MINISTERUL EDUCAŢIEI al REPUBLICII MOLDOVA

UNIVERSITATEA TEHNICĂ A MOLDOVEI

Catedra Automatică şi Tehnologii Informaţionale

Îndrumar pentru lucrări de laborator
la disciplina

Prelucrarea semnalelor folosind MATLAB

Chişinău 2011

Elaborat dr.conf. ***Romanenko Alexandru***
dr.conf. ***Fiodorov Ion***
dr.conf Ciorba Dumitru
lect.sup.magistr. Melnic Radu

Cuprins

Lucrare de laborator Nr.5

Tema: Sisteme discrete în timp continuu sau discret

Scopul lucrării: De a învăţa cum să creăm diferite sisteme în conformitate cu sistemul iniţial.

Noţiuni teoretice

Sistemele discrete în timp convertesc semnalul de intrare în domeniul de timp cu scopul de a obţine la ieşire un semnal cu proprietăţi dorite. Asupra semnalului de intrare se aplică diferiţi, care constau în operaţii simple. Scopul acestei lucrări de laborator constă în prezentarea cîtorva tipuri simple de discrete, cu prorprietăţile lor.

Într-un sistem liniar în timp discret pentru un semnal de intrare $x[n]=\alpha x_1[n]+\beta x_2[n]$ ca răspuns vom avea $y[n]=\alpha y_1[n]+\beta y_2[n]$ unde $y_1[n]$ şi $y_2[n]$ sunt răspunsuri la secvenţele corespunătoare $x_1[n]$ и $x_2[n]$.

Într-un sistem discret în timp continuu ca răspuns la semnalul de intrare $x[n]=x_1[n-n_0]$ va fi semnalul $y[n]=y_1[n-n_0]$ unde n_0 –număr intreg nenul şi $y_1[n]$ raspuns la $x_1[n]$.

Sistemul continuu după valori şi în timp cotinuu (system continuu ,Linear Time-Invariant-LTI) satisfice ambele criterii de liniaritate şi continuitate.

Dacă $y_1[n]$ şi $y_2[n]$ sunt răspunsuri ale unui sistem discret cauzal la intrările corespunzătoare $u_1[n]$ şi $u_2[n]$, atunci $u_1[n]= u_2[n]$ cînd $n<N$, implică $y_1[n]=y_2[n]$ cînd $n<N$.

Comenzile MATLAB utilizate:

Comenzi de uz general
`disp`

Operatori şi simboluri speciale
`:    .    +    -    *    /    ;    %`

Construcţii sintactice
`break    end    for    if    input`

Matrici elementare şi operaţiile asupra lor
`ones    pi    zeros`

Funcţii elementare
`cos    abs`

Funcţii de interpolare şi poliomiale
`conv`

Grafică bidimensională
`axis    plot    stem    title    xlabel    ylabel`

Funcţii grafice de uz general
`clf    subplot`

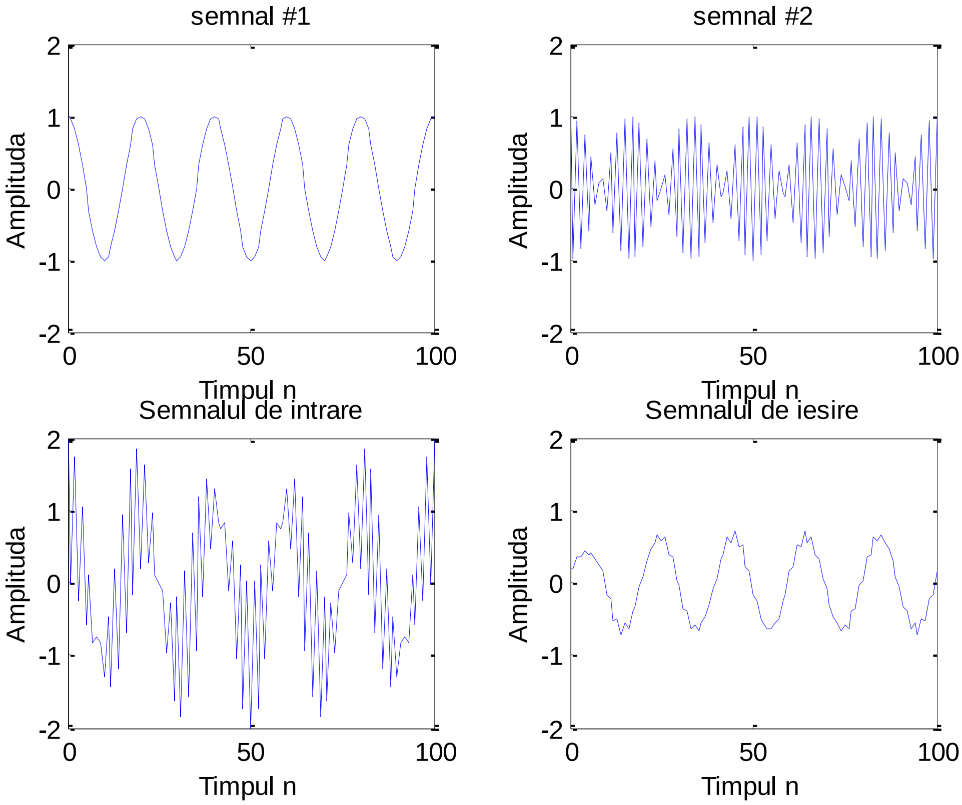
Funcţii de lucru cu şirui
`num2str`

Funcţii pachetului de prelucrare a semnalelor
`filter    impz`

Programul 5_1

Prezentarea unui filtru simplu cu M-vîrfuri, numărul cărora se introduce de către utilizator.

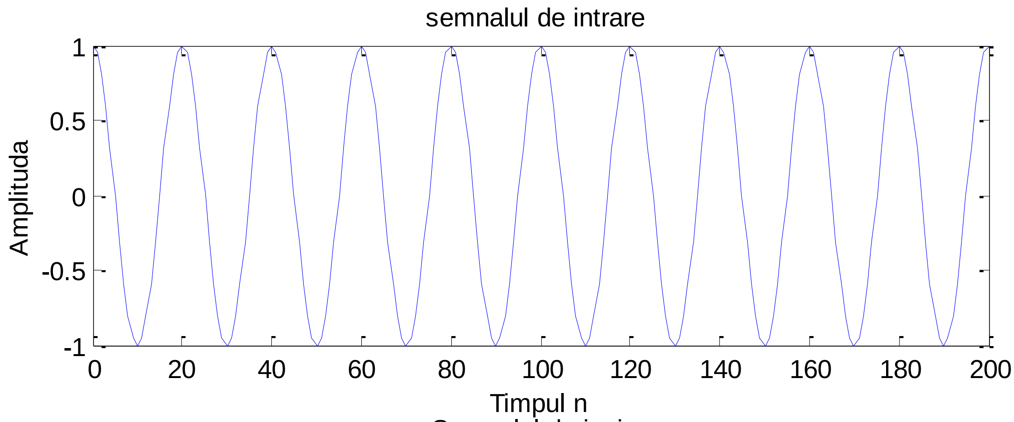
```
% Programul P2_1
% Simularea filtrului FIR de o lungime arbitrara
% generarea semnalului de intrare
clf;   n=0:100;
s1=cos(2*pi*0.05*n); % semnal de o frecventa joasa
s2=cos(2*pi*0.47*n); % semnal de o frecventa inalta
x=s1+s2;
% Realiyarea filtrului FIR
M=input('Lungimea dorita a filtrului = ');
num=ones(1,M);
y=filter(num,1,x)/M;
% Afisarea semnalelor de intare si de isire
subplot(2,2,1);
plot(n,s1);
axis([0, 100, -2, 2]);
xlabel('Timpul n'); ylabel('Amplituda');
title('semnal #1');
subplot(2,2,2);
plot(n,s2);
axis([0, 100, -2, 2]);
xlabel('Timpul n'); ylabel('Amplituda');
title('semnal #2');
subplot(2,2,3);
plot(n,x);
axis([0, 100, -2, 2]);
xlabel('Timpul n'); ylabel('Amplituda');
title('Semnalul de intrare');
subplot(2,2,4);
plot(n,y);
axis([0, 100, -2, 2]);
xlabel('Timpul n'); ylabel('Amplituda');
title('Semnalul de iesire');
axis;
```

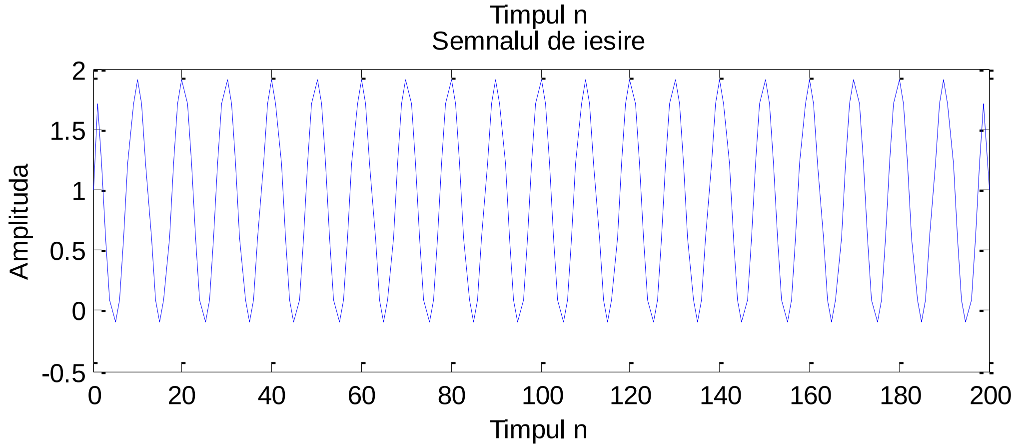


Programul 5_2

Programul de mai jos poate fi folosit pentru generarea semnalului de intrare x[n], compus ditr-o secenta sinusoidala.

```
% Programul P2_2
% generarea semnalului sinusoidal de intrare
clf;
n=0:200;
x=cos(2*pi*0.05*n);
% Calcularea semnalului de iesire
x1 = [x 0 0]; % x1[n] = x[n+1]
x2 = [0 x 0]; % x2[n] = x[n]
x3 = [0 0 x]; % x3[n] = x[n-1]
y=x2.*x2 + x1.*x3;
y=y(2:202);
% Vizualizarea semnalului de intre si a celui de iesire subplot(2,1,1);
plot(n,x);
xlabel('Timpul n'); ylabel('Amplituda');
title('semnalul de intrare ');
subplot(2,1,2);
plot(n,y);
xlabel('Timpul n'); ylabel('Amplituda');
title('Semnalul de iesire');
```





