului de Tip 2');

disp(roots(num2));

disp(' Zeroul FIR filtrului de Tip 3');

disp(roots(num3));

disp(' Zeroul FIR filtrului de Tip 4');

disp(roots(num4));

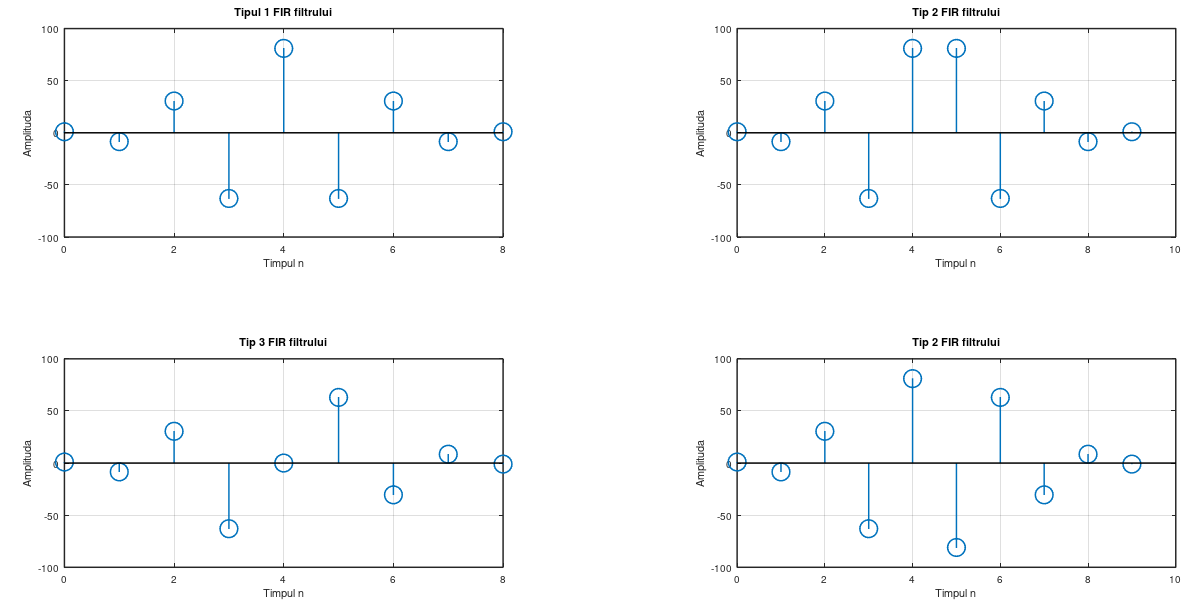


Figura 3 – Răspunsul impuls

Programul generează și afișează punctele zero pentru fiecare tip de filtru FIR, atât în domeniul timpului cât și în domeniul frecvenței. Acest lucru permite o analiză detaliată a structurii și comportamentului fiecărui tip de filtru FIR, oferind informații despre modul în care acestea vor răspunde la diferite semnale de intrare.

În continuare, programul utilizează comanda zplane pentru a afișa punctele zero ale filtrului în planul complex. Aceste puncte zero reprezintă locațiile în planul complex în care funcția de transfer a filtrului este zero. Prin urmare, ele joacă un rol important în determinarea caracteristicilor de filtrare ale unui filtru.

