

2009.03.20.

# Rezgés analizáló függvények

*Függvény fejlesztés MATLAB-ban*

Dobján Tibor

# Rezgés analizáló függvények

*Függvény fejlesztés MATLAB-ban*

## Analizálás menete

---

1. Beolvasás

2. Szűrés

Trendtelenítés

Csonkolás

Zajszűrés

3. Spektrumképzés

Gyors Fourier Transzformáció

*3.1.1. Ablakolás*

APSD

*3.1.2. Csúszóátlag képzés*

4. Összehasonlítás (kereszt spektrum képzés)

CPSD

Koherencia

Fázis

5. Megjelenítés

Spektrumrajzoló

6. Kiértékelés

## Függvények

---

### 1. Beolvasás

#### 1.1. Beolvas

##### Feladata

A Labview által készített adatfájlból átalakítja az adatokat a Matlab-ban eddig használt formátumra.

##### Használata

```
[sxr_g]=beolvas(nev);
```

##### Példa a használatra

```
[eredmeny]=beolvas('D:\downloads\test.lvm');
```

##### Algoritmus

```
function [sxr_g]=beolvas(nev)
%[sxr_g]=beolvas(nev)

[adat]=importdata(nev, '\t', 22);
sxr_g.csatornak=str2double(strrep(adat.textdata(13), 'Channels', ''));

%sxr_g.szoveg=adat.textdata;

for i=1:sxr_g.csatornak
    sxr_g(:,i).t=adat.data(:,1);
    sxr_g(:,i).y=adat.data(:,i+1);
    sxr_g(:,i).datum=strrep(adat.textdata(9), 'Date', '');
    sxr_g(:,i).ido=strrep(adat.textdata(10), 'Time', '');
    sxr_g(:,i).start=adat.data(1:1);
    sxr_g(:,i).end=adat.data(length(adat.data(:,1)),1);
end;
plot(sxr_g(1,1).t, sxr_g(1,1).y, sxr_g(1,2).t, sxr_g(1,2).y, sxr_g(1,3).t, sxr_g(1,3).y, sxr_g(1,4).t, sxr_g(1,4).y);
```

##### Ismert hibák

- Bizonyos kinyerhető információkat nem olvas ki.

##### Fejlesztési javaslatok

- Lehet, nem feltétlen kellene a kapott formátumhoz ragaszkodni, viszont ha módosítunk rajta a többi függvényt is módosítani kell.

## 2. Szűrés

### 2.1. Trendtelenítés

#### Feladata

A mért jeltől el távolítja a trendet, és 0 középvé alakítja.

ATL – a szűrőeljárás bekért paramétere (10-100)

