Versiune revãzutã la 20 februarie 2004

GHEORGHE M.PANAITESCU

PROCESAREA NUMERICÃ A SEMNALELOR

Ghid de lucrãri

Universitatea "Petrol-Gaze" Ploiesti
2004

Prezentul **Ghid** vizează aplicatiile la cursul de **Procesarea numerică a semnalelor** din programul de pregãtire al anului III la specializãrile Automaticã si informaticã industrialã si Calculatoare. El contine zece lucrãri de dificultãti variate. Unele mai simple se intind pe durata unei singure sedinte, altele mai complexe pot dura douã sau trei sedinte consecutive rezervate lucrãrilor aplicative.
La una din lucrari se poate cere, în completare sau ca alternativã la un numãr de probleme legate de cunostintele predate, un **Referat tehnic** care vrea sã fie o simulare a unor rapoarte explicative relative la lucrãri pe care in calitate de inginer, viitorul absolvent le va avea de realizat întru satisfacerea unor comenzi ale unor potentiali solicitanti.
Se sugereaza imediat o structura posibilã a unui asemena **Referat**.

Ghid de redactare a Referatelor tehnice

Referatele tehnice asupra unor lucrãri efectuate în orele de aplicatii au ca prim scop o verificare partialã a cunostintelor.
Al doilea scop urmãrit, care nu este mai putin important constã în o testare a capacitãtii autorului-student de a aseza în paginã, în stil **concis** si **clar**, **enuntul**, **scopul** si **solutia** problemei care constituie obiectul lucrãrii. Este, sub acest aspect, o simularea a cazurilor în care un inginer este pus în situatia de a elabora un raport scris asupra lucrãrilor sale. Punctualitatea depunerii raportului este, de asemenea, apreciatã.
Cu aceste precizãri, referatele trebuie sã continã:

- formularea obiectivului lucrãrii;
- un minim suport teoretic cu eventuale trimiteri bibliografice;
- în ce constã solutia si mijloacele prin care a fost obtinutã;
- rãspunsuri sustinute de explicatii la chestiunile ridicate în ghidul scris al lucrãrii;
- concluzii.

În plus:

- Raportul trebuie sã se întindã pe cca. 5-6 pagini;
- Raportul trebuie sã fie individual; rapoartele identice, diferite doar prin semnãturã vor fi apreciate negativ;
- Raportul se predã exact la data indicatã verbal de conducãtorul lucrãrilor.

Lucrarea selectatã pentru **Referat**, termenul de predare si alte posibile detalii se stabilesc în una din primele sedinte de aplicatii.

C U P R I N S U L   G H I D U L U I

LUCRAREA ZERO

ELEMENTE DE UTILIZARE
A MEDIULUI DE PROGRAMARE *Matlab*®

1. Obiectivele lucrãrii

- Initierea în utilizarea mediului ***Matlab***
- Operatii în ***Matlab***, functii si *script*-uri ***Matlab***

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Manuale de programare în ***Matlab***, documentarea *on-line (Help)*

3. Breviar teoretic

Matlab este un limbaj de programare foarte performant orientat pe calcule inginerestí. Limbajul ***Matlab*** pune laolaltã posibilitãti de calcul, de vizualizare si de programare într-un mediu foarte accesibil în care problemele si solutiile sunt exprimate într-o manierã matematicã familiarã. Utilizãri tipice:

- Calcule matematice
- Dezvoltarea de algoritmi
- Modelare, simulare, dezvoltare de modele originale
- Analiza datelor experimentale, explorarea si vizualizarea datelor
- Graficã stiintificã si inginereascã
- Dezvoltarea de aplicatii oricât de complicate, inclusiv de aplicatii cu interfete grafice

Matlab este un sistem interactiv pentru care elementul de bazã cu care se opereazã este matricea. Matricile nu necesitã dimensionare prealabilã ceea ce reprezintã o facilitate de programare extraordinarã.
Numele de ***Matlab*** vine de la *matrix laboratory*.
Matlab-ul a evoluat de-a lungul anilor prin multe contributii ale utilizatorilor. În mediul universitar este mijlocul standard de instruire practicã la multe discipline introductive sau de aprofundare din domeniul matematicii, ingineriei, stiintelor. În industrie este utilizat pentru cercetãri variate, pentru dezvoltãri de tehnologii si pentru analize.
Mediul ***Matlab*** contine câteva pachete (*toolboxes*) specifice unor domenii ale stiintei si ingineriei. Aceste pachete permit învãtarea si aplicarea unor tehnici specializate de calcul. Pachetele sunt colectii de functii ***Matlab*** (fisiere *M*) particulare unei clase de probleme. În legãturã cu disciplina *Procesarea numericã a semnalelor* existã cel putin un pachet specific: *Signal Processing*

Toolbox. În afarã de acesta vor mai fi folosite, se-ntelege, si alte pachete complementare, încã mai specializate.

4. Modul de lucru

Pentru a accede la mediul de programare ***Matlab*** este necesar a face cu *mouse*-ul un dublu *click* pe icoana asociatã cu ***Matlab***-ul. Iesirea se poate face prin *click* pe coltul din dreapta sus al ferestrei de lucru sau prin introducerea la promptul Matlab a cuvântului *quit*. Pentru a nu încãrca excesiv spatiul de lucru atribuit prin *default*, la aparitia ferestrei ***Matlab*** este recomandatã trecerea imediatã în spatiul de lucru definit de utilizator printr-o comanda

`cd [cale]`

cu `[cale]` un sir de genul **`c:\aplicatii\1911a`**

Cu comanda

`help matlab/ops`

se obtine o listã completã a operatorilor utilizati de matlab. În general, cu comanda **`help`** urmatã de numele unei functii se obtin informatii referitoare la argumentele acelei functii si la rezultatele produse.

Calea de a obtine informatii de interes este formularea comenzii **`lookfor`** urmatã de un cuvânt caracteristic subiectului urmãrit.

Cele douã comenzi, ca si multe altele pot aduce pe ecran informatii care depãsesc spatiul ferestrei curente. Lectura integralã a spatiului se poate face uzând de tastele **PgUp** si **PgDn** sau de cursorul lateral al ferestrei. Tastele cu sãgeti sunt rezervate pentru regãsirea (si eventual reactualizarea/reactivarea) unor comenzi din sirul de comenzi deja introduse.

Initializarea unor variabile se face prin comenzi de genul

```
a=2;
b1=[1 3 2 1];
c=[1 4 3 2;1 1 1 1];
da1=0:0.1:2;
```

etc. Caracterul `;` opreste afisarea rezultatului obtinut. Afisarea se produce automat dacã acest caracter lipseste sau dacã variabila este mentionatã la promptul de pe ecran prin numele ei. Se fac operatii cu operanzi initializati în maniera arãtatã.

Orice comandã devine activã la actionarea tastei **Enter**.

Cu comenzile **`help function`** si **`help script`** se pot aduce pe ecran informatii relativ la posibilitãtile scrierii de cãtre utilizator a unor functii si a unor secvente de program (*script*-uri). Se recomandã scrierea unor functii/*script*-uri simple si activarea lor.

Cu comanda **`help plot`** se obtin informatii referitoare la realizarea graficelor. Se realizeazã grafice simple. Se studiazã si se experimenteazã parametrizarea lor.

5. Chestiuni de studiu

- Initializarea unor variabile, executarea de operatii cu variabilele initializate
- Scrierea si utilizarea unor functii simple
- Scrierea si utilizarea unor *script*-uri simple
- Reprezentãri grafice simple, parametrizarea graficelor

Observatie: Lucrarea prezentã se executã pe durata primelor douã sedinte de aplcatii

LUCRAREA I

SEMNALE, SERII FOURIER, TRANSFORMAREA FOURIER

1. Obiectivele lucrãrii

- Descompunerea Fourier a semnalelor periodice si aperiodice
- Studiul spectrelor unor semnale periodice si aperiodice

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Semnale periodice sunt acele semnale pentru care

$$s(t) = s(t + T_0) \quad (\forall)\, t \in R$$

cu T_0 o constantã numitã perioadã.
Semnalelor periodice le este asociatã o serie Fourier

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t)$$

cu coeficientii calculati cu relatiile

$$a_n = \frac{2}{T_0} \int_{\alpha}^{\alpha+T_0} s(t) \cos n\omega_0 t \qquad b_n = \frac{2}{T_0} \int_{\alpha}^{\alpha+T_0} s(t) \sin n\omega_0 t$$

Frecventa unghiularã (pulsatia) este legatã de perioadã prin relatia $\omega_0 = 2\pi/T_0$. Seria Fourier este convergentã aproape pretutindeni la valorile functiei dacã sunt îndeplinite conditiile Dirichlet. În punctele de discontinuitate, care nu pot fi decât de prima specie, seria converge cãtre media aritmeticã a celor douã limite laterale ale functiei-semnal $s(t)$.
O formã alternativã a seriei de mai sus este

$$s(t) = \sum_{n=-\infty}^{+\infty} C(n\omega_0) e^{jn\omega_0 t}$$

cu coeficientii $C_n = C(n\omega_0) = (a_n \pm jb_n)/2$, cu alegerea semnului dupã regula: plus pentru $n < 0$, minus pentru $n > 0$. Coeficientii se pot calcula si direct cu relatia

$$C_n = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{+\frac{T_0}{2}} s(t) e^{-jn\omega_0 t} dt$$

Trecerea la limitã, $T_0 \to \infty$ aduce relatia ultimã la forma

$$S(\omega) = \int_{-\infty}^{+\infty} s(t) e^{-j\omega t} dt$$

cunoscutã ca *transformata Fourier* directã a semnalului *s*(*t*), de data aceasta aperiodic. Integrala este convergentã dacã semnalul satisface conditiile Dirichlet pe orice interval finit. Existã si transformarea inversã, analogã seriilor Fourier pentru semnalele periodice,

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(\omega) e^{j\omega t} d\omega$$

care reconstituie semnalul în domeniul timp, *s*(*t*), din exprimarea lui absolut echivalentã informational din domeniul frecventelor, $S(\omega)$.
Atât pentru semnalele periodice cât si pentru cele aperiodice prin tratarea Fourier se obtin spectre ale semnalelor, care sunt proiectii ale acestor semnale pe baze în spatiul semnalelor alcãtuite din sinusoide.

4. Modul de lucru

Pentru întelegerea dezvoltãrii Fourier a semnalelor periodice se recomandã executarea mai întâi a functiei ***Matlab*** **XFOURIER** care este un *demo* aplicatie la cazul unui semnal rectangular periodic, fãrã componentã continuã, aranjat pe axa timpului pentru ca semnalul sã fie functie impară. Acest *demo* se activeazã cu numele lui scris cu caractere *lower-case* la promptul ***Matlab*** si se urmãreste pas cu pas, cu lectura atentã a comentariilor care apar pe monitorul calculatorului.
Pentru câteva semnale aperiodice au fost pregãtite câteva *script*-uri ***Matlab***: **FRECT**, **FTRIUNGHI**, **FSINUS**, **FCOSINUS**, **FSINUSFI**, **FTRENSIN,** date în anexã. Executarea *script*-urilor se obtine prin apelarea lor pe nume (*lower-case*) la promptul ***Matlab***. În prealabil se copiazã fisierele respective care trebuie sã aibã extensia **.m** în directorul/*folder*-ul de lucru si se face trecerea platformei ***Matlab*** în acest director (comanda `cd` <director de lucru>). Aceastã trecere este recomandatã pentru a face posibilã modificarea *script*-urilor dupã vointa studentului.
Se recomandã scrierea, dupã modelele oferite sau pe baza lecturilor proprii, a altor *script*-uri pentru semnale diferite de cele propuse.

5. Chestiuni de studiu

- Pentru semnalul periodic rectangular observati apropierea semnalului recompus dintr-un numãr finit de componente ale sale, pe mãsurã ce numãrul acestora creste. Este de observat, de asemenea, efectul Gibbs care constã în faldurile care afecteazã aproximãrile semnalului originar prin sume partiale ale seriei Fourier. Se constatã imposibilitatea practicã de a realiza vreodatã semnalul teoretic dat.