Programul 5_3

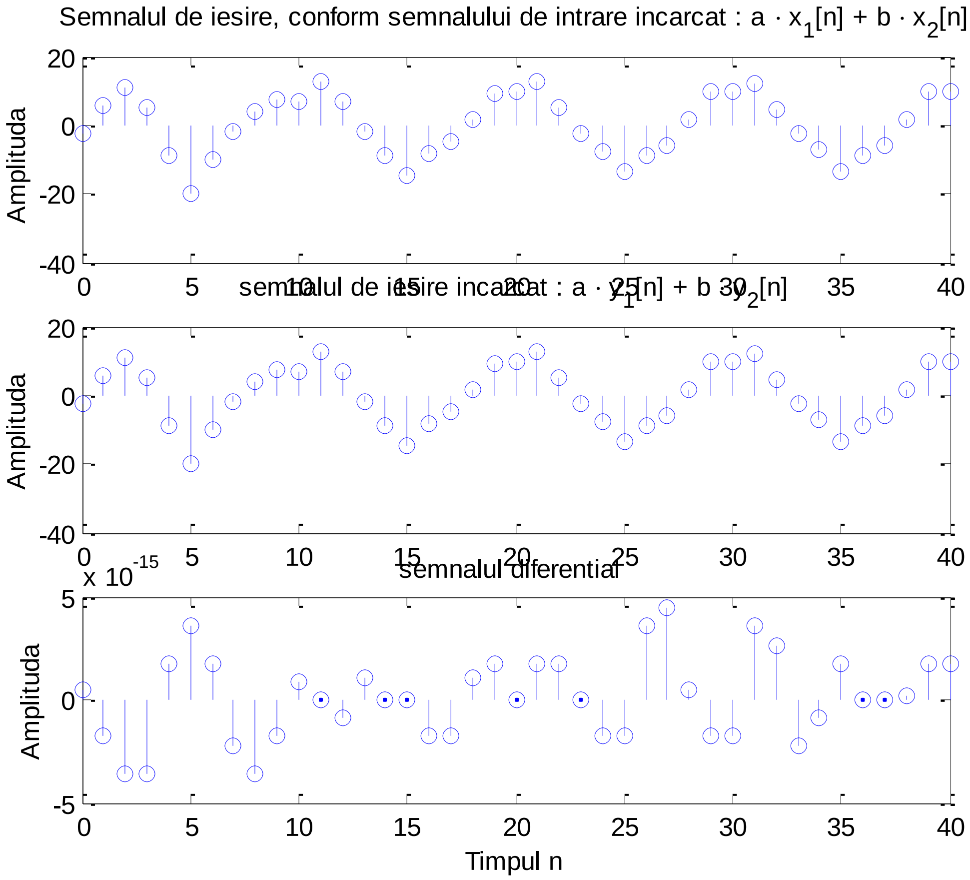
În programul acesta este realizat sistemul reprezentat prin următoarea expresie:

$y[n]-0.4y[n-1]+0.75y[n-2]=2.2403x[n]+2.4908x[n-1]+2.2403x[n-2]$.

Se generează trei secvenţe de intrare $x_1[n]$, $x_2[n]$ и $x[n]=a*x_1[n]+b*x_2[n]$. Se calculează trei secvenţe de ieşire $y_1[n]$, $y_2[n]$ и $y[n]$, cu reprezentarea lor grafică.

```
% Programul  P2_3
% Generarea secvenţelor de iesire
clf;
n=0:40;
a=2; b=-3;
x1=cos(2*pi*0.1*n);
x2=cos(2*pi*0.4*n);
x=a*x1+b*x2;
num=[2.2403 2.4908 2.2403];
den=[1 -0.4 0.75];
ic=[0 0]; % initializarea
y1=filter(num,den,x1,ic); % Calcularea semnalului de iesire y1[n]
y2=filter(num,den,x2,ic); % Calcularea semnalului de iesire y2[n]
y=filter(num,den,x,ic); % Calcularea semnalului de iesire y[n]
yt=a*y1+b*y2;
d=y-yt; % calcularea abaterii d[n]
%Vizualizarea semnalelor de iesire si a semnalului de abatere
subplot(3,1,1);
stem(n,y);
ylabel('Amplituda ');
      title('Semnalul de iesire, conform semnalului de intrare incarcat : a
  \cdot x_{1}[n] + b \cdot x_{2}[n]');
subplot(3,1,2);
stem(n,yt);
ylabel('Amplituda');
title('semnalul de iesire incarcat : a \cdot y_{1}[n] + b \cdot y_{2}[n]');
subplot(3,1,3);
```

```
stem(n,d);
xlabel('Timpul n'); ylabel('Amplituda');
title('semnalul diferential ');
```



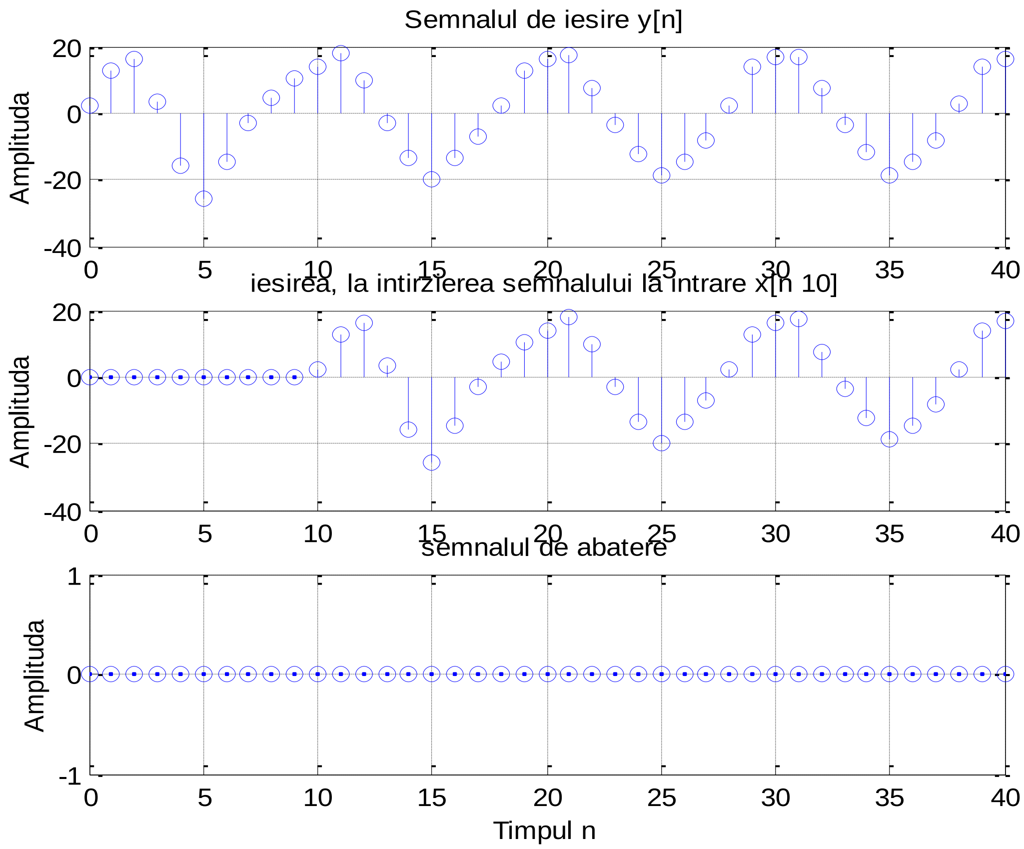
Programul 5_4

Pentru această expresie se crează sistemul ce o simulează:

y[n]-0.4y[n-1]+0.75y[n-2]=2.2403x[n]+2.4908x[n-1]+2.2403x[n-2]. Două secvenţe de intrare diferite x[n] şi x[n-D]. Se calculează şi se vizualizează corespunzător doua secvenzede ieşire şi abaterea y1[n] – y2[n+D].

```
% Programul P2_4
% Generarea secventelor de intrare
clf;  n=0:40; D=10; a=3.0; b=-2;
x=a*cos(2*pi*0.1*n)+b*cos(2*pi*0.4*n);
xd=[zeros(1,D) x];
num=[2.2403 2.4908 2.2403];  den=[1 -0.4 0.75];
ic=[0 0]; % setarea conditiilor initiale
%Calcularea semnalului de iesire y[n]
y=filter(num,den,x,ic);
% Calcularea semnalului de iesire yd[n]
yd=filter(num,den,xd,ic);
% Calcularea semnalului deabatere d[n]
d=y-yd(1+D:41+D);
%Afisarea graficelor semnalelor de iesire
subplot(3,1,1);
stem(n,y);
ylabel('Amplituda');
title('Semnalul de iesire y[n]'); grid;
subplot(3,1,2);
stem(n,yd(1:41));
ylabel('Amplituda');
```