XL, XM – belső ellenőrző vektorok

#### Használata:

```
[trendtelen,atl,xl,xm] = trendtelenit(trendes);
```

#### Példa a használatra:

```
[trendtelen,atl,xl,xm] = trendtelenit(a);
```

#### Algoritmus:

```
function [trendtelen,atl,xl,xm] = trendtelenit(trendes)
% [trendtelen,atl,xl,xm] = trendtelenit(trendes)

n=length(trendes);
valasz=inputdlg('ÁTL:', 'Átlagolt levonás', 1, {'100'});
atl=str2double(valasz);

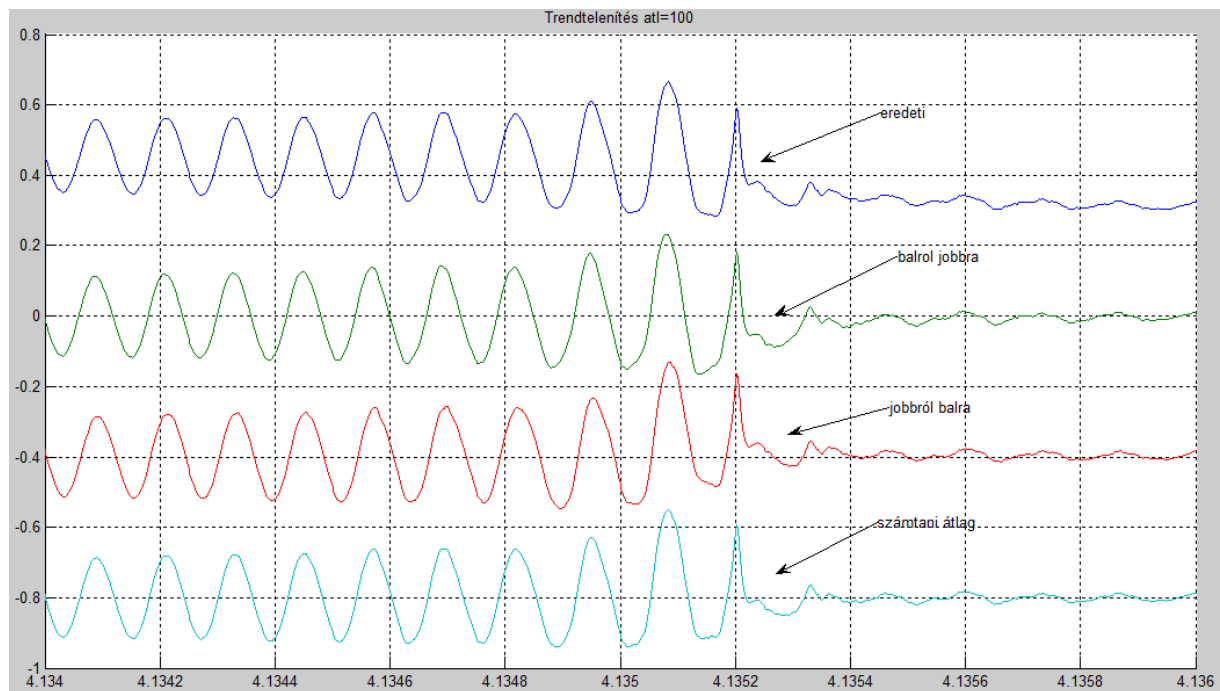
%elozetes helyfoglalas
xl=zeros(n,1);
xm=zeros(n,1);
trendtelen=zeros(n,1);

%szures balrol jobbra
for l=atl+1:n
    a=0;
    for i=1:atl
        a=a+trendes(l-i);
    end;
    xl(l)=trendes(l) - (a/atl);
end;

%szures jobbról balra
for m=n-atl:-1:1
    a=0;
    for j=atl:-1:1
        a=a+trendes(m+j);
    end;
    xm(m)=trendes(m) - (a/atl);
end;

%ket szures atlagolasa
for k=atl+1:1:n-atl
    trendtelen(k) = (xl(k) + xm(k)) / 2;
end;
```

## Teszt eredmények:



## Ismert hibák:

- A szűrőalgorithmus az első és az utolsó ATL darabszámú értéket változatlanul hagyja.

## Fejlesztési javaslatok

- A szűrőalgorithmust ki lehetne bővíteni egy automatikus csonkolással.
- Több szűrő algorithmust is lehetne integrálni.

## 2.2. Csonkolás

### Feladata:

A trendtelenített tömböt, kettő lehető legmagasabb hatványa hosszúságúra csonkítani.