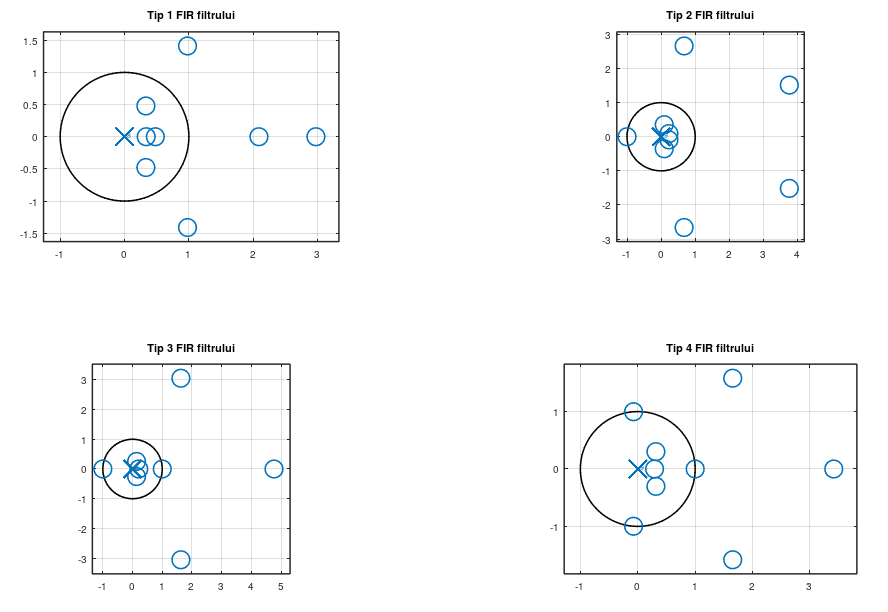


Figura 4 – Cele 4 tipuri de filtre prezentate în plan complec.

Zeroul FIR filtrului de Tip 1

2.9744 + 0i

2.0888 + 0i

0.9790 + 1.4110i

0.9790 - 1.4110i

0.3319 + 0.4784i

0.3319 - 0.4784i

0.4787 + 0i

0.3362 + 0i

Zeroul FIR filtrului de Tip 2

3.7585 + 1.5147i

3.7585 - 1.5147i

0.6733 + 2.6623i

0.6733 - 2.6623i

-1.0000 + 0i

0.0893 + 0.3530i

0.0893 - 0.3530i

0.2289 + 0.0922i

0.2289 - 0.0922i

Zeroul FIR filtrului de Tip 3

4.7627 + 0i

1.6279 + 3.0565i

1.6279 - 3.0565i

-1.0000 + 0i

1.0000 + 0i

0.1357 + 0.2549i

0.1357 - 0.2549i

0.2100 + 0i

Zeroul FIR filtrului de Tip 4

3.4139 + 0i

1.6541 + 1.5813i

1.6541 - 1.5813i

-0.0733 + 0.9973i

-0.0733 - 0.9973i

1.0000 + 0i

0.3159 + 0.3020i

0.3159 - 0.3020i

0.2929 + 0i

**Exercițiu 4**

Acest program prezintă cercetarea stabilităţii a filtrului numeric IIR. Stabilitatea filtrului− ereprezintă o calitate foarte importantă a filtrului. Filtrul numeric IIR este stabil dacă polii funcţiei de transfer se află în interiorul cercului unitate.

% Program\_4

% Test de verificare a stabilitatii

% Introduceti coeficientii de numitor

den = input('Introduceti coeficientii de numitor: ');

% Verificam daca numarul de coeficienti este egal cu 2

if numel(den) ~= 2

error('Va rugam introduceti doi coeficienti pentru numitor.');

end

% Calculul radacinilor polinomului de numitor

poles = roots(den);

% Afisarea polilor si evaluarea stabilitatii

disp('Parametrii testului de stabilitate:');

disp(poles);

if all(abs(poles) < 1)

fprintf('Filtrul este stabil.\n');

else

fprintf('Filtrul nu este stabil.\n');

end

Acest program are rolul de a verifica stabilitatea unui filtru numeric IIR, adică un filtru cu răspuns la impuls infinit. Testul de stabilitate este crucial în proiectarea și implementarea filtrelor, deoarece un filtru instabil poate produce rezultate imprevizibile sau chiar divergență în procesul de filtrare. La început, utilizatorul este invitat să introducă coeficienții polinomului de numitor ai filtrului. După aceasta, programul verifică dacă numărul de coeficienți este exact 2, deoarece un filtru IIR simplu are de obicei doi coeficienți în numitor. Dacă numărul de coeficienți este diferit, programul afișează o eroare și se încheie. În continuare, programul calculează rădăcinile polinomului de numitor folosind funcția roots(). Aceste rădăcini reprezintă polii funcției de transfer a filtrului. Acești poli sunt apoi afișați pentru a oferi utilizatorului o imagine a distribuției lor pe planul complex. Ulterior, programul evaluează stabilitatea filtrului prin verificarea dacă toate rădăcinile polinomului de numitor se află în interiorul cercului unitate din planul complex. Dacă toate polii au o valoare absolută mai mică decât 1, filtrul este considerat stabil și se afișează un mesaj corespunzător. În caz contrar, când cel puțin un pol are o valoare absolută mai mare sau egală cu 1, se afișează un alt mesaj care indică faptul că filtrul nu este stabil. Astfel, acest program oferă o modalitate rapidă și simplă de a evalua stabilitatea unui filtru numeric IIR, contribuind la procesul de proiectare și dezvoltare a sistemelor de filtrare.