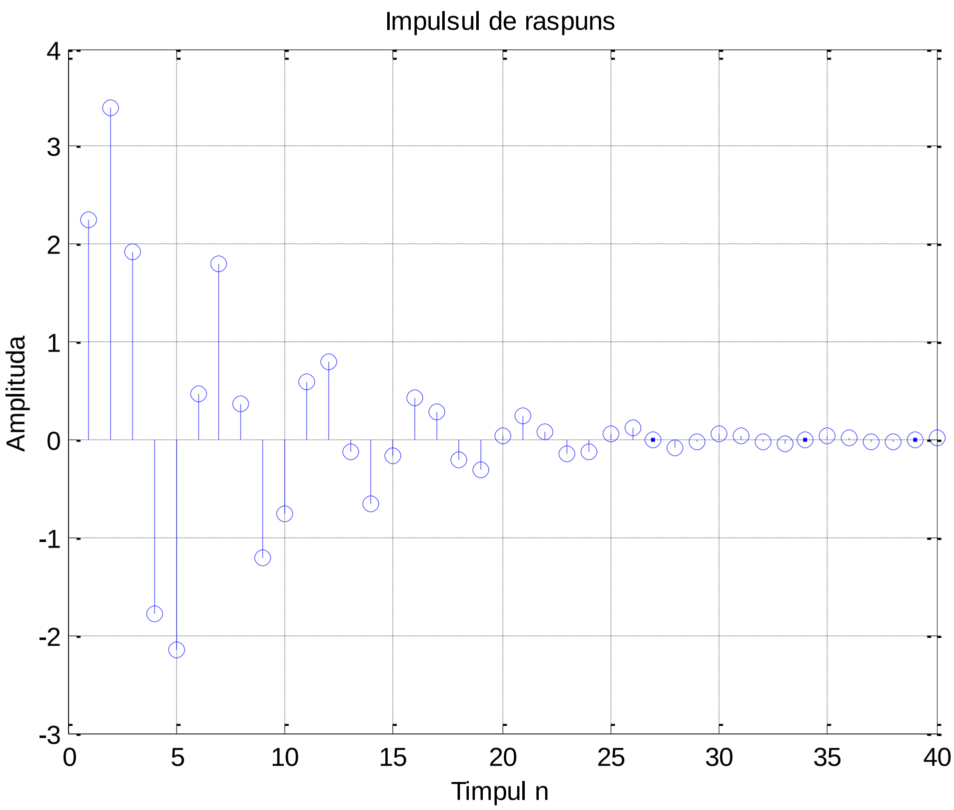
```
      title(['iesirea,la intirzierea semnalului la intrare
 x[n',num2str(D),']']); grid;
subplot(3,1,3);
stem(n,d);
xlabel('Timpul n'); ylabel('Amplituda');
title('semnalul de abatere'); grid;
```



Programul 5_5

Aici se calculează şi se afişează la ecran impulsul răspunsului sistemului, corespunzător expresiei: y[n]-0.4y[n-1]+0.75y[n-2]=2.2403x[n]+2.4908x[n-1]+2.2403x[n-2].

```
% Programul P2_5
% calcularea impulsului de raspuns y
clf;
N=40;
num=[2.2403 2.4908 2.2403];
den=[1 -0.4 0.75];
y=impz(num,den,N);
% Reprezentareagrafică a impulsului de raspuns
stem(y);
xlabel('Timpul n'); ylabel('Amplituda');
title('Impulsul de raspuns'); grid;
```



Programul 5_6

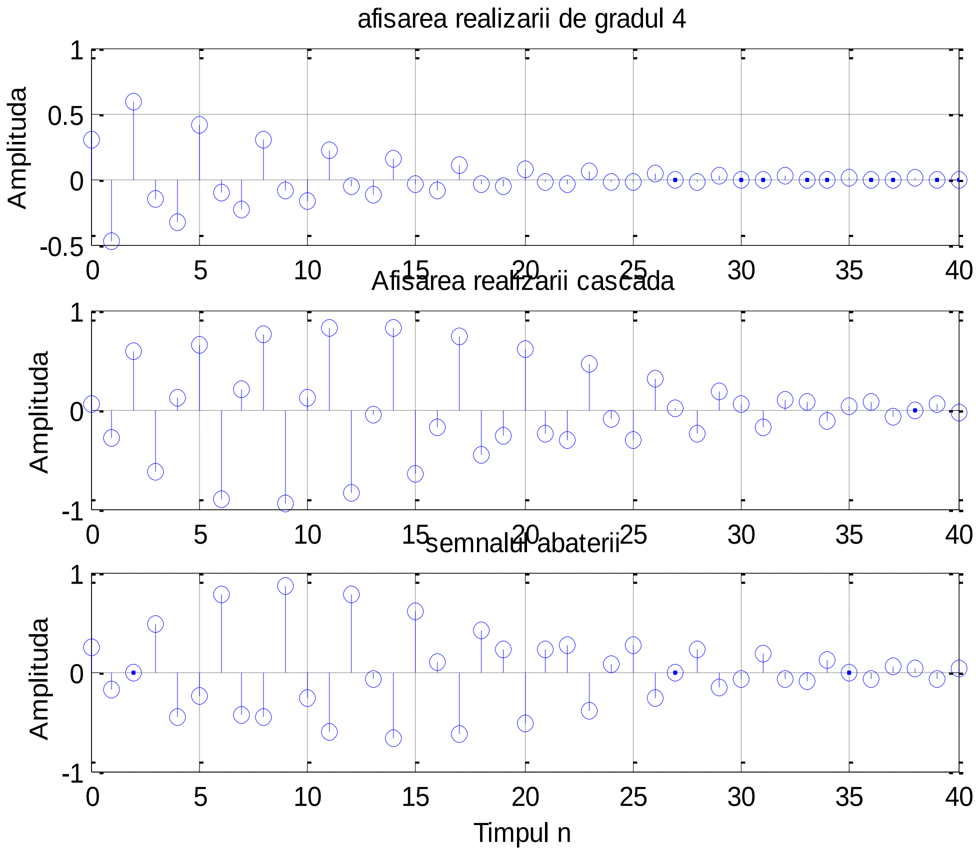
Programul acesta realizează sistemul de 4 conditii: y[n]+1.6y[n-1]+2.28y[n-2]+1.325y[n-3]+0.68y[n-4] = 0.06x[n]-0.19x[n-1]+0.27x[n-2]-0.26x[n-3]+0.12x[n-4] şi Sisteme cascade :

Etapa 1: y1[n]+0.9y1[n-1]+0.8y1[n-2]=0.3x[n]-0.3x[n-1]+0.4x[n-2];

Etapa 2: y2[n]+0.7y2[n-1]+0.85y2[n-2]=0.2y1[n]-0.5y1[n-1]+0.3y1[n-2];

```
% Programul P2_6
% Realizarea cascadelor
clf;
x=[1 zeros(1,40)]; % Generarea semnalului de intrare
n=0:40;
% Coeficientii sistemului de gradul 4
den=[1 1.6 2.28 1.325 0.68];
num=[0.06 -0.19 0.27 -0.26 0.12];
% calcularea semnalului de ieasire asistemului de gradul 4 y=filter(num,den,x);
% Coefocoentii sistemelordegradul 2
num1=[0.3 -0.2 0.4]; den1=[1 0.9 0.8];
num2=[0.2 -0.5 0.3]; den2=[1 0.7 0.85];
% Semnalul de iesire y1[n] a primei etape a cascadei
y1=filter(num1,den1,x);
% Semnalul deiesire y2[n] etapei a doua a cascadei
y2=filter(num2,den2,y1);
% abaterea dintre y[n] si y2[n]
d=y-y2;
% Graficele semnalelor de iesire si a abaterii
subplot(3,1,1);
stem(n,y);
ylabel('Amplituda');
title('afisarea realizării de gradul 4'); grid;
subplot(3,1,2);
stem(n,y2);
ylabel('Amplituda');
title('Afisarea realizarii cascada'); grid;
subplot(3,1,3);
```

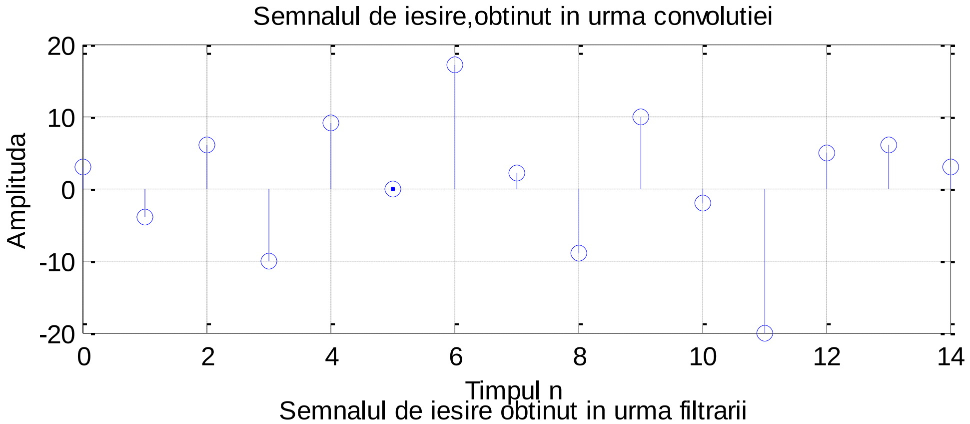
```
stem(n,d);
xlabel('Timpul n'); ylabel('Amplituda ');
title('semnalul abaterii'); rid;
```

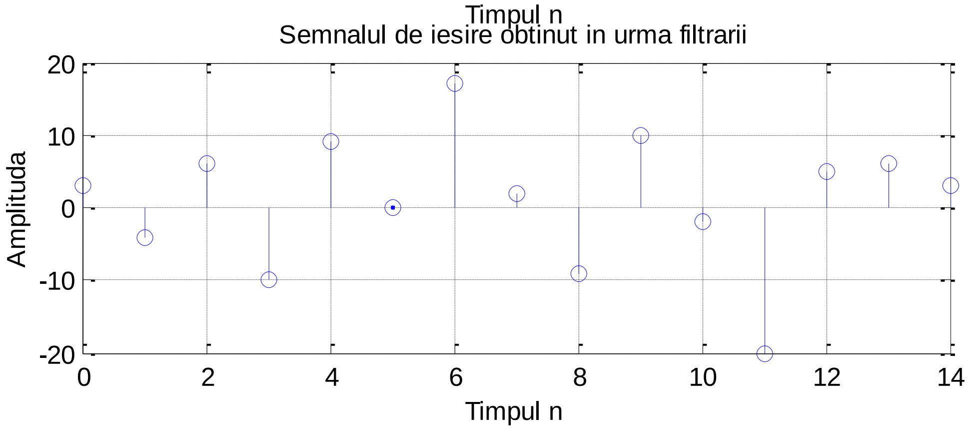


Programul 5_7

Operaţiunea de convoluţie se realizează în MATLAB cu comanda conv. Pentru aceasta se folosesc două secvenţe finite şi de aceeaşi lungime.

```
% Programul P2_7
clf;
h=[3 2 1 -2 1 0 -4 0 3]; % raspuns de impuls
x=[1 -2 3 -4 3 2 1]; % secventa deintrare
y=conv(h,x);
n=0:14;
subplot(2,1,1);
stem(n,y);
xlabel('Timpul n'); ylabel('Amplituda');
title('Semnalul de iesire,obtinut in urma convolutiei'); grid;
x1=[x zeros(1,8)];
y1=filter(h,1,x1);
subplot(2,1,2);
stem(n,y1);
xlabel('Timpul n'); ylabel('Amplituda');
title('Semnalul de iesire obtinut in urma filtrarii'); grid;
```