### Használata:

```
[csonkolt,tki,i]=csonkol(csonkolando,tbe,atl);
```

ATL – a trendtelenítéskor nem átalakított részek levágásához

TBE – a bemenő idő

TKI – a csonkoláshoz korrigált idő

I – a hosszúság kettesalapú hatványkitevője

## Példa a használatra:

```
[csonkolt,tki,i]=csonkol(trendtelen,t,atl);
```

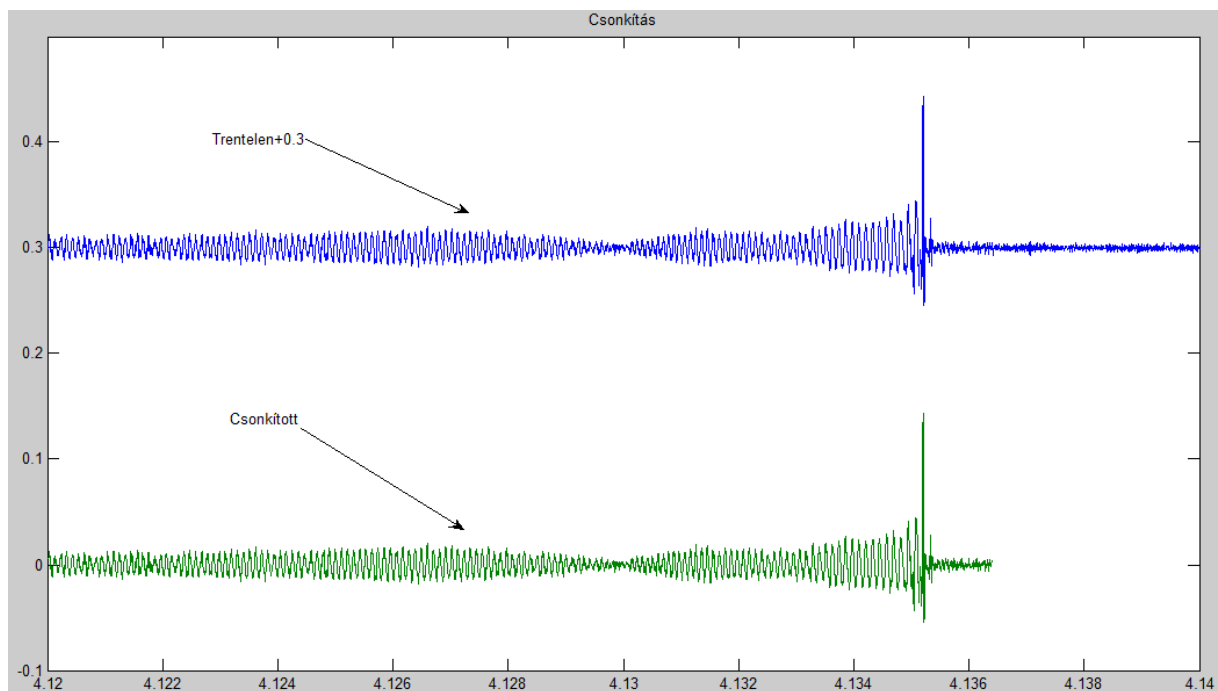
## Algoritmus:

```
function [csonkolt,tki,i]=csonkol(csonkolando,tbe,atl)
%[csonkolt,tki,i]=csonkol(csonkolando,tbe,atl)

%inicializacio
b=length(csonkolando);
a=1;
i=0;
%legnagyobb hossz keresés
while a < b
    a=a*2;
    i=i+1;
end;
i=i-1;

%preallokacio
csonkolt=zeros(2^i,1);
tki=zeros(2^i,1);
%csonkolas
for j=1:1:2^i
    csonkolt(j)=csonkolando(atl+j);
    tki(j)=tbe(atl+j);
end;
```

## Teszt eredmények:



## Fejlesztési javaslatok

- A mérőpontokat és a hozzájuk tartozó időjel darabokat egy tömbben kellene tárolni.

### 2.3. Zajszűrés

#### Feladata:

A mérést zavaró nagyfrekvenciás összetevők eltávolítása.

$I$  – a hosszúság kettesalapú hatványkitevője

#### Használata:

```
[zajszurt]=zajszures(zajos,i);
```

#### Példa a használatra:

```
[zajszurt]=zajszures(csonkolt,i);
```

#### Algoritmus:

```
function [zajszurt]=zajszures(zajos,i)
%[zajszurt]=zajszures(zajos,i)

%ablakválaszték feltöltése
for j=1:i
    S(j)=cellstr(num2str(2^j));
end;

%ablak definíció
[c,d] =
listdlg('Name','Szűrés','PromptString','Válassz!','SelectionMode','Single',
'ListString',S,'InitialValue',1,'CancelString','Mégse','ListSize',[100
i*15]);
%ablak kezelés
if (d == 1)
    transzformalt=fft(zajos);
    for j=1+2^c:2^i
        transzformalt(j)=0;
    end;
    %ell=transzformalt; %ellenőrzés képpen az fft taranszformált
    zajszurt=ifft(transzformalt);
end;
```

#### Ismert hibák

Nem nulla szűrés esetén az ifft eredménye komplex.