- Calculati puterea semnalului rectangular periodic si puterea rãmasã înafara reprezentãrilor trunchiate.
- Rezultatul transformãrii Fourier este în general o functie complexã. Se vor urmãri componentele reale si imaginare ale transformatelor. Se sugereazã reprezentarea graficã a modulelor si fazelor acestor transformate. Se va încerca reprezentarea spectrelor de puteri ale semnalelor studiate.
- Se recomandã încercarea unui comentariu la graficele obtinute la executarea programului *script* **FTRENSIN**.

FRECT

```
% Scriptul FRECT calculeaza transformata Fourier a
semnalului rectangular, simetric fata
% de origine, de durata a si de arie unitara
syms x w % sunt declarate variabilele simbolice
wmax=50;
int(1.0/a*exp(-j*w*x),-a/2,a/2); % se calculeaza integrala
Fourier
ezplot(ans,[-wmax wmax]) % se reprezinta grafic
hold on
u=-wmax:wmax:wmax;
y=0.0*u;
plot(u,y) % se traseaza orizontala y=0
hold off % Scriptul FRECT calculeaza transformata Fourier
a semnalului rectangular, simetric fata
% de origine, de durata a si de arie unitara
syms x w % sunt declarate variabilele simbolice
wmax=50;
int(1.0/a*exp(-j*w*x),-a/2,a/2); % se calculeaza integrala
Fourier
ezplot(ans,[-wmax wmax]) % se reprezinta grafic
hold on
u=-wmax:wmax:wmax;
y=0.0*u;
plot(u,y) % se traseaza orizontala y=0
hold off
```

FTRIUNGHI

```
% Scriptul FTRIUNGHI calculeaza transformata Fourier a
semnalului triunghiular, simetric fata
% de origine, de durata a si de pante b si -b
syms x w f f1 f2 % sunt declarate variabilele simbolice
wmax=50;
f1=int(b*(x+a/2)*exp(-j*w*x),-a/2,0); % se calculeaza
integrala Fourier pentru jumatatea stanga
f2=int(b*(-x+a/2)*exp(-j*w*x),0,a/2); % se calculeaza
integrala Fourier pentru jumatatea dreapta
```

```
z=strcat(char(f1),char(f2)); % se concateneaza cele doua
expresii f1 si f2
if strncmp(char(f2),'-',1)
   z=strcat(char(f1),char(f2));
else
   z=strcat(char(f1),'+',char(f2));
end
f=sym(z); % se revine la simbolic f='z'
ezplot(f,[-wmax wmax]) % se reprezinta grafic
hold on
u=-wmax:wmax:wmax;
y=0.0*u;
plot(u,y) % se traseaza orizontala y=0
hold off
```

FSINUS

```
% Scriptul FSINUS calculeaza transformata Fourier a unui
puls sinusoidal de durata unei
% perioade, centrat pe origine, de durata a si de arie
totala unitara
syms x w % sunt declarate variabilele simbolice
int(pi/(4*a)*sin(2*pi*x/a)*exp(-j*w*x),-a/2,a/2); % se
calculeaza integrala Fourier
ezplot(real(ans),[-5 5]) % se reprezinta grafic partea
reala a transformatei Fourier
hold on
ezplot(imag(ans),[-5 5]) % se reprezinta grafic partea
imaginara a transformatei Fourier
hold on
u=-5:1:5;
y=0.0*u;
plot(u,y) % se traseaza orizontala y=0
hold off
```

FCOSINUS

```
% Scriptul FCOSINUS calculeaza transformata Fourier a unui
puls sinusoidal, simetric fata
% de origine, de durata a si de arie unitara
syms x w % sunt declarate variabilele simbolice
int(pi/(2*a)*cos(pi*x/a)*exp(-j*w*x),-a/2,a/2);    %    se
calculeaza integrala Fourier
ezplot(ans,[-5 5]) % se reprezinta grafic
hold on
u=-5:1:5;
y=0.0*u;
plot(u,y) % se traseaza orizontala y=0
hold off
```

FSINUSFI

```
% Scriptul FSINUSFI calculeaza transformata Fourier a unui
puls sinusoidal de durata unei
% perioade, deplasat fata de origine cu b, de durata a si
de arie totala unitara
syms x w % sunt declarate variabilele simbolice
int(pi/(4*a)*sin(2*pi*(x-b)/a)*exp(-j*w*x),-a/2-b,a/2-b);
% se calculeaza integrala Fourier
ezplot(real(ans),[-5 5]) % se reprezinta grafic partea
reala a transformatei Fourier
hold on
ezplot(imag(ans),[-5 5]) % se reprezinta grafic partea
imaginara a transformatei Fourier
hold on
u=-5:1:5;
y=0.0*u;
plot(u,y) % se traseaza orizontala y=0
hold off
```

FTRENSIN

```
% Scriptul FTRENSIN calculeaza transformatele Fourier ale
unei scuccesiuni de sinusoide
% de frecvente diferite, de durate a/2 si juxtapuse si a
sumei a doua sinusoide de frecvente
% egale cu cele dintai dar de amplitudine pe jumatate si
existente pe durata a
u=-300:100:300;
y=0.0*u;
syms x w f f1 f2 % sunt declarate variabilele simbolice
% Se calculeaza integrala Fourier pentru prima sinusoida
f1=int(sin(70*pi*x/a)*exp(-j*w*x),-a/2,0);
% Se reprezinta grafic modulul transformatei Fourier a
primei sinusoide
ezplot(abs(f1),[-300 300],1)
hold on
plot(u,y) % se traseaza orizontala y=0
hold off
% Se calculeaza integrala Fourier pentru a doua sinusoida
f2=int(sin(35*pi*x/a)*exp(-j*w*x),0,a/2);
% Se reprezinta grafic modulul transformatei Fourier a
celei de a doua sinusoide
ezplot(abs(f2),[-300 300],2)
hold on
plot(u,y)
hold off
```

```
% Se sintetizeaza transformata semnalului egal cu cele
doua sinusoide juxtapuse
if strncmp(char(f2),'-',1)
   z=strcat(char(f1),char(f2));
else
   z=strcat(char(f1),'+',char(f2));
end
f=sym(z);
% Se reprezinta grafic modulul transformatei Fourier a
celor doua sinusoide succesive
ezplot(abs(f),[-300 300],3)
hold on
plot(u,y)
hold off
% Se calculeaza integrala Fourier pentru suma a doua
sinusoide de amplitudine pe jumatate
f=int(0.5*sin(70*pi*x/a)*exp(-
j*w*x)+0.5*sin(35*pi*x/a)*exp(-j*w*x),-a/2,a/2);
% Se reprezinta grafic modulul transformatei Fourier a
sumei celor doua sinusoide
ezplot(abs(f),[-300 300],4)
hold on
plot(u,y)
hold off
```

LUCRAREA II

INTERPOLAREA SEMNALELOR ÎN DOMENIUL TIMP (I)

1. Obiectivele lucrãrii

- Esantionarea semnalelor de bandã limitatã
- Reconstituirea semnalelor din esantioanele lor.

2. Aparaturã si suport documentar

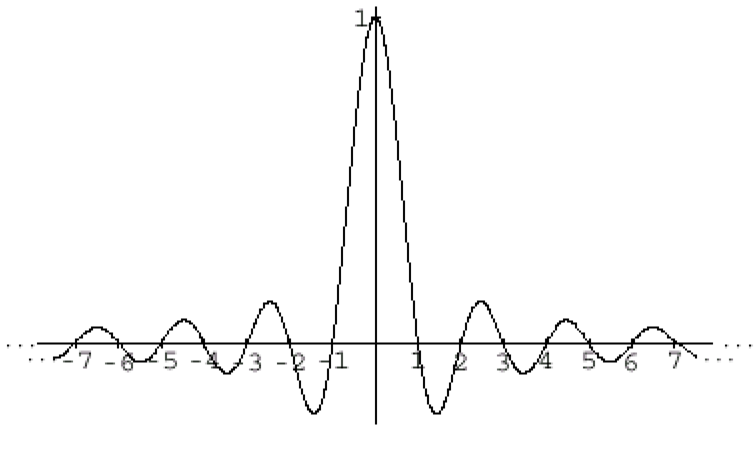
- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Teorema de esantionare a lui Shannon afirmã cã un semnal de bandã limitatã superior de frecventa (unghiularã) Ω poate fi deplin reconstituit dacã sunt date esantioanele sale prelevate cu regularitate, mai frecvent decât dublul frecventei maxime Ω din spectrul semnalului. Perioada esantioanelor trebuie sã fie, asadar, mai micã decât π/Ω. Dacã frecventa de esantionare este $2W$, frecventa pe jumãtate W este cunoscutã ca frecventa Nyquist. Reconstituirea semnalului $s(t)$ se realizeazã cu relatia de interpolare

$$s(t) = \sum_{n=-\infty}^{+\infty} s\left(\frac{n}{2W}\right) \frac{\sin 2\pi W\left(t - \frac{n}{2W}\right)}{2\pi W\left(t - \frac{n}{2W}\right)}$$

în care se disting esantioanele $s\left(\frac{n}{2W}\right)$, $n \in Z$ si, pentru variate translatii cu multipli întregi ai perioadei de esantionare, $\frac{n}{2W}$, functia specialã cunoscutã în literaturã ca functia esantion o variantã a functiei *sinc* definitã ca



$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

cu argumentul $x = 2Wt - n$.

4. Modul de lucru

Obiectivul lucrãrii este perceperea corectã a problemelor esantionãrii semnalelor de bandã limitatã si a reconstituirii semnalelor din aceastã clasã, din esantioanele lor. Se propune în *script*-ul următor esantionarea si apoi reconstituirea unui semnal, sumã a douã sinusoide

$$s(t) = \cos(2\pi t) + \cos(4\pi t - \varphi)$$

care este, evident, de bandã limitatã. Forma semnalului se poate modifica prin valorile fazei initiale φ. Fazele lucrãrii într-o versiune minimalã sunt urmãtoarele:

- Introducerea în calculator, cu editorul ***Matlab***, a *script*-ului care urmeazã.

```
%Conditii tipice de executare a "script-ului"
%
%tmax=5        - timpul maxim de reprezentare
%fi=pi/2       - faza componentei secundare
%pasmic=0.001  - pasul utilizat la reprezentarea
grafica
%fes=7.0       - frecventa de esantionare
%rep=0         - optiune de reprezentare (rep=0) grafic
unic, (rep>0) grafice separate
tmax=5;
fi=pi/2;
pasmic=0.001;
fes=7.0;
rep=0;
tes=1/fes;     % perioada esantioanelor
t=0:pasmic:tmax;
y=cos(2*pi*t)+cos(4*pi*t-fi); % pregatirea graficului 1
(semnal)
t1=0:tes:tmax;
n=round(tmax/tes)+1;
y1=cos(2*pi*t1)+cos(4*pi*t1-fi); % esantionarea
y2=y1(1)*sinc(t/tes);     % pregatirea graficului 2
(reconstituire)
for k=1:(n-1)
   y2=y2+y1(k+1)*sinc(t/tes-k);
end
if rep>0
   subplot(3,1,1)
   plot(t,y)           % trasarea graficului 1 (semnal)
   hold on
   plot([0 tmax],[0 0])
   for i=1:n
```

```
      plot([t1(i) t1(i)],[0 y1(i)],'b:') % trasarea
graficului 2 (reconstituire)
    end
    ylabel('Original')
    title('ESANTIONAREA SEMNALELOR')
    subplot(3,1,2)
    plot(t,y2,'k')
    hold on
    plot([0 tmax],[0 0])
    for i=1:n
      plot([t1(i) t1(i)],[0 y1(i)],'b:')
    end
    ylabel('Reconstituire')
    subplot(3,1,3)
    plot(t,y-y2,'r')          % trasarea graficului 3
(diferente)
    hold on
    plot([0 tmax],[0 0])
    ylabel('Diferente')
    xlabel('Timp(s)')
else
    plot(t,y)           % trasarea graficului 1 (semnal)
    hold on
    plot([0 tmax],[0 0])
    for i=1:n
      plot([t1(i) t1(i)],[0 y1(i)],'b:') % trasarea
graficului 2 (reconstituire)
    end
    plot(t,y2,'k')
    plot(t,y-y2,'r') % trasarea graficului 3 (diferente)
    plot([0 tmax],[0 0])
    title('ESANTIONAREA SEMNALELOR')
    ylabel('Semnal(albastru)/Reconstituire(negru)/Diferen
    te(rosu)')
    xlabel('Timp(s)')
end
```

- Executarea calculelor cu valorile recomandate în antetul *script*-ului, reprezentarea semnalelor în forma originarã si dupã reconstituire precum si a diferentelor dintre ele, în spatii grafice diferite sau în acelasi spatiu
- Executarea de calcule cu valori diferite pentru frecventa de esantionare, pentru faza φ, eventual pentru amplitudini diferite ale sinusoidelor componente

5. Chestiuni de studiu

- Se apreciazã vizual si/sau cantitativ modificãrile semnalului reconstituit, calitatea reconstituirii semnalului în zona de început si de final a intervalului de timp observat
- Studentii sunt îndemnati a studia si alte semnale de bandã limitatã, la liberã alegere. O sugestie usor de implementat: un semnal asemãnãtor

celui prezentat mai sus dar cu sinusoidele componente de frecvente al cãrui raport nu este un numãr rational

LUCRAREA III

INTERPOLAREA SEMNALELOR ÎN DOMENIUL TIMP (II)

1. Obiectivele lucrãrii

- Esantionarea semnalelor
- Evaluarea aproximativã (interpolarea) semnalelor pe baza esantioanelor lor.