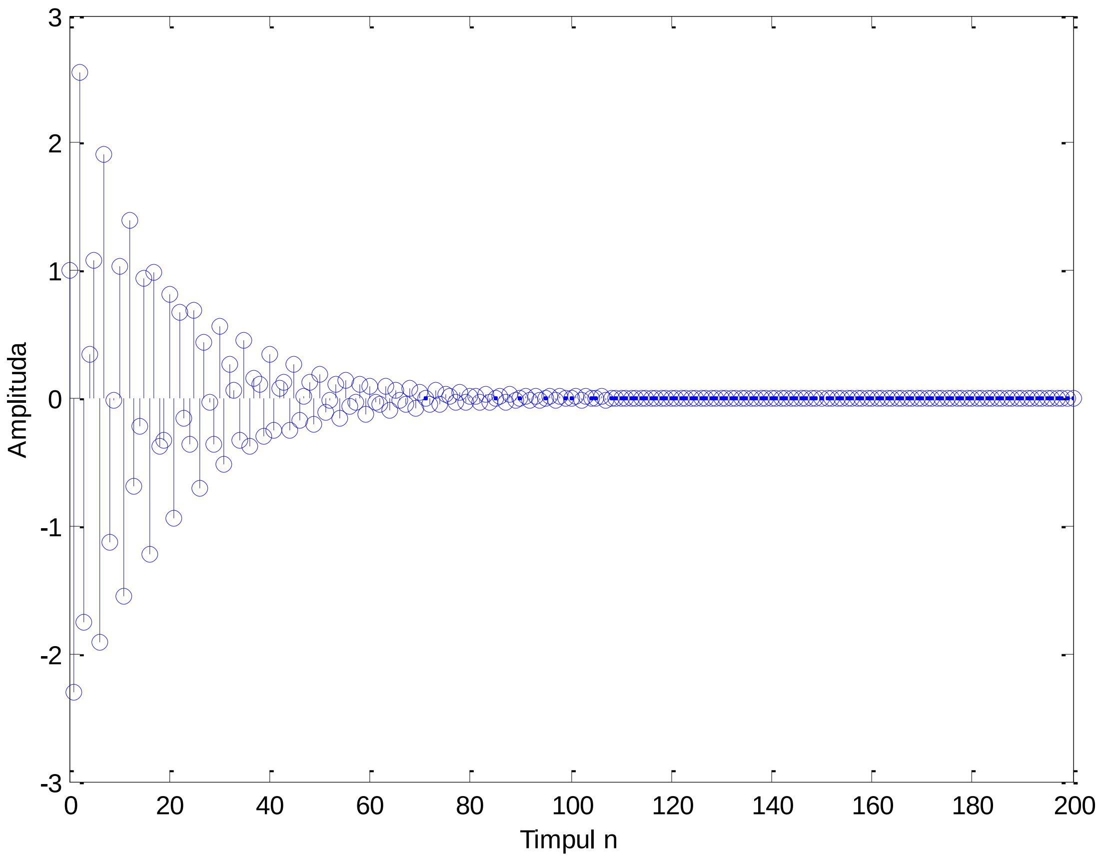




Programul 5_8

Programul calculează suma valorilor absolute ale raspunsului inpuls

```
% Programul P2_8
      % Testarea stabilitatii bazat pe suma valorilor absolute ale elementelor
  raspuns impuls
clf;
num=[1 -0.8]; den=[1 1.5 0.9];
N=200;
h=impz(num,den,N+1);
parsum=0;
for k=1:N+1;
    parsum=parsum+abs(h(k));
    if abs(h(k))<10^(-6), break, end
end
% Afisarea răspuns impuls
n=0:N;
stem(n,h);
xlabel('Timpul n'); ylabel('Amplituda');
disp('Valoarea='); disp(abs(h(k))); %Afisarea valorilor
```



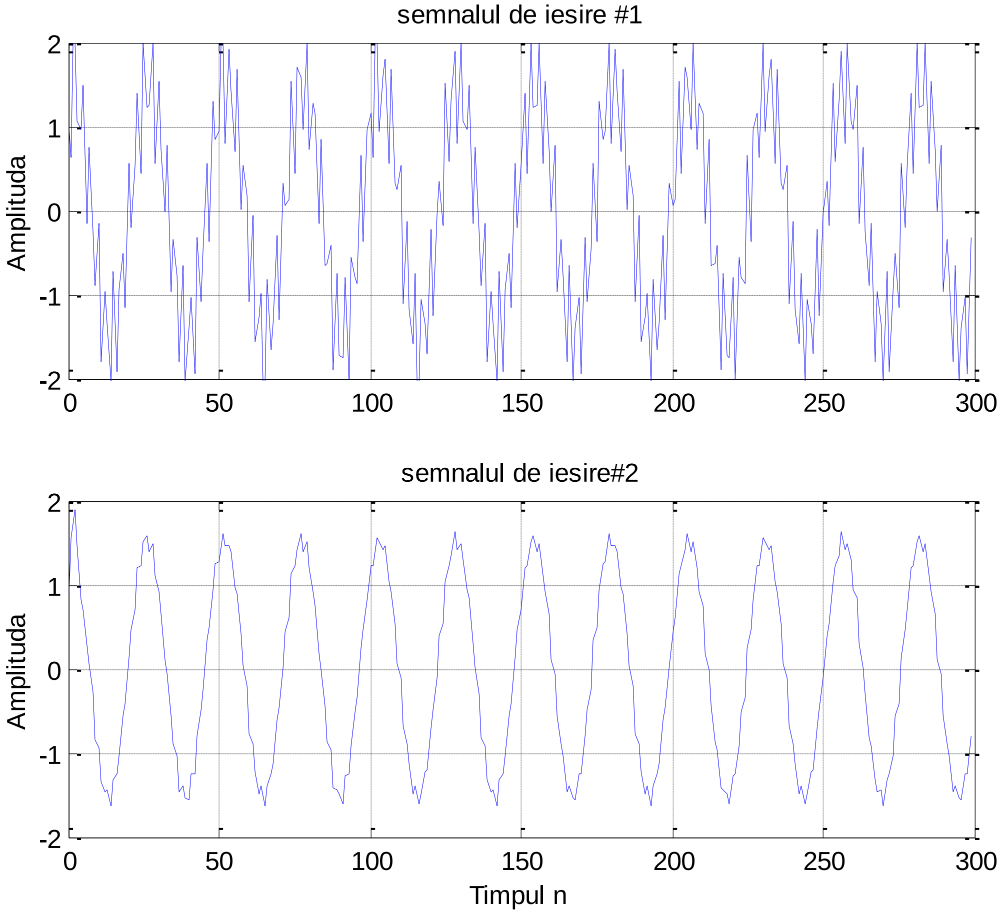
Programul 5_9

Aici se prezintă două sisteme:

$$y[n] = 0.5x[n]+0.27x[n-1]+0.77x[n-2]$$

şi $$y[n] = 0.45x[n] +0.5x[n-1]+0.45x[n-2]+0.53y[n-1]-0.46y[n-2].$$

```
% Programul P2_9
% Generarea semnalului deintare
clf;
n=0:299;
x1=cos(2*pi*10*n/256);
x2=cos(2*pi*100*n/256);
x=x1+x2;
% Calcularea secvenţelor de iesire
num1=[0.5 0.27 0.77];
y1=filter(num1,1,x); % Iesirea sistemului #1
den2=[1 -0.53 0.46];
num2=[0.45 0.5 0.45];
y2=filter(num2,den2,x); % Iesirea sistemului #2
% Graficele secventelor de iesire
subplot(2,1,1);
plot(n,y1); axis([0 300 -2 2]);
ylabel('Amplituda');
title('semnalul de iesire #1'); grid;
subplot(2,1,2);
plot(n,y2); axis([0 300 -2 2]);
xlabel('Timpul n'); ylabel('Amplituda');
title(semnalul de iesire#2'); grid;
```


semnalul de iesire #1
Amplituda
2
1
0
-1
-2
0
50
100
150
200
250
300
semnalul de iesire#2
Amplituda
2
1
0
-1
-2
0
50
100
150
200
250
300
Timpul n

Lucrare de laborator Nr.6

Tema: Sisteme discrete liniare în timp continuu cercetate în domeniul de frecvenţă
Scopul lucrăii: De studiat proprietăţile de bază ale sistemelor discrete în timp continuu în domeniul de frecvenţă.

Noţiuni teoretice

Orice sistem discret liniar în timp continuu este complet caracterizat de domeniul de frecvenţă în secvenţa de impulsuri de raspuns. Astfel semnalul de ieşire poate fi obţinut pentru orice sistem de aşa tip prin convoluţia secvenţei de intrare cu secvenţa impulsului de raspuns al ei. Anumite categorii de astfel de sisteme, de asemenea, pot fi caracterizate printr-o ecuaţie (de diferenţiere) liniară cu coefcienţi constanţi. Pentru aceste categorii de sisteme semnalul de ieşire poate fi calculat recursiv pentru orice secvenţă de intrare. Aplicînd DTFT sau transformata-z la rezultatul convolutiei sau la ecuaţia de diferenţiere, sistemele discrete liniare şi continue în timp pot fi carcterizate şin domeniul de frecvenţă. Acest tip de specificarea sistemelor furnizează informaţii suplimentare despre comportamentul sistemului, pe lîngă faptul că o astfel de specificare a sistemelor acordă posibilităţi mai uşoare de proiectări, realizări şi utilizări în diferite aplicaţii.