### 3.1. Spektrumképző

#### Feladata:

Auto spektrum tömb kiszámítása súlyozott átlaggal.

## Használata:

```
[APSD,ATLAPSD,ACF]=spektrumkepzes(szurtjel);
```

## Példa a használatra:

```
[APSD,ATLAPSD,ACF]=spektrumkepzes(sxr_g(1,1).y);
```

## Algoritmus:

```
function [APSD,ATLAPSD,ACF]=spektrumkepzes (szurtjel)
%Spektrumkepzes 1.1
%[APSD,ATLAPSD,ACF]=spektrumkepzes (szurtjel)

%FFT ablakválaszték feltöltése
for j=1:6
    S(j)=cellstr(num2str(2^(j+5)));
end;

%ablak kezelés
[c,d] = listdlg('Name','FFT
tömb','PromptString','Válassz!','SelectionMode','Single','ListString',S,'I
nitialValue',1,'CancelString','Mégse','ListSize',[100 90]);
c=c+5; % válasz kompenzáció

%Ablakoló ablak választék feltöltése
S={'Boxcar','Hanning','Hamming','Triangle','Bartlett-
Hann','Bartlett','Blackman','Blackman-Harris','Bohman','Flat Top
weighted','Gaussian','Nuttall-defined Blackman-Harris','Parzen (de la
Valle-Poussin) '};
s=[@boxcar,@hann,@hamming,@triang,@barthannwin,@bartlett,@blackman,@blackma
nharris,@bohmanwin,@flattopwin,@gausswin,@nuttallwin,@parzenwin];
%dialógus definíció
[e,f] = listdlg('Name','Ablak
típus','PromptString','Válassz!','SelectionMode','Single','ListString',S,'I
nitialValue',1,'CancelString','Mégse','ListSize',[200 195]);
x=s(e); %választott függvénykezel?

%Lépés hossz bekérés
valasz=inputdlg('Lépésköz:', 'Lépésköz bekérése',1,{ '1' });
y=str2double(valasz);

%Súly bekérés
valasz=inputdlg('Súly:', 'Súly bekérése APSD súlyozáshoz',1,{ '0.99' });
es=str2double(valasz);

%preallokáció
a=zeros(2^c,1);
apspd=zeros(2^c,1);
APSD=zeros(2^c, length(szurtjel));
ATLAPSD=zeros(2^c,1);

w=window(x,2^c); %ablak függvény létrehozása

if (d == 1) && (f == 1) % ha OK gombot nyomtak akkó:
```



```
for g=(2^c):y:(length(szurtjel)) %minden ponthoz
    for i=1:2^c %ablakméret
        a(i)=szurtjel(g-(2^c)+i); %ablak létrehozása
    end;
    a=a.*w; %szorzás ablakfüggvényel
    a=fft(a); %FFT
    con=conj(a); % FFT Konjugát el?állítás
    b=a.*con; %APSD(1)
    apsd=(b+apsd*es)/(1+es); %súlyozás
    APSD(:,g)=apsd;
    ATLAPSD=ATLAPSD+apsd;
end;
ATLAPSD=ATLAPSD/(length(szurtjel)-(2^c));
ACF=ifft(ATLAPSD);
end;
```

### Ismert hibák

- ATLAPSD és az ACF számítása vagy megjelenítése nem pontos

## Spektrum rajzolás

### Feladata:

A kiszámított értékek megjelenítése a kiértékeléshez felhasználható formában.