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

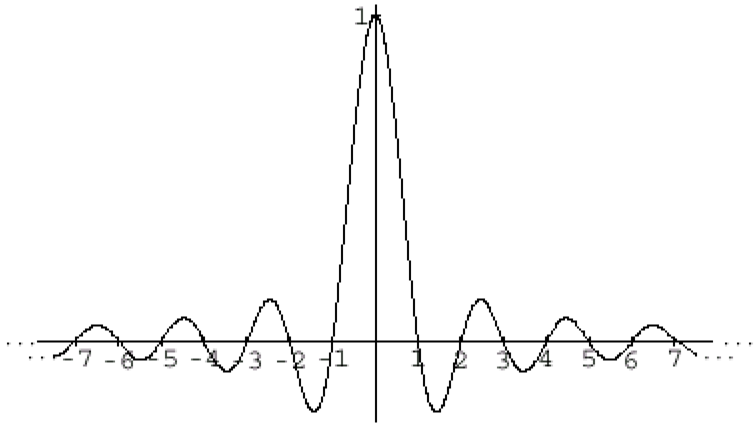
Teorema de esantionare a lui Shannon afirmã cã un semnal de bandã limitatã de frecventa Ω poate fi deplin reconstituit dacã sunt date esantioanele sale prelevate cu regularitate, mai frecvent decât dublul frecventei maxime Ω din spectrul semnalului. Perioada esantioanelor trebuie sã fie, asadar, mai micã decât $1/(2\,\Omega)$. Dacã frecventa de esantionare este $2W$, frecventa pe jumãtate W este cunoscutã ca frecventa Nyquist.
Reconstituirea semnalului $s(t)$ se realizeazã cu relatia de interpolare

$$s(t) = \sum_{n=-\infty}^{+\infty} s\left(\frac{n}{2W}\right) \frac{\sin 2\pi W\left(t - \frac{n}{2W}\right)}{2\pi W\left(t - \frac{n}{2W}\right)}$$

în care se disting esantioanele $s\left(\frac{n}{2W}\right)$, $n \in Z$ si, pentru variate translatii cu multipli întregi ai perioadei de esantionare, $\frac{n}{2W}$, functia specialã cunoscutã în literaturã ca functia esantion o variantã a functiei *sinc* definitã ca

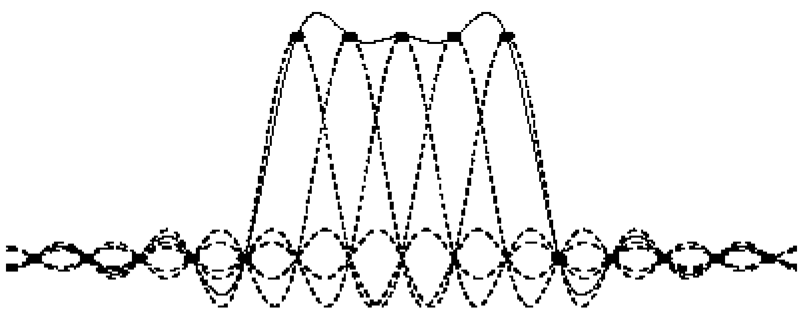
$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

cu argumentul $x = 2Wt - n$. Graficul acestei functii este prezentat în figura de mai jos.



Dupã cum este cunoscut, un semnal de bandã limitatã are în mod necesar o duratã infinitã. Invers, un semnal de duratã finitã are un spectru de frecvente infinit. Formula de interpolare de mai sus se utilizeazã totusi pentru semnale foarte diverse: periodice si aperiodice, pe suport finit sau de duratã infinitã. Afirmatia ultimã contrazice clar teorema esantionãrii. Atunci, cum se poate utiliza o relatie stabilitã în anumite conditii, în conditii diferite de cele de bazã? Este vorba, evident, de reprezentãri aproximative. Dacã, de regulã, semnalele sunt definite pe suport compact nu rãmâne decât ca o parte a spectrului de frecvente sã fie ignorat si, pe aceastã cale, sã se reducã artificial spectrul la o bandã finitã. Efectele acestei tratãri fac obiectul lucrãrii prezente. Din esantioanele unui semnal sã se reconstituie semnalul, nu importã de ce tip, din esantioanele sale prelevate mai frecvent sau mai putin frecvent.

Figura care urmeazã este o ilustrare a modului cum se poate recupera un semnal rectangular din esantioanele sale prelevate cu o anumitã periodicitate.



Sunt luate în considerare cinci esantioane nenule prelevate pe durata semnalului, alte esantioane fiind, evident, nule deoarece sunt situate în afara suportului compact al semnalului rectangular. Însumarea celor cinci functii *sinc* multiplicate cu valorile esantioanelor [..., 0, 1, 1, 1, 1, 1, 0, ...] produce "caricatura" (curba cu linie plinã) din figurã a semnalului rectangular esantionat. Semnalul "reconstituit" este nenul acolo unde ar trebui sã fie nul, este ondulat acolo unde ar trebui sã fie constant, are cresteri si descresteri în timp finit acolo unde ar trebui sã varieze brusc. Dar aceastã aproximare sau

una întrucâtva mai bunã ar putea fi satisfãcãtoare sub aspect practic-ingineresc sau ar putea fi corectatã pentru a fi utilizatã practic.
Dincolo de aspectul aproximativ comentat mai este de discutat non-cauzalitatea functiilor *sinc* din formula de interpolare. Functia *sinc* atasatã unui esantion "existã" prin valori nenule încã înainte ca esantionul sã fie prelevat/mãsurat. Ceea ce este, desigur, absurd. De aceea, în operatiile practice de reconstituire a unui semnal din esantioanele sale existã o lipsã de informatie la începutul secventei de esantioane si la finalul ei, lipsã care se traduce în aproximãri si mai grosiere în zonele apropiate capetelor intervalului finit pe care semnalul este evaluat. Erorile de aproximare pentru zonele extreme se numesc chiar asa: "efecte de capãt".
O sansã de a obtine efecte de capãt mai reduse o oferã asa-numitele ferestre. Ferestrele sunt ele însesi niste semnale care intervin multiplicativ pe lângã functia (functiile) esantion. De pildã, o fereastrã **rectangularã**

$$Rectangular(x,\tau)=\begin{cases}1 & pentru\ \ |x|\le\tau\\ 0 & \hat{i}n\ rest\end{cases}$$

anuleazã în cvasitotalitate valorile functiei esantion mai depãrtate decât cu cel mult τ de momentul propriu esantionului. Fereastra rectanghularã nu este cea mai potrivitã alegere. Din cauza variatiei bruste la anularea din capetele suportului sãu ea introduce propriile ei efecte de capãt care pot fi importatnte. De aceea, de la caz la caz, sunt utilizate ferestre de forme variate, dintre care câteva sunt definite (în ordine alfabeticã) în continuare, fãrã alte comentarii.
Fereastra **Bartlett** este o functie "cort", adicã

$$Bartlett(x,\tau)=\begin{cases}1-\dfrac{|x|}{\tau} & \text{pentru } |x|<\tau\\ 0 & \text{în rest}\end{cases}$$

Fereastra **Blackman** este definitã astfel:

$$Blackman(x,\tau)=\begin{cases}0.42+0.50\cos\left(\pi\dfrac{x}{\tau}\right)+0.08\cos\left(2\pi\dfrac{x}{\tau}\right) & \text{pentru } |x|<\tau\\ 0 & \text{în rest}\end{cases}$$

Fereastra **gaussianã** are forma generalã

$$Gauss(x,\tau,\sigma)=\begin{cases}2^{-\left(\frac{x}{\sigma}\right)^2} & \text{pentru } |x|<\tau\\ 0 & \text{în rest}\end{cases}$$

cu σ numitã si deviatia standard. Cu cât σ este mai mare cu atât este mai largã este fereastra si cu atât mai putin severã trunchierea.
Ferestrele **Hann** si **Hamming** sunt foarte asemãnãtoare. Ele diferã numai printr-un parametru, α.

$$H(x,\tau,\alpha)=\begin{cases}\alpha+(1\text{-}\alpha)\cos\left(\pi\dfrac{\mathrm{x}}{\tau}\right) & \text{pentru } |x|<\tau\\ 0 & \text{în rest}\end{cases}$$

Pentru $\alpha = 0.5$ este vorba de fereastra Hann, pentru $\alpha = 0.54$ fereastra poartã numele lui Hamming.

Fereastra **Kaiser** are un parametru ajustabil α care controlează cât de rapid se apropie de zero laturile ei. Se defineste astfel

$$Kaiser(x,\tau,\alpha)=\begin{cases}\dfrac{I_0\left(\alpha\sqrt{1-(x/\tau)^2}\right)}{I_0(\alpha)} & \text{pentru } |x|<\tau\\ 0 & \text{în rest}\end{cases}$$

cu $I_0(x)$ functia Bessel modificată de ordinul zero. Cu cât α este mai mare cu atât mai îngustă este fereastra.
Ferastra **Lanczos** este lobul central al functiei *sinc* întins pe un interval dat

$$Lanczos(x,\tau)=\begin{cases}\dfrac{\sin\left(\pi\dfrac{x}{\tau}\right)}{\pi\dfrac{x}{\tau}} & \text{pentru } |x|<\tau\\ 0 & \text{în rest}\end{cases}$$

Fereastra **Parzen** este o aproximare cubică pe portiuni a ferestrei **Gauss** de întindere doi.

$$Parzen(x)=\frac{1}{4}\begin{cases}(2+x)^3 & \text{pentru } -2\le x<-1\\ 4-6x^2-3x^3 & \text{pentru } -1\le x<0\\ 4-6x^2+3x^3 & \text{pentru } 0\le x<1\\ (2-x)^3 & \text{pentru } 1\le x<2\\ 0 & \text{în rest}\end{cases}$$

Fereastra **Welch** este simplã ca formã si anume este un polinom de gradul al doilea

$$Welch(x,\tau)=\begin{cases}1-\left(\dfrac{x}{\tau}\right)^2 & \text{pentru } |x|<\tau\\ 0 & \text{în rest}\end{cases}$$

Pachetul de programe ***Matlab*** are el însusi implementate câteva ferestre, **`bartlett`**, **`blackman`**, **`boxcar`** – o fereastrã rectangularã, **`chebwin`** – o fereastrã Cebîsev, care nu apare în lista de mai sus, **`hamming`**, **`hanning`**, **`kaiser`** si **`triang`** – o fereastrã triungiularã.