Comenzile MATLAB utilizate:

Comenzi de uz general
`disp`

Operatori şi simboluri speciale
`:   .   +   -   *   /   ;   %`

Construcţii sintactice
`function     pause`

Matrici elementare şi operaţiile asupra lor
`fliplr     pi`

Funcţii elementare
`cos   abs   angle   imag   log10  real`

Grafică bidimensională
`axis     plot     stem     title     xlabel     ylabel    grid`

Funcţii grafice de uz general
`clf   subplot`

Funcţii pachetului de prelucrare a semnalelor
`filter   impz   filtfilt   fregz   grpdelay   poly2rc   sinc   zplane`

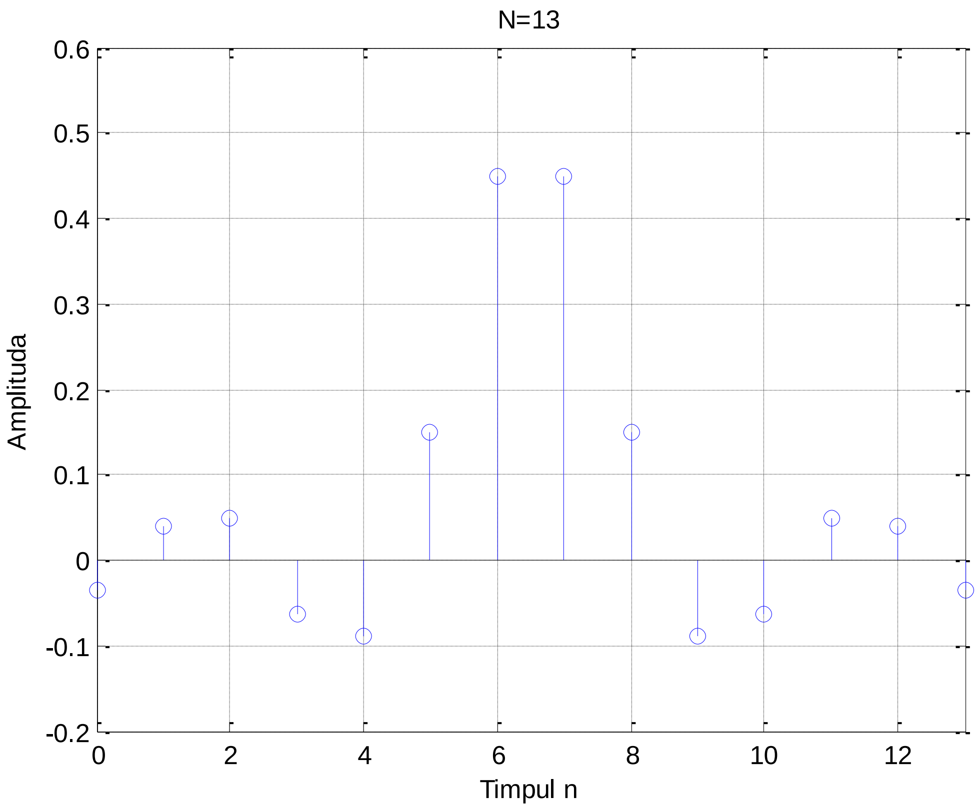
Programul 6_1

În acestprogram se calculează aproximarea de sus. Se foloseşte funcţia *sinc* din 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=13'); axis([0 13 -0.2 0.6]);
```

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

Rezultatul programului:

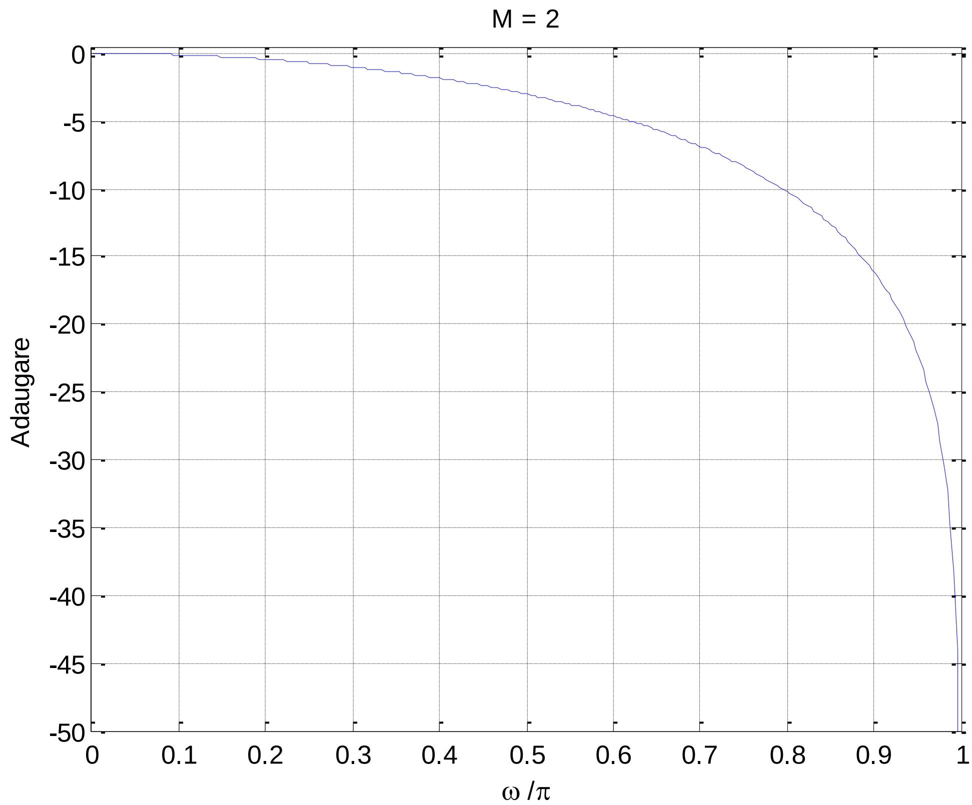


Programul 6_2

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)]);
```

Rezultatul programului:



Programul 6_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;

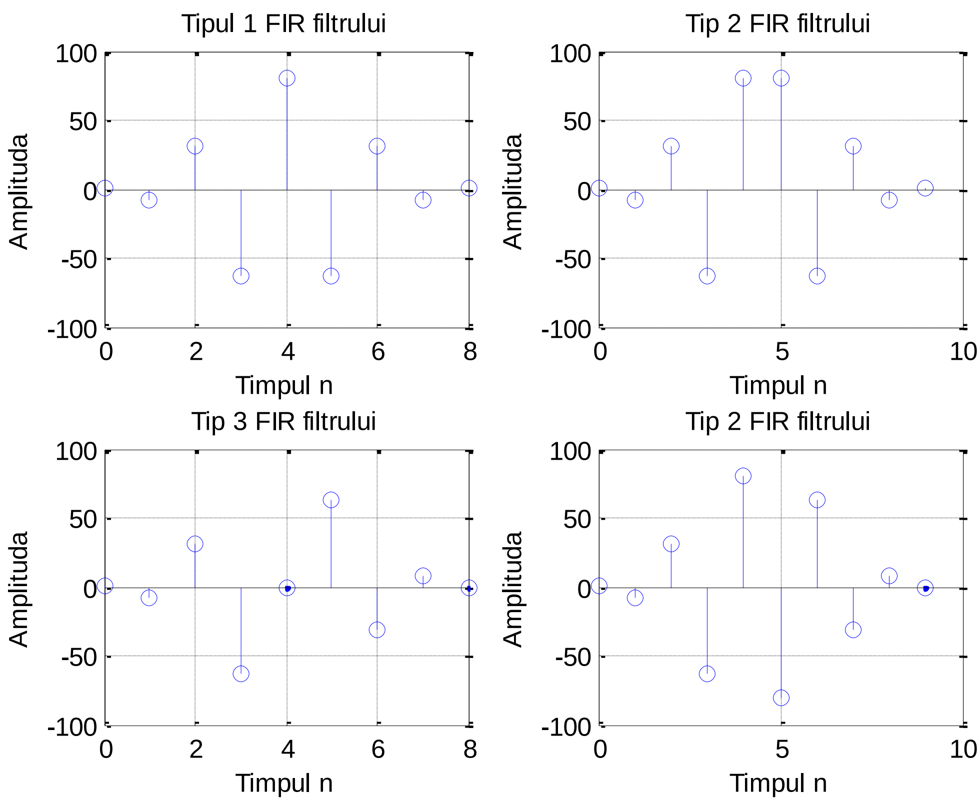
Tipul 3: Răspunsul impuls asimetric de lungime impară;

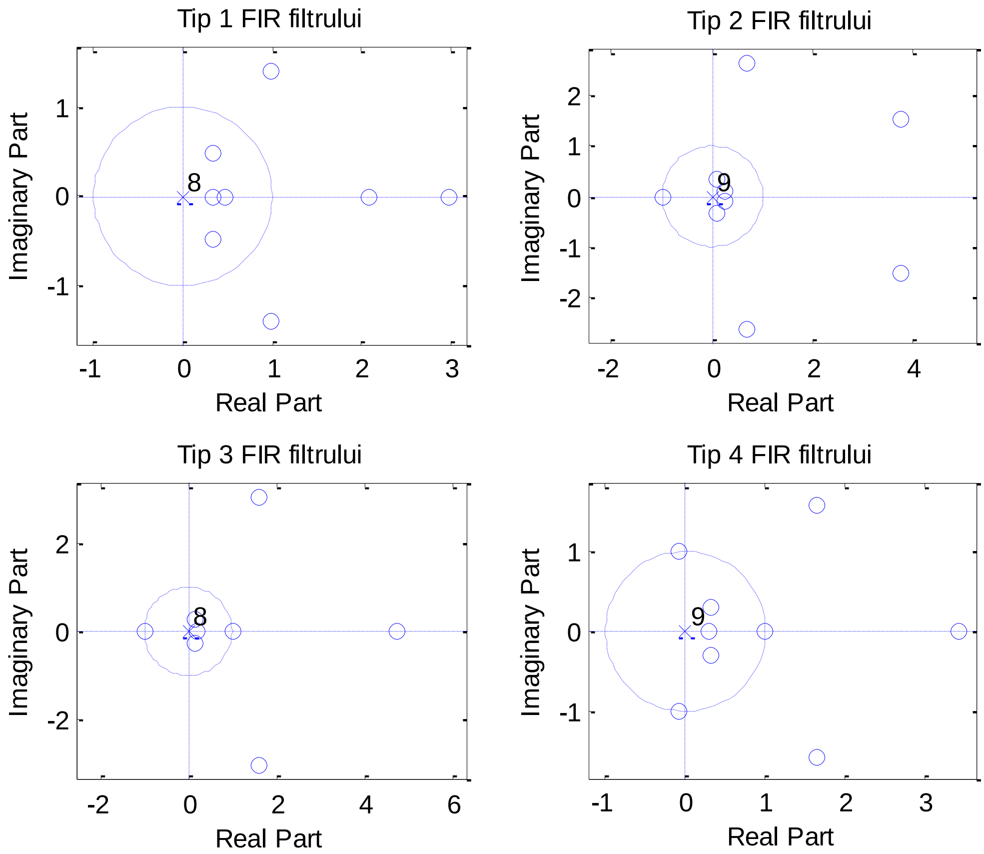
Tiplu 4: Răspunsul impuls asimetric de lungime pară.

```
% 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
```

```
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));
```