Introduceti coeficientii de numitor: [20,5]

Parametrii testului de stabilitate:

-0.2500

Filtrul este stabil.

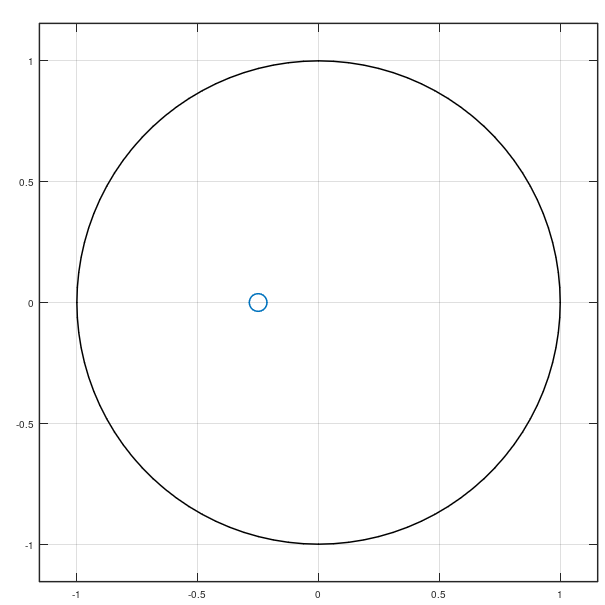


Figura 5 – Filtru stabil.

Introduceti coeficientii de numitor: [5,25]

Parametrii testului de stabilitate:

-5

Filtrul nu este stabil.

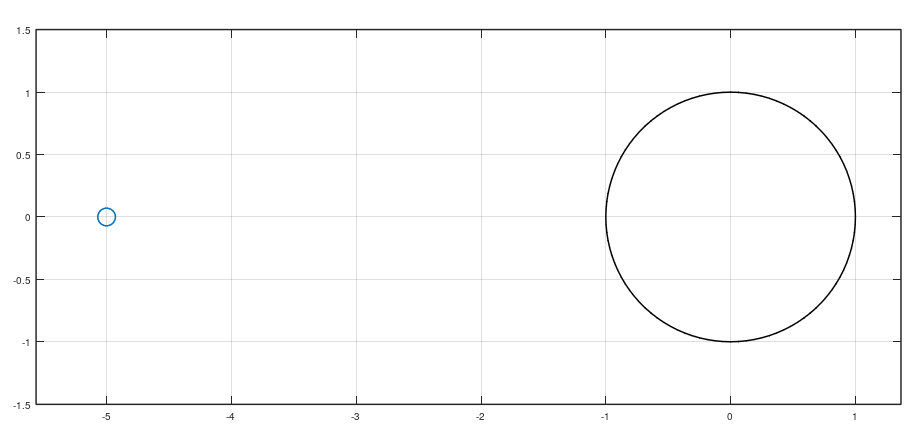


Figura 6 – Filtru nestabil.

**Concluzii**

Realizarea acestei lucrări a implicat implementarea a patru programe distincte în MATLAB, fiecare având un scop specific în analiza și evaluarea filtrelor digitale. În primul exercițiu, s-a calculat și afișat răspunsul impulsului unui filtru ideal, utilizând funcția sinc pentru a aproxima acest răspuns. Graficul rezultat a ilustrat caracteristicile frecvențiale ale filtrului ideal. Al doilea exercițiu a constat în calculul și afișarea răspunsului în frecvență al unui filtru de frecvență joasă, folosind funcția freqz. Acest lucru a furnizat o reprezentare vizuală a caracteristicilor de atenuare ale filtrului. Exercițiul al treilea a prezentat analiza a patru tipuri de filtre FIR (Răspunsul la impulsuri finite), evidențiind diferențele între acestea în ceea ce privește simetria și lungimea răspunsului la impuls. A fost utilizată funcția zplane pentru a afișa polii și zerourile fiecărui tip de filtru în planul complex. Ultimul exercițiu a avut ca scop verificarea stabilității unui filtru numeric IIR, determinând dacă polii săi se află în interiorul cercului unitate pe planul complex. Programul a oferit un mesaj corespunzător în funcție de rezultatul verificării. Prin intermediul acestor programe, s-au evidențiat diverse aspecte legate de proiectarea și evaluarea filtrelor digitale, contribuind la înțelegerea caracteristicilor și comportamentului acestora în diverse contexte.