4. Modul de lucru

Obiectivul acestei lucrãri este studiul efectelor utilizãrii ferestrelor prezentate în sectiunea precedentã si a altor ferestre, pentru cazul special al interpolãrii unei secvente de cinci esantioane unitare, datã într-un exemplu mai sus.
Ilustrativ, este tratat de autorul prezentului *Ghid* cazul ferestrei rectangulare (functia **rect** si *script*-ul **wrect** scrise în ***Matlab***).
Urmând exemplul-model sau în viziune proprie se studiazã pe rând utilizarea si calitãtile interpolative ale celorlalte ferestre. Ferestrele se pot implementa de asemenea în viziune proprie sau se pot utiliza cele deja implementate în pachetul ***Matlab***.

Se recomandã atentie la parametrizarea functiilor-fereastrã. Dacã înafarã de duratã ferestrele depind si de alti parametri, se urmãreste efectul schimbãrii acelor parametri. În toate cazurile se studiazã efectul duratei ferestrelor.
Aprecierea poate fi numai vizual-calitativã dar este de preferat o apreciere cantitativã prin observarea deviatiilor semnalului interpolat fatã de semnalul initial.

5. Chestiuni de studiu

Pentru semnalul rectangular reprezentat prin cinci esantioane egale cu unitatea [..., 0, 1, 1, 1, 1, 1, 0, ...] se studiazã rezultatul interpolãrii cu functia esantion (*sinc*) ferestruitã cu fiecare din ferestrele date în sectiune **Breviar teoretic**.
Se vor observa netezimea semnalului interpolat pe portiunea nenulã, netezimea pe portiunile laterale nule, viteza de trecere de la zero la non-zero si invers, numãrul si amplitudinea faldurilor pe care interpolarea le introduce.
Este permis si chiar recomandat un studiu similar pentru alte semnale diferite de cel rectangular. Pentru extinderea studiului în aceastã directie este necesarã desigur introducerea esantioanelor corespunzãtoare semnalului propus.

```
function [y]=rect(x,tau);
% Functie fereastra de forma rectangulara de durata
2*tau
y=1.0;
if x<-tau
   y=0.0;
end
if x>tau
   y=0.0;
end
```

WRECT

```
% Script pentru combinarea functiei sinc cu ferestra
rectangulara
t=-8:0.02:8;
[l,n]=size(t);
for k=1:n
   y(k)=0;
   for l=1:5
      y(k)=y(k)+sinc(t(k)-l+3)*rect(t(k)-l+3,a);
   end
end
plot(t,y)
hold on
u=-2:2:2;
v=1+0*u;
plot(u,v,'k')
```

```
u=-8:3:-2;
v=0*u;
plot(u,v,'k')
plot(-u,v,'k')
v=0:0.5:1;
u=2+0*v;
plot(u,v,'k')
plot(-u,v,'k')
hold off
```

LUCRAREA IV

SEMNALE ALEATOARE

1. Obiectivele lucrãrii

- Generarea de semnale aleatoare
- Studiul semnalelor aleatoare.

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Un semnal aleator se mai numeste si **proces aleator**. Un proces aleator este o functie de timp $\xi(t)$ care la fiecare moment t se prezintã ca o variabilã aleatoare $X_t \in V(\Omega, K, P)$, apartinând asadar unei multimi de variabile aleatoare definite pe un câmp de probabilitate (Ω, K, P).

La un moment t fixat, valorile pe care le poate lua variabila aleatoare X_t se pot cuprinde în statistica obisnuitã a unei variabile aleatoare. Se pot defini o functie de repartitie, o functie densitate de repartitie, momente de diferite ordine etc.

Spre deosebire de semnalele de tip determinist functia de timp $\xi(t)$ cunoaste mai multe realizãri posibile. Pentru aceste realizãri, oricare din ele, se pot calcula valori medii temporale, medii temporale pãtratice etc. Fiind vorba de un semnal, aspectul evolutiei temporale este central în studiul semnalelor aleatoare.

Se spune cã un semnal este stationar dacã variabilele aleatoare X_t la momente t variate se supun unei aceleiasi legi de repartitie. Dacã legea de repartitie este cea normalã se spune cã semnalul este gaussian. Existã, desigur, si semnale care nu sunt stationare.

Semnalele (stationare) care au mediile statistice egale cu cele temporale sunt semnale ergodice.

Se vorbeste frecvent de spectrul de puteri al unui semnal aleator. Definirea acestui spectru porneste de la restrictia la un interval finit a unei realizãri (k)

$$\xi_T^{(k)}(t) = \begin{cases} \xi^{(k)}(t) & \text{pentru} \, |t| \le \dfrac{T}{2} \\ 0 & \text{pentru} \, |t| > \dfrac{T}{2} \end{cases}$$

pentru care se evaluazã transformata Fourier $X_T^{(k)}(\omega)$ pe calea obisnuitã. Cu aceste exprimãri partiale se evalueazã energia semnalului trunchiat si puterea lui medie

$$q_T^{(k)}(\omega) = \frac{1}{T} X_T^{(k)}(\omega) X_T^{*(k)}(\omega) = \frac{\left|X_T^{(k)}(\omega)\right|^2}{T}$$

O mediere pe toate realizãrile (k) posibile produce o putere medie a semnalului pe durata limitatã T. Expresia acestei puteri

$$q_T(\omega) = M\left\{\frac{\left|X_T^{(k)}(\omega)\right|^2}{T}\right\} = \frac{1}{T} M\left\{X_T^{(k)}(\omega) X_T^{*(k)}(\omega)\right\} =$$

$$= \frac{1}{T} M\left\{\int_{-T/2}^{+T/2}\int_{-T/2}^{+T/2} \xi_T^{(k)}(t_1)\xi_T^{(k)}(t_2) e^{-j\omega(t_1-t_2)} dt_1 dt_2\right\} =$$

$$= \frac{1}{T}\int_{-T/2}^{+T/2}\int_{-T/2}^{+T/2} B_T(t_1, t_2) e^{-j\omega(t_1-t_2)} dt_1 dt_2$$

contine functia de autocorelatie $B_T(t_1, t_2)$. Prin trecerea la limitã $(T \to \infty)$ se obtin distributia spectralã de puteri a semnalului în forma completã $\xi(t)$

$$q(\omega) = \int_{-\infty}^{+\infty} B(\tau) e^{-j\omega\tau} d\tau$$

în care τ noteazã diferenta $t_1 - t_2$. Asadar, densitatea spectralã de puteri si functia de autocorelatie sunt pereche Fourier.

Un rationament similar poate conduce la densitãti de puteri de interactiune între douã semnale diferite. Sub semnul integralei apare în acest caz functia de corelatie a celor douã semnale.

În prelucrarea numericã a semnalelor se lucreazã, este bine stiut, cu esantioane. De aceea, integralele de mai sus devin sume pe secvente (finite) de esantioane.

Pachetul ***Matlab*** contine douã functii fundamentale **`xcorr`** si **`xcov`** care sunt utilizate în evaluãri referitoare la semnale aleatoare (sau deterministe). Aceste functii nu sunt singurele disponibile. Prin apelul **`help`** urmat de numele acestor functii se obtin explicatii asupra parametrizãrilor necesare si/sau posibile dar, la finalul afisajului de ajutor, la linia ultimã, *see also*, sunt date si alte functii care pot fi utilizate la prelucrarea si studiul semnalelor aleatoare.

La generarea de semnale aleatoare pot fi folosite functiile **`rand`** si **`randn`**, pentru legea de repartitie uniformã pe intervalul (0, 1), respectiv pentru legea de repartitie normalã cu media zero si dispersia 1, cu parametrizarea descrisã prin **`help`**.

Pentru evaluarea spectrului de puteri, pachetul ***Matlab*** contine programe bazate pe metode de calcul diverse, unele parametrice, altele neparametrice. Toate pornesc de la expresiile

$$q_{xx}(\omega) = \sum_{m=-\infty}^{\infty} b_{xx}(m) e^{-j\omega m}$$

respectiv

$$q_{xy}(\omega) = \sum_{m=-\infty}^{\infty} b_{xy}(m) e^{-j\omega m}$$

care sunt variantele discrete ale relatiilor pentru spectrele de puteri si de puteri de interactiune ale unui semnal $x(t)$ sau ale unei perechi de semnale $x(t)$, $y(t)$.
Metoda Welch este cea mai popularã printre inginerii de specialitate, motiv pentru care este prezentatã sumar imediat.
Metoda este desigur estimativã si constã în linii mari, în evaluarea transformatei Fourier discrete a secventei în studiu

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}$$

si luarea modulului la pãtrat al rezultatului. Un semnal $x(t)$ reprezentat prin 1001 esantioane, un amestec de douã sinusoide si un zgomot alb gaussian este prelucrat dupã acest algoritm în secventa urmãtoare:

```
fs = 1000;      % frecventa de esantionare
t = 0:1/fs:1;   % momentele esantionarii pe durata unei
                  secunde
x = sin(2*pi*50*t) + 2*sin(2*pi*120*t) +
randn(size(t));
                % esantioanele semnalului
```

O evaluare brutã a spectrului de puteri al secventei `x`, numit si *periodogramã* este obtinutã prin

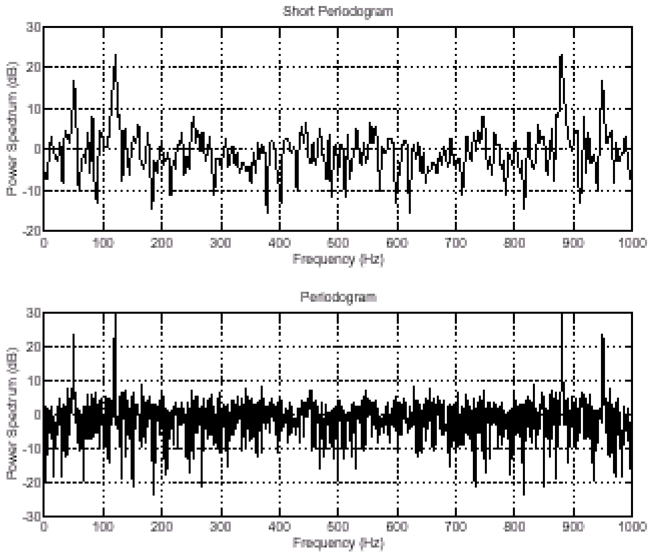
```
qxx = abs(fft(x,1024)).^2/1001;
```

în care **`fft`** este functia ***Matlab*** care calculeazã transformata Fourier a secventei (vezi **`help`**). O problemã deranjantã care însoteste acest mod de tratare este dispersia mare a rezultatului, dispersie care nu scade cu cresterea numãrului de esantioane luate în calcul. Cele douã exemple care urmeazã ilustreazã problema enuntatã

```
qxx_scurt = abs(fft(x,256)).^2/256;
plot((0:255)/256*fs,10*log10(qxx_scurt))
plot((1:1023)/1024*fs,10*log10(qxx))
```

Graficele rezultate nu diferã prea mult în ceea ce priveste dispersia puterilor din spectru.
Disperia scade dacã semnalul este segmentat pe axa timpului fãrã suprapuneri (printr-o partitionare a axei timpului), iar periodogramele rezultate sunt mediate într-una singurã

```
qxx = (abs(fft(x(1:256))).^2 + abs(fft(x(257:512))).^2
+ ...
      abs(fft(x(513:768))).^2)/(3*256);
```



O reprezentare graficã face acest fenomen observabil vizual. Cantitativ, în cazul mentionat dispersia scade de trei ori fatã de dispersia din varinta `qxx_scurt` de mai devreme. Partitionarea este limitatã de lungimea finitã a secventei de esantioane evaluatã. Existã însã posibilitatea segmentãrii cu suprapunere partialã a segmentelor

```
qxx = (abs(fft(x(1:256))).^2 + abs(fft(x(129:384))).^2
+ ...
       abs(fft(x(257:512))).^2 + ...
       abs(fft(x(385:640))).^2 + ...
       abs(fft(x(513:768))).^2 + ...
       abs(fft(x(641:896))).^2)/(6*256)
```

De data aceasta componentele mediate nu mai sunt statistic independente si dispersia începe din nou sã creascã. Existã totusi un compromis avantajos între numãrul de segmente si gradul de suprapunere.