Rezultatul programului:





b = 1.0000 -8.5000 30.5000 -63.0000

Zeroul FIR filtrului de Tip 1
2.9744
2.0888
0.9790 + 1.4110i
0.9790 - 1.4110i
0.3319 + 0.4784i
0.3319 - 0.4784i
0.4787
0.3362

Zeroul FIR filtrului de Tip 2
3.7585 + 1.5147i
3.7585 - 1.5147i
0.6733 + 2.6623i
0.6733 - 2.6623i
-1.0000
0.0893 + 0.3530i
0.0893 - 0.3530i
0.2289 + 0.0922i
0.2289 - 0.0922i

Zeroul FIR filtrului de Tip 3
4.7627
1.6279 + 3.0565i
1.6279 - 3.0565i
-1.0000
1.0000
0.1357 + 0.2549i
0.1357 - 0.2549i
0.2100

Zeroul FIR filtrului de Tip 4
3.4139
1.6541 + 1.5813i
1.6541 - 1.5813i
-0.0733 + 0.9973i
-0.0733 - 0.9973i
1.0000
0.3159 + 0.3020i
0.3159 - 0.3020i
0.2929

Programul 6_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.

```
% Programul P4_4
% Test de verificare a stabilitatii
clf;
den=input('Introduceti coeficientii de numitor:');
ki=poly2rc(den);
disp('Parametrii testului de stabilitate: ');
disp(ki);
```

```
Exemplu:

Introduceti coeficientii de numitor:10

Parametrii testului de stabilitate:

>>
```

Lucrare de laborator Nr.7

Tema: Prelucrarea digitală a semnalelor continue în timp.

Sarcina: De a studia restabilirea unui semnal continuu dintr-un semnal discret dat, cu o cantitate minimă de pierderi.

Consideraţii teoretice

Algoritmii de prelucrare a semnalelor discrete sunt adesea aplicate asupra semnalelor continue în timp. Această lucrare de laborator permite studierea algoritmilor principali de transformare a semnalelor continue în semnale discrete, mai apoi, după aplicarea corespunzătoare a algoritmilor de transformare, va avea loc transformarea semnalului într-un semnal continuu echivalent lui, după posibilitate cu cantităţi minime de cheltuieli.

Restabilirea semnalului discret: sarcina – de găsit condiţiile necesare, cu ajutorul cărora semnalul poate fi restabilit conform eşantioanului discret. Înainte de toate:

$$\cos(2\pi w_0 t) \Leftrightarrow [\delta(w-w_0)+\delta(w+w_0)]/2.$$

Transformata Fourier a secvenţei

Fie avem un semnal *x(t),* şi este ales un pas de discretizare *S.* Funcţia este înlocuită printr-o secvenţă $y[n]=x(nS)$.

Definiţie. Transformata Fourier a unei secvenţe este numită funcţia

$$Y(w)=\sum_{n=-\infty}^{\infty} y[n]e^{-2\pi wiSn} \qquad \textbf{(4).}$$

Funcţia $Y(w)$ este o funcţie periodică. Deseori, pentru simplitate, vom presupune *S=1,* şi în acest caz perioada funcţiei este egală cu 1. Aceasta este diferenţa fundamentală între transformatele Fourier a funcţiilor şi a secvenţelor. În acelaşi timp, ambele transformate sunt strîns legate între ele. Fie avem

$u(t)=\sum_{n=-\infty}^{\infty}\delta(t-nS)$, atunci $Y(w)=\int_{-\infty}^{\infty}u(t)x(t)e^{-2\pi wit}dt$ **(5),**

adică, este transformata Fourier a produsului a două funcţii, dintre care una este funcţie generalizată. Conform teoriei generale, transformata Fourier a produsului a două funcţii este egală cu convoluţia de imagini ale factorilor. Pentru simplificarea notaţiilor *S=1.* Găsim $U(w)=\int_{-\infty}^{\infty}u(t)e^{-2\pi wit}dt$. Punem $U_N(w)=\int_{-\infty}^{\infty}\sum_{n=-N}^{N}\delta(t-n)e^{-2\pi iwt}dt = \sum_{n=-N}^{N}e^{-2\pi iwn}$. Atragem atenţia la faptul că, aceasta este o funcţie periodică cu perioada egală cu 1, prezentată ca suma progresiei geometrice.

Avem $U_N(w)=\dfrac{e^{-2\pi iNw}-e^{2\pi i(N+1)w}}{1-e^{2\pi iw}}$. Înmulţim numărătorul şi numitorul la $e^{-\pi iw}$.

Obţinem $U_N(w)=\dfrac{\sin(2\pi(N+1/2)w)}{\sin(\pi w)}$.

În apropierea la $\int_{-c}^{c}\varphi(w)U_N(w)dw=\int_{-c}^{c}\dfrac{\varphi(w)\sin(2\pi(N+1/2)w)}{\pi iw}dw$ se pretinde către $N\to\infty$ la $\varphi(0)$. Pentru orice *S* se poate de scris formula: $\sum\delta(t-Sn)\Leftrightarrow\sum\delta(w-\frac{n}{S})/S$ **(6).**

Relaţiile dintre transformatele Fourier de tip discret şi continue. Frecventa Nyquist.

Utilizînd formulele (5) şi (6) şi, presupunînd adevărată declaraţia despre transformarea Fourier de la produsul a două funcţii, primim relaţiile:

$$Y(w)=U(w)*X(w)$$

unde $x(t)\Leftrightarrow X(w)$,

de unde reiese:

$$Y(w) = \sum X(w - \frac{n}{S}) / S \qquad \textbf{(7)}$$

Această formulă stabileşte legătura între transformata Fourier de tip continuu şi cea de tip discret. După cum urma de aşteptat, $Y(w)$ are perioada $1/S$, ceea ce este în concordanţă cu (4).

Comenzile MATLAB utilizate

Comenzi de uz general

`Length   size`

Operatori şi simboluri speciale

`:   .   +   -   *   /   ;   %   ==   ~   &   |`

Matrici elementare şi operaţiile asupra lor

`ones   linspace   pi`

Funcţii elementare

`abs   cos   exp`

Grafică bidimensională

`axis   grid   plot   stem   title   xlabel   ylabel`

Funcţii grafice de uz general

`clf   grid   plot   stem   subplot`

Funcţii ale pachetului de prelucrare a semnalelor

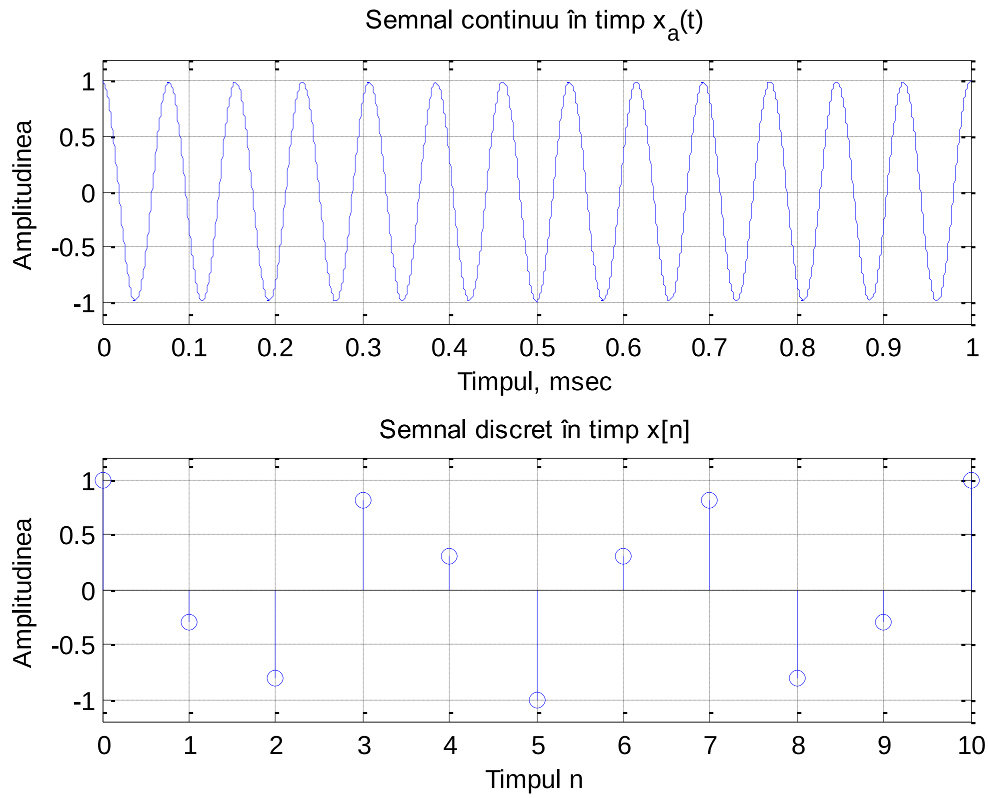
`butter   buttord   cheb1ord   cheb2ord   cheby1   cheby2   ellip`
`ellipord   freqz   sinc`

Program 7_1

Aici vom transforma condiţionat un semnal continuu într-un semnal discret corespunzător, cu redarea ambelor grafice.

```
% Program P5_1
% Ilustrarea procesului de discretizare în domeniul de timp
clf;
t=0:0.0005:1;
f=13;
xa=cos(2*pi*f*t);
subplot(2,1,1);
plot(t,xa); grid;
xlabel('Timpul, msec');
ylabel('Amplitudinea');
title('Semnal continuu în timp x_{a}(t)');
axis([0 1 -1.2 1.2]);
subplot(2,1,2);
T=0.1;
n=0:T:1;
xs=cos(2*pi*f*n);
k=0:length(n)-1;
stem(k,xs); grid;
xlabel('Timpul n');
ylabel('Amplitudinea');
title('Semnal discret în timp x[n]');
axis([0 (length(n)-1) -1.2 1.2]);
```

Rezultatul programului:

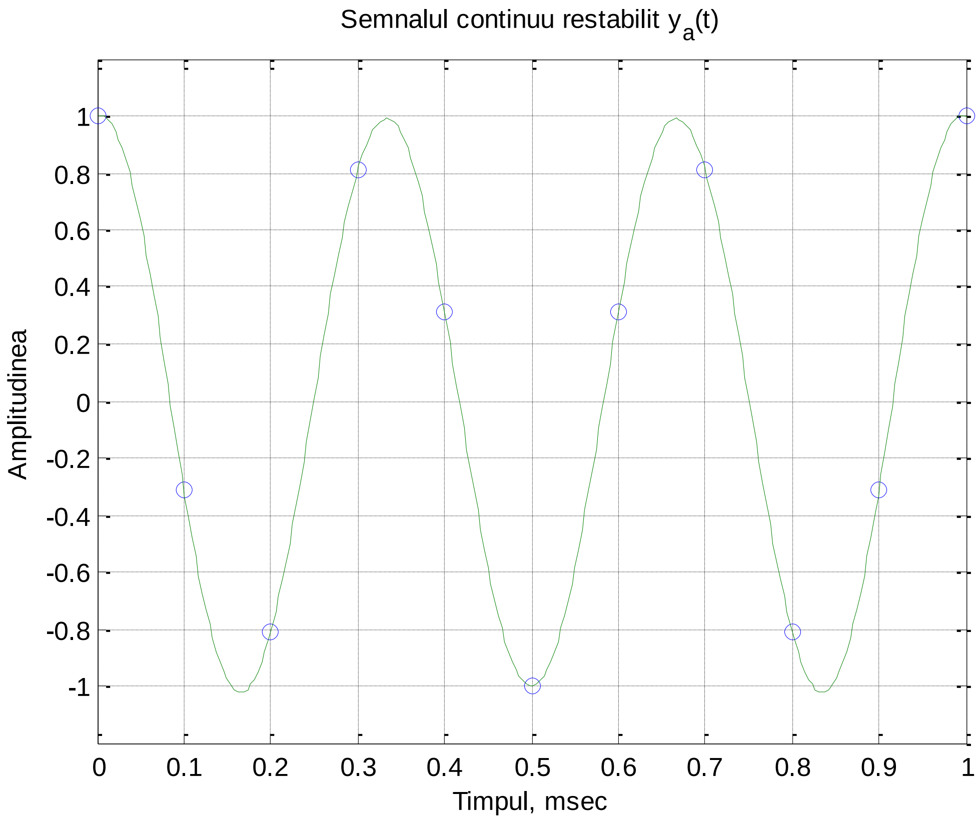


Program 7_2

Aici vom restabili semnalul continuu din semnalul discret deja existent. În ambele cazuri vom primi o secvenţă cu o valoare finită de elemente, dar secvenţa semnalului continuu rezultată va fi mai aproape de continuitate. Restabilirea are loc în timp.

```
%Program P5_2
% Demonstrarea efectului de „imaginaţie” în domeniul de timp
clf;
T=0.1; f=13;
n=(0:T:1)';
xs=cos(2*pi*f*n);
t=linspace(-0.5,1.5,500)';
ya=sinc((1/T)*t(:,ones(size(n))) - (1/T)*n(:,ones(size(t)))')*xs;
plot(n,xs,'o',t,ya);
grid;
xlabel('Timpul, msec');
ylabel('Amplitudinea');
title('Semnalul continuu restabilit y_{a}(t)');
axis([0 1 -1.2 1.2]);
```

Rezultatul programului:



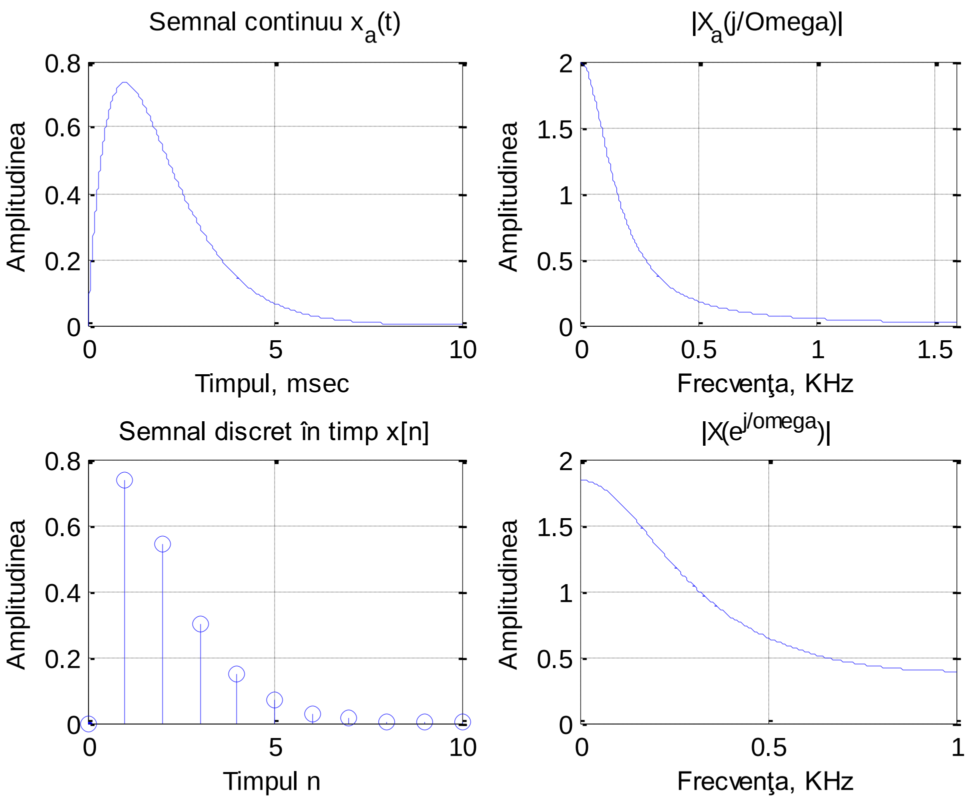
Program 7_3

Programul dat deasemenea restabileşte semnalul continuu dintr-un semnal discret existent, dar utilizează alt algoritm care lucrează cu frecenţa.

```
% Program P5_3
% Demonstrarea efectului de „imaginaţie” în domeniul de frecvenţă
clf;
t=0:0.005:10;
xa=2*t.*exp(-t);
subplot(2,2,1);
plot(t,xa); grid;
xlabel('Timpul, msec');
ylabel('Amplitudinea');
title('Semnal continuu x_{a}(t)');
subplot(2,2,2);
wa=0:10/511:10;
ha=freqs(2,[1 2 1],wa);
plot(wa/(2*pi),abs(ha)); grid;
```

```
xlabel('Frecvenţa, KHz');
ylabel('Amplitudinea');
title('|X_{a}(j/Omega)|');
axis([0 5/pi 0 2]);
subplot(2,2,3);
T=1;
n=0:T:10;xs=2*n.*exp(-n);
k=0:length(n)-1;
stem(k,xs); grid;
xlabel('Timpul n');
ylabel('Amplitudinea');
title('Semnal discret în timp x[n]');
subplot(2,2,4);
wd=0:pi/255:pi;
hd=freqz(xs,1,wd);
plot(wd/(T*pi),T*abs(hd)); grid;
xlabel('Frecvenţa, KHz');
ylabel('Amplitudinea');
title('|X(e^{j/omega})|');
axis([0 1/T 0 2]);
```

Rezultatul programului:

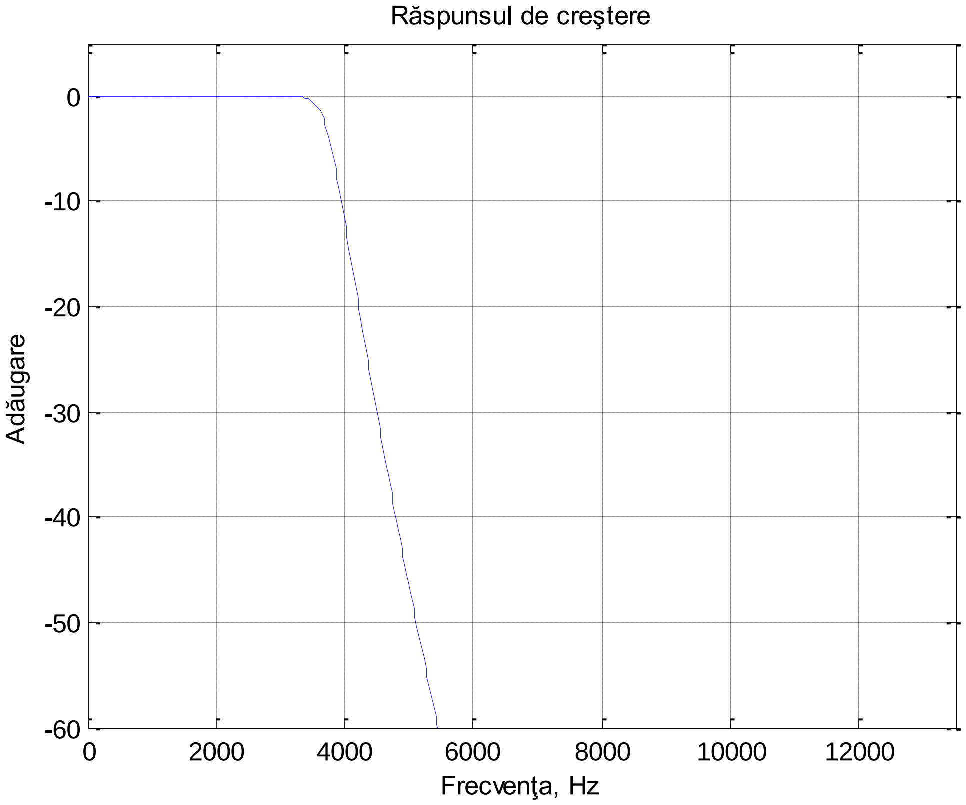


Program 7_4
Aici se demostrează lucrul filtrului analogic, utilizat la restabilirea semnalului continuu.

```
% Program P5_4
% Proiectarea filtrului analogic
clf;
Fp=3500; Fs=4500;
Wp=2*pi*Fp; Ws=2*pi*Fs;
[N,Wn]=buttord(Wp,Ws,0.5,30,'s');
[b,a]=butter(N,Wn,'s');
wa=0:(3*Ws)/511:3*Ws;
h=freqs(b,a,wa);
```

```
plot(wa/(2*pi),20*log10(abs(h))); grid;
xlabel('Frecvenţa, Hz');
ylabel('Adăugare');
title('Răspunsul de creştere');
axis([0 3*Fs -60 5]);
```

Rezultatul programului:



Lucrare de laborator Nr.8

Tema: Structura filtrelor numerice

Scopul lucrării: De a învăţa cum se construiesc diferite structuri a filtrelor în dependenţă de proprietăţile dorite a viitoarelor structuri.