O altã posibilitate de a îmbunãtãti periodogramele o oferã utilizarea prealabilã a unor ferestre diferite de cele rectangulare. Rezultã asa-numitele *periodograme modificate* cu o dependentã între periodogramele segmentelor mai redusã datoratã scãderii graduale la zero a ferestrelor *non*-rectangulare. Cu alegerea potrivitã a ferestrei (Hamming, Hanning, Kaiser, de pildã), la o ratã a suprapunerii de jumãtate din lãrgimea ferestrei, deci a segmentelor, se constatã o scãdere semnificativã a dispersiei periodogramelor estimate. O secventã ***Matlab*** ca urmãtoarea

```
w = hanning(256)';
qxx = (abs(fft(w.*x(1:256))).^2 + ...
       abs(fft(w.*x(129:384))).^2 + ...
       abs(fft(w.*x(257:512))).^2 + ...
       abs(fft(w.*x(385:640))).^2 + ...
       abs(fft(w.*x(513:768))).^2 + ...
       abs(fft(w.*x(641:896))).^2)/(6*norm(w)^2)
plot((0:255)/256*fs,10*log10(qxx))
```

ilustrează noua situatie.
Dacă **`psd`** este functia ***Matlab*** pentru spectrul de puteri al unui semnal, pentru spectrele interactive ale unor perechi de semnale există functia **`csd`**, în multe privinte analogă (vezi **`help`**).

4. Modul de lucru

- Se generează secvente aleatoare stationare cu instructiuni de genul
  `x = randn(1000,1);`
  urmate de reprezentarea lor grafică
  `plot(x)`
- Se calculează functiile de (auto)covariatie si de (auto)corelatie cu functiile Matlab **`xcorr`** si **`xcov`** cu optiuni diverse, se reprezintă grafic rezultatele
- Se generează secvente de zgomot colorat $(a, b \neq 0)$ stationare $b \in (-1,1)$ sau nestationare $b \in \{-1,1\}$, stabile (|*b*| < 1) sau nestabile (|*b*| >1) cu secventa

```
clear
r(1)=randn;
x(1)=r(1);
a=0.3;
b=0.85;
for i=2:1000
   r(2)=randn;
   x(i)=b*x(i-1)+r(2)+a*r(1);
   r(1)=r(2);
end
plot(x)
```

- Se realizează programe ***Matlab*** de tip *script* conform indicatiilor din sectiunea **Breviar teoretic** si din sectiunea prezentă

5. Chestiuni de studiu

- Se observă vizual forma semnalelor generate
- Se studiază spectrele de puteri ale semnalelor generate dar si ale semnalelor propuse în sectiunea **Breviar teoretic**
- Se evaluează spectrul de interactiune a două semnale diferite
- Se apreciază "culoarea" secventelor aleatoare evaluate

LUCRAREA V

CARACTERISTICILE FILTRELOR CU RÃSPUNS IMPULSIONAL FINIT (FIR)

1. Obiectivele lucrãrii

- Studiul caracteristicilor filtrelor cu rãspuns impulsional finit (FR)

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Un filtru numeric care rãspunde la intrarea standard *impuls Dirac* într-un numãr finit de pasi dupã care rãspunsul este nul este un filtru cu rãspuns impulsional finit, prescurtat FR (F inite Im pulse Response). Functia de transfer a unui filtru de acest gen este în domeniul frecventelor

$$H(f) = \sum_{i=0}^{N-1} a_i e^{-j2\pi f iT}$$

si în domeniul complex al varibilei z

$$H(z) = \sum_{i=0}^{N-1} a_i z^{-i}$$

Functia pondere corespunzãtoare a filtrului este

$$h(t) = \sum_{i=0}^{N-1} a_i \delta(t - iT)$$

În toate cele trei relatii se observã combinatii liniare finite cu coeficientii a_i ($i = 0, 1, \ldots, N-1$), coeficienti care definesc pe deplin filtrul. Rãspunsul filtrului la impulsul $\delta(t)$ aplicat la intrare este tocmai aceastã secventã de coeficienti.

Rãspunsul filtrului la o secventã (periodicã) $x(n)$ ($n = 0, 1, \ldots, N-1$) se calculeazã în domeniul timp ca o convolutie numericã

$$y(n) = \sum_{i=0}^{N-1} a_i x(n - i)$$

cu echivalentul în domeniul frecventelor

$$Y(k) = H(k)X(k) \text{ pentru } k = 0, 1, \ldots, N-1$$

În expresia din urmã, sunt notate cu majuscule componentele transformatelor Fourier numerice ale secventelor temporale din expresia anterioarã.

Filtrele de tipul FR sunt cu fazã linia rã adicã functia de transfer introduce în rãspuns o fazã care variazã liniar cu frecventa. În scrierea

$$H(f) = R(f)e^{-j\Phi(f)}$$

functia $R(f)$ este realã, iar faza are forma $\Phi(f) = \Phi_0 + 2\pi f\tau$, cu τ o constantã care reprezintã întârzierea trecerii prin filtru. Întârzierea de grup, derivata fazei în raport cu frecventa este o constantã, τ.

Dacã functia $R(f)$ se descompune într-o sumã a douã functii, una parã, cealaltã imparã

$$R(f) = P(f) + I(f)$$

dacã se pune $\Phi_0 = 0$, functia pondere a filtrului se poate scrie

$$h(t+\tau) = 2\int_0^{\infty} P(f)\cos 2\pi ft\, df + 2j\int_0^{\infty} I(f)\sin 2\pi ft\, df$$

Scrierea aceasta ascunde în sine o simetrie a coeficientilor a_i ai filtrului fatã de momentul τ si permite particularizãri utile cu $I(f)$ sau $P(f)$ identic nule.

Din teoria generalã a sistemelor se cunoaste exprimarea functiei de transfer a unui sistem discret ca un raport a douã polinoame în variabila complexã z. Se mai spune cã functia de transfer este o functie rationalã în variabila z. Aceastã functie are un numãr de zerouri, rãdãcini ale numãrãtorului si un numãr de poli, care sunt rãdãcini ale numitorului. Functiile de transfer ale filtrelor FR sunt tot rationale în z dar au numai zerouri.

Lucrarea prezentã recurge ca si celelalte la functiile ***Matlab***. Unele functii genereazã structuri numite *system*. Asemenea structuri sunt argumente sintetice, obiectuale, foarte potrivite pentru alte functii ***Matlab***. De pildã functia (vezi **`help`**)

```
tf([0.2 0.5],[1 -0.3 0.7],1)
```

produce rezultatul

```
Transfer function:
   0.2 z + 0.5
-----------------
z^2 - 0.3 z + 0.7
Sampling time: 1
```

în care se observã functia rationalã (fractia) în z si perioada de esantionare. În particular, aceeasi functie cu argumentele modificate

```
sis=tf([0.25 0.5 0.25],1,1);
```

depune în **`sis`** urmãtoarea structurã

```
Transfer function:
0.25 z^2 + 0.5 z + 0.25
Sampling time: 1
```

care are numai zerouri (numitorul este constant si egal cu unitatea si de aceea nu apare). Sistemul **`sis`** si functia lui de transfer se referã la fitrul FR numit si filtrul *cosinus plus*, despre care s-a vorbit si la orele de curs. Obiectul **`sis`** poate servi ca argument functiei **`bode`**

```
bode(sis)
```

care în aceastã formã de apelare produce graficele modulului (dublu logaritmic) si fazei (semilogaritmic) pentru functia de transfer în domeniul frecventelor, pe o perioadã. Pentru diagrame liniare se poate apela functia bode sub forma

```
[a,ph,w]=bode(sis);
```

si apoi din matricea cu amplitudinile `a`, din cea cu fazele `ph` si din vectorul frecventelor `w` se recolteazã date pentru

```
plot(w,b)
```
care traseazã doritele (eventual) diagrame liniare.
Date pentru reprezentãri grafice se pot obtine si prin utilizarea prealabilã a functiei **`freqz`** (vezi **`help`**) urmatã de **`plot`** sau **`loglog`**, **`semilogx`**, **`semilogy`** etc.

4. Modul de lucru

Se compun *script*-uri ***Matlab*** pentru urmãtoarele puncte:

- Se studiazã rãspunsul în frecventã al filtrelor exemplificate la curs, adicã filtrul *cosinus* si filtrul *cosinus plus*
- Se studiazã câteva filtre cu lista coeficientilor simetricã sau antisimetricã, mai lungã decât în exemplele date la curs
- Se studiazã caracteristicile de amplitudine si de fazã ale câtorva filtre FR oarecare, cu lista coeficientilor neregulatã

5. Chestiuni de studiu

- Se examineazã caraceristicile filtrelor FR cunoscute ca filtrul *cosinus* si filtrul *cosinus plus*
- Se observã simetria sau antisimetria carateristicii de amplitudine a filtrelor FR cu coef icienti cu simetrie si cu antisimetrie fatã de pozitia temporalã medianã
- Se observã liniaritatea cu frecventa a fazelor
- Se fac observatii comparative asupra benzii de trecere a filtrelor studiate (banda de trecere se defineste prin frecventa la care atenuarea este de 3 *dB*) si asupra selectivitãtii prin calculul diferentei între frecventele la care atenuarea este de 3 *dB* si de 20 *dB*

Se recomandã utilizarea, dupã caz, atât a diagramelor liniare cât si a celor logaritmice, semilogaritmice sau dublu logaritmice.
Se recomandã încercarea de a utiliza functia Matlab **`grpdelay`** care calculeazã întârzierea de grup în functie de frecventã. Se cer concluzii proprii si apropriate.

LUCRAREA VI

PROIECTAREA FILTRELOR CU RÃSPUNS IMPULSIONAL FINIT

1. Obiectivele lucrãrii

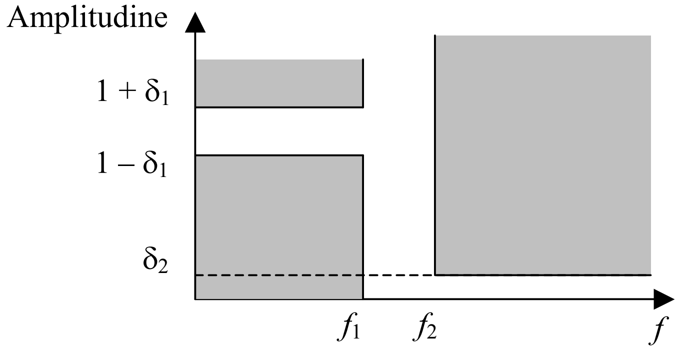
- Studiul metodelor de proiectare a filtrelor cu rãspuns impulsional finit

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Proiectarea unui filtru numeric constã uzual în determinarea unor coeficienti din expresia unei functii de tranfer astfel ca anumite caracterisitici ale filtrului, de regulã cea de amplitudine, sã satisfacã anumite conditii. Figura alãturatã este o ilustrare a unei "teme" de proiectare. Caracteristica de amplitudine a filtrului trebuie sã se înscrie în spatiul fãrã umbre din diagramã: în zona de trecere unitatea sã fie aproximatã cu eroarea $\pm\delta_1$, în zona de oprire sã se realizeze o atenuare pânã sub nivelul δ_2, iar banda de tranzitie situatã între banda de trecere si banda de oprire sã fie de o anumitã întindere, de preferat cât mai micã, între frecventele f_1 si f_2.