Noţiuni generale

În această lucrare de laborator se analizează diferite structuri de filtre (paralelă şi cascadă). Structura definită a filtrelor permite căpătarea relaţiilor corespunzătoare (convenabile) între variabilele interne, şi, în consecinţă, semnalul de ieşire corespunzător.

Orice cauzalitate IIR(Infinite Impulse Response – filtru cu caracteristică infinită a impulsului) filtrul se caracterizează prin funcţia de transformare:

$$H(z) = \frac{\sum_{k=0}^{N} p_k z^{-k}}{1 + \sum_{k=1}^{N} d_k z^{-k}}$$

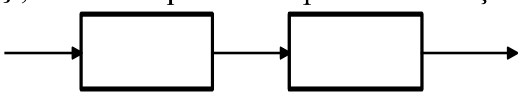
Filtrele numerice sunt cazuri particulare ale sistemelor liniare invariante. Limitarea semnificativă datorată realizării fizice a sistemului.

Definiţie: Sistema este numită **fizic realizabilă,** dacă semnalul la ieşire în momentul de timp t depinde de semnalul de intrare în momentul de timp $\leq t$.

Fie un SLI (sistem liniar invariant) *T.* Să analizăm întrun punct al secvenţei $\Delta : \Delta[0] = 1, \Delta[t] = 0, t \neq 0$.

Fie $T\{\Delta\} = \{h[n]\}$, şi prin definiţie $\Delta_k[t] = \Delta[t-k]$. Pentru secvenţa arbitrară $\{x[n]\}$ este validă extinderea $\{x[n]\} = \sum_k x[k]\Delta_k$. De liniaritate $T\{x[n]\} = \sum_k x[k]T\{\Delta_k\}$, iar de invariantităţii $T\{\Delta_k\} = \{h[n-k]\}$. În cele din urmă, în cazul în care $\{y[n]\} = T\{x[n]\}$, atunci $y[n] = \sum_k x[k]h[n-k]$ (9).

Cu alte cuvinte, reacţia la orice secvenţă se obţine prin convoluţia acestei secvenţe şi secvenţa $\{h[n]\}$, numită impuls de raspuns sau funcţia de răspuns.



Comenzile MATLAB utilizate:

Comenzi de uz general

```
Disp length
```

Operatori şi simboluri speciale

```
:    .    +    -    *    /    ;    %
```

Matrici elementare şi operaţiile asupra lor

```
ones     pi
```

Funcţii pachetului de prelucrare a semnalelor

```
residue latc2tf poly2rc  residuez  tf2latc  zp2sos
```

Programul 8_1

Cu ajutorul funcţiei din MATLAB zp2sos programul calculează coeficienţii *H(z)* apelînd funcţia de tranziţie.

```
% Programul P6_1
% Tratarea funcţiei de tranziţie în forma coeficienti
```

```
num=input('Vectorul coeficientilor numaratorului:');
den=input('Vectorul coeficientilor numitorului: ');
[z,p,k]=tf2zp(num,den);  sos=zp2sos(z,p,k);
```

Exemplu:

```
Vectorul coeficientilor numaratorului:[10,20,30]

Vectorul coeficientilor numitorului: [12,34,45]
```

sos = 0.8333 1.6667 2.5000 1.0000 2.8333 3.7500

Programul 8_2

Există două forme paralele de realizare funcţiei de conversie cauzale IIR. Prima forma se bazează pe porţiuni de spaţiu dat, în funcţie de z^{-1}. Ea se realizează cu ajutorul funcţiei MATLAB residuez. Iar forma a doua depinde de *z* şi se realizează cu ajutorul funcţiei MATLAB residue.

```
% Programul P6_2
% Realizarea in forma paralela a functiei de tranzitie IIR
num=input('Vectorul coeficientilor numaratorului ');
den=input('Vectorul coeficientiloe numitorului ');
[r1,p1,k1]=residuez(num,den);
[r2,p2,k2]=residue(num,den);
disp('Forma paralela 1')
disp('Ramasite: '); disp(r1);
disp('Polii: '); disp(p1);
disp('Constanta: '); disp(k1);
disp('Forma paralela 2')
disp('Ramasite: '); disp(r2);
disp('Polii: '); disp(p2);
disp('Constanta: '); disp(k2);
```

Rezultatul rularii programului:

Vectorul coeficientilor numaratorului [10,20,30]

Vectorul coeficientiloe numitorului [12,34,45]

Forma paralela 1
Ramasite:
0.0833 + 0.1736i
0.0833 - 0.1736i

Polii:
-1.4167 + 1.3202i
-1.4167 - 1.3202i

Constanta:
0.6667

Forma paralela 2
Ramasite:
-0.3472 - 0.1359i
-0.3472 + 0.1359i

Polii:
-1.4167 + 1.3202i
-1.4167 - 1.3202i

Constanta:
0.8333

Programul 8_3

În acest program se realizează algoritmul Gray-Markel. Realizarea grilaj cascadă Gray-Markel funcţiei de taranziţie H(z) de ordinul – N, se bazează pe realizarea grilaj cascadă funcţiei intermediare de tranzacţie $A_N(z)$, avînd acelaşi numitor, ca şi funcţia H(z).

```
% Programul P6_3
% Structura grilaj cascada Gray-Markel
% k - vectorul parametrilor grilajului
% alpha - vectorul mutiplicatorilor de legatura directa
format long
% introducerea coeficientilor
num=input('Vectorul coeficientilor numaratorului: ');
den=input('Vectorul coeficientilor numitorului: ');
```

```
N=length(den)-1; % Gradul polinomului de la numitor
k=ones(1,N);
a1=den/den(1);
alpha=num(N+1:-1:1)/den(1);
for ii=N:-1:1,
    alpha(N+2-ii:N+1)=alpha(N+2-ii:N+1)-alpha(N-ii+1)*a1(2:ii+1);
    k(ii)=a1(ii+1);
    a1(1:ii+1)=(a1(1:ii+1)-k(ii)*a1(ii+1:-1:1))/(1-k(ii)*k(ii));
end
disp('Parametrii grilajului: '); disp(k)
disp('Multiplicatorii legaturii directe: '); disp(alpha)
subplot(1,2,1); title('Parametrii grilajului');
bar(k);
subplot(1,2,2); title('Multiplicatorii legaturii directe');
bar(alpha);
```

Rezultatul rulării programului:

Vectorul coeficientilor numaratorului: [10,20,30]

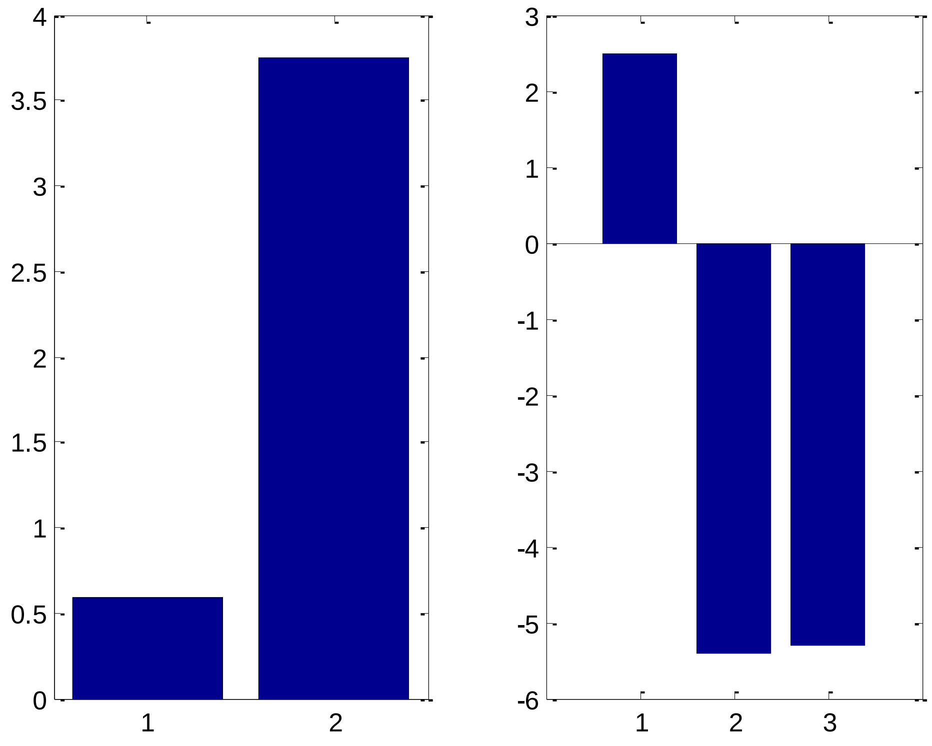
Vectorul coeficientilor numitorului: [12,34,45]

Parametrii grilajului:

0.596491228070175   3.750000000000000

Multiplicatorii legaturii directe:

2.500000000000000 -5.416666666666667 -5.310672514619883



Bibliografia

1. Сергиенко А. Б. **,** *Цифровая обработка сигналов*, изд. Питер, 2003

2. S.Haykin, B.V. Veen, *Signals and Sistems*, New Yorc, 1999

3. D.M.Etter, *Engineering Problem Solving with MATLAB*, Matlab Curriculum Series, 1996