O metodã de proiectare este cea care utilizeazã ferestre. Se considerã filtrul numeric *trece-jos* ideal cu frecventa de tãiere ω_0. Acest filtru are amplificarea unitarã la frecvente mai mici decât frecventa de tãiere si amplificare nulã pentru frecvente mai mari decât aceasta. Seceventa de rãspuns la intrarea *impuls* este

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{jn\omega}d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{jn\omega}d\omega = \frac{\omega_0}{\pi}\operatorname{sinc}\left(n\frac{\omega_0}{\pi}\right)$$

Acest filtru nu poate fi implementat deoarece este cu rãspuns impulsional infinit si nu este cauzal. Pentru a crea un rãspuns în timp finit este necesarã o fereastrã. Prin retinerea sectiunii centrale a rãspunsului impulsional, prin trunchiere, se obtine un filtru FR cu f azã liniarã. De pildã, un filtru cu 51 de coeficienti, cu frecventa de tãiere $\omega_0 = 0.4\pi$, într-o fomulare ***Matlab*** este

```
b = 0.4*sinc(0.4*(-25:25));
```

Fereastra este dreptunghiularã în aceastã tratare si este dupã teorema lui Parseval un filtru de lungime 51 care aproximeazã cel mai bine filtrul ideal. Pentru a vedea rãspunsul se utilizeazã secventa ***Matlab***

```
[H.w] = freqz(b,1,512,2);
plot(w,abs(H)), grid
```

În grafic se vor putea observa faldurile datorate fenomenului Gibbs, care nu dispar dacã numãrul coeficientilor creste, dar pot fi atenuate, diminuate dacã se utilizeazã ca multiplicator în domeniul timp o altã fereastrã. Fereastra Hamming, de pildã, introdusã în secventa de instructiuni

```
B = b.*hamming(51);
[H.w] = freqz(b,1,512,2);
plot(w,abs(H)), grid
```

produce o certã ameliorare.

Functiile ***Matlab*** **`fir1`** si **`fir2`** se bazeazã tocmai pe utilizarea ferestrelor si fac operatiile descrise mai devreme automat. Prima din cele douã functii este implementarea metodei clasice de proiectare a filtrelor în configuratii standard: *trece-jos*, *trece-bandã*, *trece-sus*, *opreste-bandã*.

Declaratiile

```
n = 50;
wn = 0.4;
b = fir1(n,wn);
```

genereazã vectorul linie `b` de coeficienti ai filtrului de ordinul `n` prin utilizarea ferestrei Hamming. Filtrul obtinut este un filtru FR *trece-jos* cu frecventa de tãiere `wn`, un numãr între 0si 1, unde unitatea corespunde frecventei Nyquist, jumãtatea frecventei de esantionare. Pentru un filtru *trece-sus* se ataseazã ca argument sirul `'high'`. Pentru *trece-bandã* se introduce `wn` ca un vector cu douã componente, frecventele care delimiteazã banda de trecere. Pentru *opreste-bandã* se adaugã ca argument sirul `'stop'`. Alte detalii relativ la functia **`fir1`** se obtin prin **`help`**.

Este interesantã si functia **`kaiserord`** care furnizeazã parametri de intrare pentru functia **`fir1`**. Pentru cazul ferestrei Kaiser, functia **`kaiserord`**, la specificarea limitelor de frecventã si a amplitudinii maxime admise a faldurilor se obtin parametrii potriviti pentru functia **`fir1`**.

Functia ***Matlab*** **`fir2`** este destinatã proiectãrii filtrelor cu caracteristica de trecere oarecare, nu în formele standard mentionate în cazul functiei **`fir1`**. Se utilizeazã si de data aceasta ferestre dar forma caracteristicii filtrului se constituie ca în succesiunea de comenzi

```
n = 50;
f = [0 0.4 0.5 1];
m = [1 1 0 0];
b = fir2(n,f,m);
```

cu un vector de frecvente `f` si un vector de amplificãri/atenuãri `m` de aceeasi lungime.

Filtrele de tipul FR sunt filtre cu f azã liniarã adicã functia de transfer introduce în rãspuns o fazã care variazã liniar cu frecventa. În scrierea

$$H(f) = R(f)e^{-j\Phi(f)}$$

functia $R(f)$ este realã, iar faza are forma $\Phi(f) = \Phi_0 + 2\pi f\tau$, cu τ o constantã care reprezintã întârzierea trecerii prin filtru. Întârzierea de grup, derivata fazei în raport cu frecventa este o constantã, τ.

Dacã functia $R(f)$ se descompune într-o sumã a douã functii, una parã, cealaltã imparã

$$R(f) = P(f) + I(f)$$

dacã se pune $\Phi_0 = 0$, functia pondere a filtrului se poate scrie

$$h(t+\tau) = 2\int_0^{\infty} P(f)\cos 2\pi ft\, df + 2j\int_0^{\infty} I(f)\sin 2\pi ft\, df$$

Asa cum s-a verificat în **Lucrarea V**, scrierea aceasta ascunde în sine o simetrie a coeficientilor a_i ai filtrului fatã de momentul τ si permite particularizãri utile cu $I(f)$ sau $P(f)$ identic nule. Functiile **`fir1`** si **`fir2`** au în vedere filtrele cu $I(f) \equiv 0$, cunoscute si ca filtre FR de tipurile I si II, cu numãr par sau impar de coeficienti.

Pachetul ***Matlab*** contine si alte fuctii mai cuprinzãtoare, care oferã mijloace mai generale de a specifica filtrul urmãrit. Functiile **`firls`** si **`remez`** permit proiectarea altor tipuri de filtre cum sunt filtrele FR de tipul III si IV cu $P(f) \equiv 0$, filtre de diferentiere, filtre transformatoare Hilbert etc. Aceleasi functii permit ponderarea erorilor de proiectare variabil cu frecventa, ignorarea erorilor în unele zone din spectrul de frecvente.

Functia **`firls`** este o extindere a functiilor **`fir1`** si **`fir2`**. Ea realizeazã minimizarea integralei pãtratului diferentelor dintre caracteristica de amplitudine doritã si caracterisitica realizatã efectiv de filtrul proiectat.

Functia **`remez`** este o implementarea a algoritmului lui Remez în varianta îmbunãtãtitã de Park-McClellan prin utilizarea aproximãrii Cebîsev. Este minimizatã în acest caz eroarea maximã între caracteristicile doritã si proiectatã, de unde denumirea metodei ca metoda *minimax*. Faldurile rezultate sunt egale de unde o altã denumire a metodei: *echifald* (*equiripple*). Metoda Park-McClellan este poate cea mai utilizatã de inginerii din domeniu.

Cele douã functii au sintaxã similarã. Diferã numai algoritmul de calcul. În forma standard cu ele se proiecteazã filtre FR de tipul I si II. Secventa

```
n = 20;
f = [0 0.4 0.5 1];
a = [1 1 0 0];
b = remez(n,f,a);
```

ilustreazã cazul unui filtru *trece-jos* cu amplitudinea aproximativ unitarã de la 0 la 0.4 *Hz* si amplitudine aproximativ nulã de la 0.5 la 1 *Hz*. Intervalul 0.4 la 0.5 *Hz* nu intrã în nici un fel în calcule, este o zonã de tranzitie care nu intereseazã în mod deosebit lãrgimea ei influentând totusi calitatea filtrului în zonele de trecere si de oprire.

Secventa de completare a celei de mai sus

```
bb = firls(n,f,a);
[h,w] = freqz(b);
[hh,w] = freqz(bb);
plot(w/pi,abs(h),w,abs(hh),'- -'), grid
```

permite vizualizarea faldurilor obtinute prin cele două modalităti de proiectare.
Prin apel la **`help`** se pot afla si întelege si alte facilităti ale functiilor-program ***Matlab*** **`firls`** si **`remez`**. De pidă, extinderea vectorilor frecvente-amplitudini permite proiectarea unor filtre *multibandă*, iar prin specificarea unor ponderi se pot ierarhiza erorile, mai mici într-o anumită zonă a spectrului, mai relaxate în altă zonă a spectrului.
Secventa

```
n = 20;               % ordinul filtrului
f = [0 0.4 0.5 1];    % limitele benzilor
a = [1 1 0 0];        % amplitudini dorite
w = [1 10];           % ponderi
b = remez(n,f,a,w);
```

aduce faldurile în zona de oprire a spectrului la amplitudini mai mici de 10 ori fată de cele din banda de trecere.
În secvente similare celor de mai sus, initializarea multiplă a vectorilor de frecvente si amplitudini

```
    f = [0 0.1 0.15 0.25 0.3 0.4 0.45 0.55 0.6 0.7
0.75 0.85 0.9 1];
    a = [1 1 0 0 1 1 0 0 1 1 0 0 1 1];
```

produce un filtru *multibandă*. Desigur, dacă se atasează un vector de ponderi, dimensiunea lui trebuie adaptată corespunzător.

4. Modul de lucru

Se compun *script*-uri ***Matlab*** pentru următoarele puncte:

- Proiectarea unor filtre *trece-jos* de tipul FR prin m ijlocirea functiilor ***Matlab*** mentionate în sectiunea **Breviar teoretic**
- Realizarea unor grafice comparative pentru metode de proiectare diferite, pentru ponderarea diferită a erorilor
- Proiectarea unor filtre *multibandă* cu vectorii de frecvente si amplitudini dati în sectiunea **Breviar teoretic** sau constituiti la liberă alegere

5. Chestiuni de studiu

- Se examinează relatia între ordinul filtrului FR *trece-jos* si calitatea filtrului
- Se observă efectul ferestrelor de diferite tipuri asupra calitătii filtrelor
- Se observă liniaritatea fazelor cu frecventa
- Se modifică banda (benzile) de tranzitie si se observă influenta asupra faldurilor

Se recomandă utilizarea, după caz, atât a diagramelor liniare cât si a celor logaritmice, semilogaritmice sau dublu logaritmice.
Se recomandă încercarea de a utiliza functia ***Matlab*** **`grpdelay`** care calculează întârzierea de grup în functie de frecventă. Se recomandă extragerea unor concluzii proprii si apropriate.
Se recomandă explorarea si a altor facilităti de proiectare oferite de pachetul ***Matlab***.

LUCRAREA VII

CARACTERISTICILE SI PROIECTAREA FILTRELOR CU RÃSPUNS IMPULSIONAL INFINIT

1. Obiectivele lucrãrii

- Observarea caracterisiticlor de amplitudine si de fazã ale filtrelor cu rãspuns impulsional infinit (R)
- Studiul metodelor de proiectare a filtrelor cu rãspuns impulsional infinit

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Filtrele cu rãspuns impulsional infinit cele mai simple sunt cele de ordinul I si Ipur recursive. Ele au f unctiile de transfer

$$H(z) = \frac{1}{1 - bz^{-1}} = \frac{z}{z - b}$$

respectiv

$$H(z) = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}} = \frac{z^2}{z^2 + b_1 z + b_2}$$

Relatiile pentru calculul iesirilor sunt, in ordine corespondentã

$$y(n) = x(n) + by(n-1)$$

$$y(n) = x(n) - b_1 y(n-1) - b_2 y(n-2)$$

Prin înlocuirea variabilei z cu $e^{j\omega}$ se pot obtine functiile de transfer în domeniul frecventelor. Se pot separa caracteristica de amplitudine si caracteristica de fazã.

Mai general, un filtru R este un sistem care genereazã o secventã $y(n)$ din secventa $x(n)$ conform relatiei

$$y(n) = \sum_{l=0}^{L} a_l x(n-l) - \sum_{k=1}^{K} b_k y(n-k)$$

Se observã cã în relatia ultimã apar si valori anterioare ale iesirii. Functia de transfer exprimatã în operatorul de deplasare înapoi pe axa timpului $B = z^{-1}$ se poate scrie ca un raport de douã polinoame

$$H(B) = \frac{\sum_{l=0}^{L} a_l B^l}{1 + \sum_{k=1}^{K} b_k B^k}$$

Polinoamele sunt foarte frecvent de acelasi grad sau gradul numitorului este mai mare decât cel al numãrãtorului.
În ***Matlab*** existã functii care creazã obiecte de tipul *system*, de pildã functia

```
tf([0.2 0.5],[1 -0.3 0.7],1) % transfer function = 
tf
```

care produce rezultatul

```
Transfer function:
   0.2 z + 0.5
---------------
z^2 - 0.3 z + 0.7
Sampling time: 1
```

în care se observã functia rationalã (fractia) în z si perioada de esantionare. Pe calea aceasta se pot crea sisteme de ordine încã mai mari. Prin utilizarea functiei **`bode`** cu argument un obiect de tipul *system* se pot vizualiza modulul si faza functiei de transfer a sistemului în domeniul frecventelor, sau se pot colecta date privind aceste importante caracterisitici.
Dacã coeficientii a_l si b_k sunt reali atunci functia de transfer este un numãr complex astfel încât

$$\overline{H(B)} = H(\overline{B})$$

cu bara pentru luarea conjugatelor, si rãspunsul în frecventã se poate scrie ca de obicei

$$H(\omega) = |H(\omega)| e^{-j\Phi(\omega)}$$

Modulul si faza pot fi exprimate ca

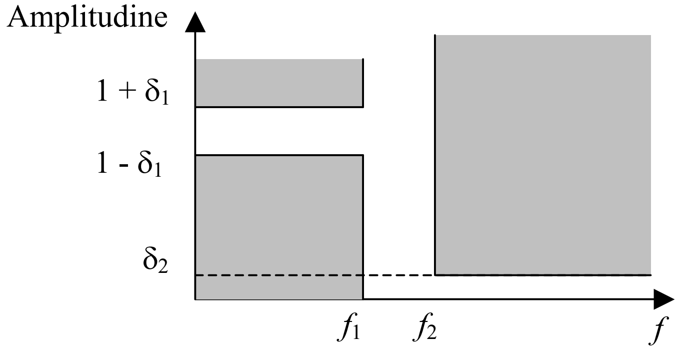
$$|H(\omega)|^2 = \left[H(B^{-1})H(B)\right]_{B=e^{-j\omega}}$$

$$\Phi(\omega) = \frac{1}{2} j \ln\left[\frac{H(B^{-1})}{H(B)}\right]_{B=e^{-j\omega}}$$

Filtrele R nu m ai au variatia fazei liniarã cu frecventa cum se întâmpla la filtrele FR si nici întârzierea de grup constantã. Ecuatia pentru întârzierea de grup este

$$\tau(\omega) = \frac{d\Phi}{d\omega} = \frac{d\Phi}{dB}\frac{dB}{d\omega} = -\mathrm{Re}\left[\frac{1}{B}\frac{d}{dB}\ln H(B)\right]_{B=e^{-j\omega}}$$

si produce un rezultat dependent de frecventã. Aceastã caracteristicã este produsã de functia ***Matlab*** **`grpdelay`**. Printre cele câteva functii consacrate analizei rãspunsului filtrelor la frecvente diferite si aici este de luat în consideratie functia **`freqz`**.
Proiectarea unui filtru numeric din clasa R necesitã, ca si în cazul filtrelor de tipul FR, determ inarea unor coeficienti din expresia unei functii de tranfer astfel ca anumite caracterisitici ale filtrului, de regulã cea de amplitudine, sã satisfacã anumite conditii. Figura alãturatã este o ilustrare a unei "teme" de proiectare. Caracteristica de amplitudine a filtrului trebuie sã se înscrie în spatiul fãrã umbre din diagramã: în zona de trecere unitatea sã fie aproximatã cu $\pm\delta_1$, în zona de oprire sã se realizeze o atenuare pânã sub nivelul δ_2, iar banda de tranzitie situatã între banda de trecere si banda de oprire sã fie de o anumitã întindere, de preferat cât mai micã, între frecventele f_1 si f_2.



Coeficientii unui filtru R se pot calcu la direct dintr-o asa-numitã functie model, o functie realã definitã pe axa frecventelor. Functiile model pot fi, de pildã, functiile Butterworth, functiile Bessel si Cebîsev sau functiile eliptice. Toate aceste functii au proprietãti selective binecunoscute.
Neajunsul major al acestor functii constã în lipsa lor de periodicitate când, este cunoscut, functia doritã este periodicã. Este necesarã o trecere de la intervalul [0 f_s] la axa realã realizatã print r-o transformare conformã în planul complex, cu proprietãtile urmãtoare:

(1) Axa imaginarã se transformã în cercul unitate
(2) O functie rationalã în variabila complexã *s* este transformatã într-o functie rationalã în variabila *z*
(3) Stabilitatea este conservatã

Existã câteva astfel de transformãri despre care s-a discuta la cursul de *Procesarea numericã a semnalelor*. Si despre utilizarea functiilor model s-a discutat acolo îndeajuns. Functiile ***Matlab*** specifice se numesc **`butter, cheby1, cheby2, ellip`**.
Prima dintre ele utilizeazã functia model Butterworth de ordinul *n* care are forma

$$|F(\omega)|^2 = \frac{1}{1+(\omega/\omega_c)^{2n}}$$

Urmãtoarele douã sunt pentru proiectarea filtrelor R *trece-jos* cu falduri în banda de trecere, respectiv în banda de oprire. Expresiile functiilor model sunt

$$|H(\omega)|^2 = \frac{1}{1+\varepsilon^2 T_n^2\left(\dfrac{\omega}{\omega_c}\right)}$$

respectiv

$$|H(\omega)|^2 = \frac{1}{1+\varepsilon^2\left[\dfrac{T_n(\omega_c)}{T(\omega_c/\omega)}\right]^2}$$

cu T_n polinoamele Cebîsev de gradul *n*, cu ε o constantã care dacã este egalã cu unitatea face din frecventa de tãiere ω_c frecventa la care atenuarea este de 3 *dB*.
Ultima din cele patru functii ***Matlab*** foloseste functiile eliptice incluse în

$$T^2(u) = \frac{1}{1+\varepsilon^2 \mathrm{sn}^2(u,k_1)}$$

cu

$$u = \int_0^{\phi} \frac{d\theta}{(1-m\sin^2\theta)^{1/2}}$$

si generează solutii de proiectare care prezintă falduri atât în banda de trecere cât si în banda de oprire.

Functia **`buttord`** predetermină ordinul unui filtru Butterworth. Există functii similare **`cheb1ord`**, **`cheb2ord`**, **`ellipord`** care fac acelasi lucru pentru filtrele de tipul respectiv ascuns în numele lor de apel.

Cu functiile **`freqz`**, **`filter`** se pot calcula caractristicile de transfer ale unui filtru dat sau proiectat, respectiv se poate filtra o secventă de intrare precizată.

Toate functiile ***Matlab*** aduse în discutie pot fi utilizate si la proiectarea altor tipuri de filtre: *trece-bandă*, *trece-sus*, *opreste-bandă* dacă parametrizarea este modificată conform cu situatia concretă urmărită (vezi **`help`**).

4. Modul de lucru

Se compun *script*-uri ***Matlab*** pentru următoarele puncte:

- Observarea caracteristicilor de frecventă pentru filtre pur recursive de ordinele I si I si pentru alte f iltre imaginate de studenti, date prin coeficientii polinoamelor din expresia ratională a functiilor de transfer
- Proiectarea unor filtre *trece-jos* de tipul R prin m ijlocirea functiilor ***Matlab*** mentionate în sectiunea **Breviar teoretic**
- Realizarea unor grafice comparative pentru metode de proiectare diferite, pentru ponderarea diferită a erorilor
- Proiectarea unor filtre de alte tipuri, diferite de tipul *trece-jos*

5. Chestiuni de studiu

- Se examinează relatia între ordinul filtrului R *trece-jos* si calitatea filtrului
- Se observă functiilor model de diferite tipuri asupra calitătilor filtrelor
- Se observă variatia fazelor cu frecventa
- Se observă dependenta de frecventă a timpului de întârziere de grup
- Se examinează amplitudinile faldurilor
- Se compară ordinele unui filtru R si unui filru FR la aceleasi performante

Se recomandă utilizarea, după caz, atât a diagramelor liniare cât si a celor semilogaritmice sau dublu logaritmice.

Se recomandă explorarea si a altor facilităti de proiectare oferite de pachetul ***Matlab***.

LUCRAREA VIII

PROIECTAREA GENERALÃ A FILTRELOR NUMERICE

1. Obiectivele lucrãrii

- Proiectarea fitrelor FR si R de tipuri diferite de tipul *trece-jos*
- Proiectarea fitrelor FR si R de alte tipuri prin transformarea unui filtru *trece-jos*

2. Aparaturã si suport documentar

- Calculatoare PC în configuratie obisnuitã
- Prezentul *Ghid de lucrãri*
- Notele de curs de la disciplina *Procesarea numericã a semnalelor*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

În lucrãrile anterioare s-au studiat si s-au proiectat exclusiv filtre de tipul *trece-jos*. Discutia si metodologia de la filtrele *trece-jos* este suficientã dacã se completeazã cu câteva reguli de trecere de la acel tip la celelalte tipuri de filtre utilizate în practicã. Regulile sunt simple si sunt prezentate imediat sub forma unor schimbãri de variabilã în reprezentãrile standard în spatiul complex ale functiilor de transfer prototip analogice. Astfel, dacã se noteazã cu ω_T frecveta de tãiere a filtrului *trece-jos* si cu ω_L, respectiv cu ω_H, frecventele de tãiere care delimiteazã banda de trecere sau de oprire a filtrului necesar pot fi întelese substitutiile din tabelul alãturat.

Tipul filtrului	Transformarea	Frecvente de tãiere
Trece-jos	$s \to \frac{\omega_T}{\omega'_T} s$	ω'_T
Trece-sus	$s \to \frac{\omega_T \omega'_T}{s}$	ω'_T
Trece-bandã	$s \to \omega_T \frac{s^2 + \omega_L \omega_H}{s(\omega_H - \omega_L)}$	ω_L, ω_H
Opreste-bandã	$s \to \omega_T \frac{s(\omega_H - \omega_L)}{s^2 + \omega_L \omega_H}$	ω_L, ω_H

Pentru transformarea unui filtru *trece-jos* în alt filtru *trece-jos* sau într-un filtru *trece-sus*, noua frecventã de tãiere a fost marcatã cu un accent.
Transformãrile conservã faldurile eventuale ale caracteristicii de frecventã dar modificã benzile de trecere si de oprire.
Se poate schita acum o strategie de proiectare a unui filtru de un tip oarecare:

1. Se fixeazã frecventele de tãiere

2. Se proiectează filtrul analogic trece-jos
3. Se adoptă transformarea de frecvente potrivită pentru trecerea la filtrul necesar
4. Se aplică o transformare filtrului analogic pentru a obtine filtrul digital urmărit

Pentru trecerea de la un filtru digital *trece-jos* la alte filtre de alte tipuri sunt necesare schimbări de variabilă în spatiul complex *z*. Tabelul alăturat contine transformările potrivite fiecărui caz.

Tipul filtrului	**Transformarea**	**Frecvente de tăiere**
Trece-jos	$z^{-1} \to \dfrac{z^{-1} - a}{1 - az^{-1}}$	$\omega'_T \quad a = \dfrac{\sin\dfrac{\omega_T - \omega'_T}{2}}{\sin\dfrac{\omega_T + \omega'_T}{2}}$
Trece-sus	$z^{-1} \to \dfrac{z^{-1} + a}{1 + az^{-1}}$	$\omega'_T \quad a = -\dfrac{\cos\dfrac{\omega_T + \omega'_T}{2}}{\cos\dfrac{\omega_T - \omega'_T}{2}}$
Trece-bandă	$z^{-1} \to -\dfrac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$	$\omega_L,\, \omega_H$ $a_1 = -2\alpha K/(K+1)$ $a_2 = (K-1)/(K+1)$ $\alpha = -\dfrac{\cos\dfrac{\omega_H + \omega_L}{2}}{\cos\dfrac{\omega_H - \omega_L}{2}}$ $K = \cot\dfrac{\omega_H - \omega_L}{2} \tan\dfrac{\omega_T}{2}$
Opreste-bandă	$z^{-1} \to -\dfrac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}$	$\omega_L,\, \omega_H$ $a_1 = -2\alpha/(K+1)$ $a_2 = -(K-1)/(K+1)$ $\alpha = -\dfrac{\cos\dfrac{\omega_H + \omega_L}{2}}{\cos\dfrac{\omega_H - \omega_L}{2}}$ $K = \tan\dfrac{\omega_H - \omega_L}{2} \tan\dfrac{\omega_T}{2}$

Strategia de proiectare constă în pasii următori:
1. Se proiectează un filtru trece jos digital
2. Se foloseste transformarea de frecvente adecvată conform tabelului prezentat.

Pachetul de programe ***Matlab*** contine functii care face posibilă proiectarea directă a filtrelor de tip diferit de tipul *trece-jos*. Sunt aceleasi functii,

`butter, cheby1, cheby2, ellip`, cu o listã de argumente modificatã (vezi **`help`**).

4. Modul de lucru

- Se proiecteazã un filtru *trece-jos* cu frecventa de tãiere 0,5.
- Se proiecteazã direct, cu functiile ***Matlab*** consacrate filtre de tipurile *trece-jos* de bandã diferitã, *trece-sus*, *trece-bandã* sau *opreste-bandã*
- Utilizând filtrul *trece-jos* proiectat la primul punct si transformãrile date în tabelele din sectiunea **Breviar teoretic** se reproiecteazã filtre de tipurile *trece-jos* de bandã diferitã, *trece-sus*, *trece-bandã* sau *opreste-bandã*
- Se comparã rezultatele obtinute pe cele douã cãi
- Se realizeazã grafice ale rãspunsurilor în frecventã al filtrelor proiectate pe una sau alta din cel douã cãi

5. Chestiuni de studiu

- Se observã si se verificã corectitudinea transformãrilor date în sectiunea Breviar teoretic prin comparatie cu rezultatele ***Matlab***
- Se examineazã conservarea sau neconservarea faldurilor la transformãrile de frecventã date în sectiunea **Breviar teoretic**
- Se observã relatia între zonele de tranzitie ale filtrelor de diverse tipuri

LUCRAREA IX

SEMNALE SI FUNCTIILE WAVELET UNI- SI BIDIMENSIONALE

1. Obiectivele lucrãrii

- Studiul câtorva tipuri de functii wavelet uni- si bidimensionale
- Utilizãrile posibile ale functiilor wavelet în analiza semnalelor uni- si bidimensionale

2. Aparaturã si suport documentar

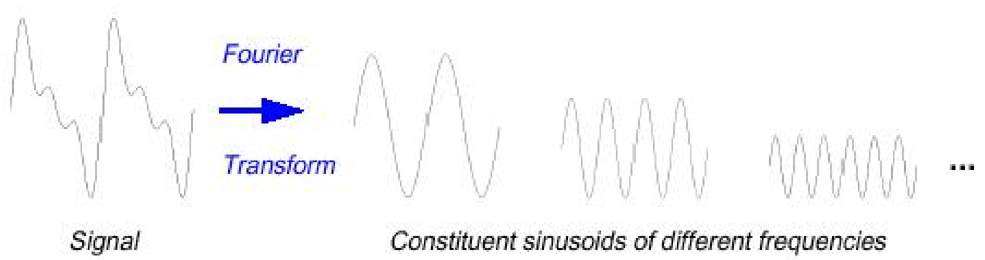
1. Calculatoare PC în configuratie obisnuitã
2. Prezentul *Ghid de lucrãri*
3. Notele de curs de la disciplina *Procesarea numericã a semnalelor*
4. Manuale de programare, documentarea *on-line* (*Help*)

3. Breviar teoretic

Functiile wavelet sunt functii de duratã finitã, definite pe suport finit cum se mai spune, care diversificate prin dilatãri si translatii se pot constitui într-o bazã a unui spatiu de functii care pot fi în particular semnale. O bazã de functii wavelet se mai numeste si hiperbazã din motivul simplu cã este o reuniune de mai multe baze, fiecare din ele capabilã a descrie anumite procese temporale, mai lente sau mai rapide, cuprinse într-un semnal.
Sub aspect matematic analiza Fourier este reprezentatã de *transformarea Fourier*

$$S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t}\,dt$$

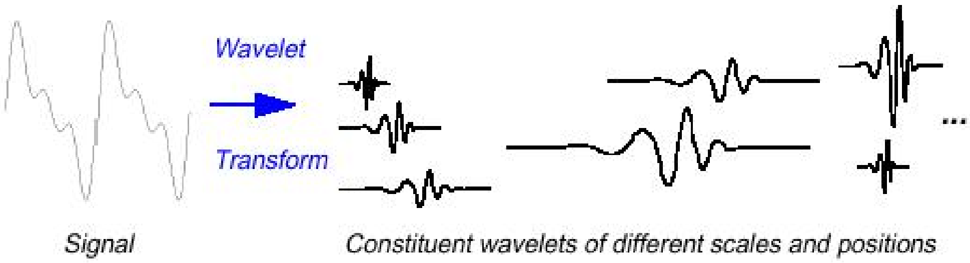
care este însumarea peste toatã axa timpului a produsului dintre semnalul $s(t)$ si o exponentialã complexã. Exponentiala complexã poate fi scrisã ca suma a douã componente sinusoidale, una realã, cealaltã imaginarã. Rezultatul transformãrii îl constituie *coeficientii Fourier* $S(\omega)$ care multiplicati cu sinusoide de frecvente potrivite ω produc componentele sinusoidale ale semnalului analizat.



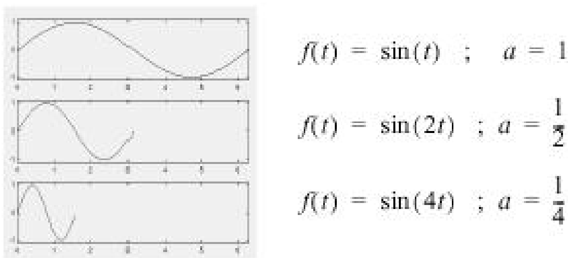
Analog existã o *transformare prin functii wavelet continuã* (CWT) care se defineste ca suma pe axa timpului a semnalului multiplicat de versiuni scalate si deplasate prin translatie ale functiei undeletã χ.

$$C(\text{scarã},\text{pozitie}) = \int_{-\infty}^{\infty} s(t)\chi(\text{scarã},\text{pozitie},t)dt$$

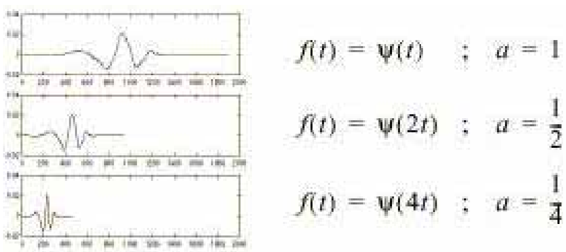
Rezultatul transformãrii prin functii wavelet continuã (CWT) este constituit si de data aceasta din mai multi coeficienti, *coeficientii C ai functiilor wavelet*, care sunt functii de scarã si de pozitie. Prin multiplicarea lor cu functiile wavelet scalate si pozitionate adecvat pe axa timpului se genereazã functiile wavelet componente ale semnalului.



Analiza prin functii wavelet produce o imagine timp-scarã a semnalului.
Scalarea unei functii wavelet nu este altceva decât comprimarea sau desirarea, întinderea ei. Matematic operatia se leagã de un factor de scalare, notat aici cu *a*. Un exemplu cu sinusoide este dat imediat. Rolul factorului de scalare este aproape evident.

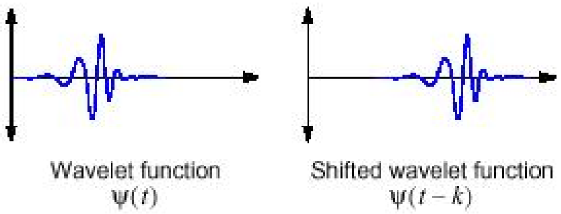


Factorul de scalare are rol asemãnãtor si în cazul functiilor wavelet.



Rezultã foarte clar din figurile date cã pentru o sinusoidã sin(ωt) factorul de scalare a se situează în raport invers cu frecventa ω. Analog, în analiza prin functii wavelet scara se aflã în raport invers de variatie cu frecventa semnalului. Subiectul va fi reluat mai departe.

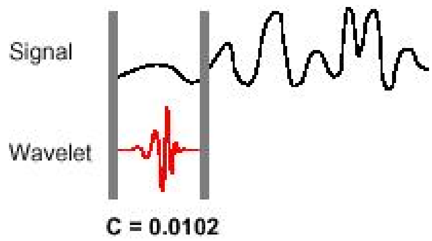
Deplasarea (prin translatie a) unei functii wavelet înseamnã simpla ei întârziere (sau grãbire, anticipare). Matematic, întârzierea functiei $f(t)$ cu k se exprimã prin $f(t-k)$.



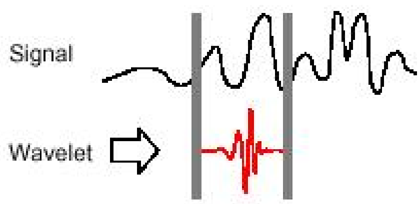
Acum, transformarea prin functii wavelet continuã este, cum s-a spus, suma peste multimea realã a timpului, a produsului dintre semnal si versiuni scalate si deplasate ale functiilor wavelet. Rezultã coeficientii functiilor wavelet. Coeficientii sunt functii de scarã si pozitie.

Procesul este realmente simplu.

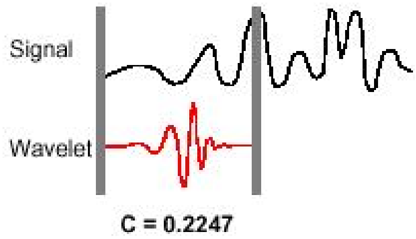
1. Se ia functia wavelet si se compară cu o portiune incipientă a semnalului.
2. Se calculează numărul C care exprimã tãria corelatiei functiei wavelet cu semnalul în acea portiune de început. Cu cât valoare lui C este mai mare cu atât mai mare-i similitudinea semnal-functie wavelet. Rezultatul calculului depinde de functia wavelet aleasã.



3. Se deplasează functia wavelet la dreapta ca în figura următoare si se repetă calculul de mai devreme până semnalul este epuizat.



4. Se scalează (desirã) functia wavelet si se repetă pasii anteriori.



5. Se repetã pasii anteriori pânã la epuizarea tuturor scãrilor.

În final se obtine o listã de coeficienti produsi la diferite scãri de diferite sectiuni ale semnalului. Coeficientii sunt rezultatul unei regresii efectuate cu semnalul originar pe functii wavelet.

Functiile wavelet exprimã foarte bine partea superioarã a spectrului de frecvente al semnalelor. Pentru partea inferioarã sunt de utilitate o functie de scalare φ si familia generatã prin deplasãri/translatii ale ei.

Din functiile wavelet unidimensionale se pot construi functii wavelet bidimensionale. Regula este urmãtoarea: a) functia de scalare se obtine prin multiplicarea functiei de scalare unidimensionale cu ea însãsi

$$\varphi(x_1, x_2) = \varphi(x_1)\varphi(x_2)$$

b) functiile undeletã se construiesc prin multiplicarea în perechi a functiei de scalare si a functiilor undeletã unidimensionale

$$\psi^1(x_1, x_2) = \varphi(x_1)\psi(x_2)$$
$$\psi^2(x_1, x_2) = \psi(x_1)\varphi(x_2)$$
$$\psi^3(x_1, x_2) = \psi(x_1)\psi(x_2)$$

Functiile undeletã bidimensionale sunt la fiecare scarã în numãr de trei. Una este destinatã exprimãrii detaliilor pe o directie, cealaltã pe directia a doua, ultima pe o directie diagonalã.

4. Modul de lucru

- Se invocã din platforma *Matlab* programul `wavedemo`.
- Se urmãresc pas cu pas semnalele propuse, mai întâi cele unidimensionale, apoi cele bidimensionale si evolutia lor în diverse etape de prelucrare prin intermediul transformãrilor prin functii wavelet.
- La fiecare actionare a butonului "Next" se observã modificãrile grafice si de parametri din fereastra principalã si se citesc comentariile din fereastra cu mesaje/comentarii.
- Se încearcã utilizarea procedurilor din programul demonstrativ pentru semnale creatie proprie, de pildã amestecuri aditive de sinusoide afectate sau nu de zgomote de asemenea aditive.

5. Chestiuni de studiu

- Se urmãresc calitãtile descriptive ale hiperbazelor de functii wavelet pentru functiile/semnalele din spatiile uni- si bidimensional.
- Se observã capacitatea functiilor wavelet de a condensa informatia intr-o formã stocabilã mai economic.
- Se urmãresc procedurile de reducere a zgomotelor care însotesc semnalele utile.

N.B. Lucrarea prezentã se executã pe durata a douã sedinte.