### Használata:

```
spektrumotrajzollog(spektrumtomb, csonkoltido, eredetitomb);
```

### Példa a használatra:

```
spektrumotrajzollog(APSD, tki, sxr_g);
```

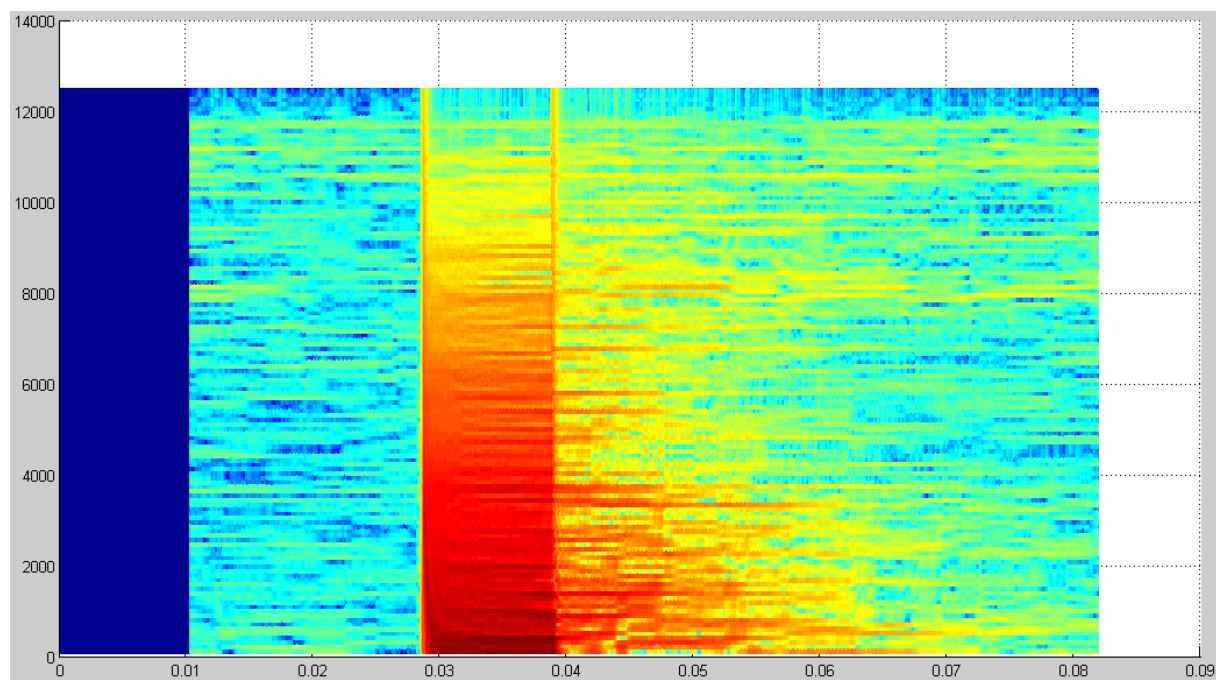
## Algoritmus:

```
function
[x,y,x2,y2,ftengely,minimum]=spektrumotrajzollog(spektrumtomb,csonkoltido,eredetitomb)
% spektrumotrajzollog(spektrumtomb,csonkoltido,eredetitomb);
% spektrumotrajzollog(APSD,tki,sxr_g);

z=size(spektrumtomb);
x=z(1);
y=z(2);
elemekszama=length(eredetitomb(1,1).y);
dt=(eredetitomb(1,1).end-eredetitomb(1,1).start)/elemekszama;
fmax=1/(2*dt);
df=(2/x)*fmax;
ftengely=0+df:df:fmax;
x2=length(ftengely);
y2=length(csonkoltido);

minimum=max(max(spektrumtomb));
for i=1:x
    for j=1:y
        if ((spektrumtomb(i,j)<minimum) && (spektrumtomb(i,j)>0))
            minimum=spektrumtomb(i,j);
        end;
    end;
end;
for i=1:x
    for j=1:y
        if (spektrumtomb(i,j)==0)
            spektrumtomb(i,j)=minimum;
        end;
    end;
end;
minimum=min(min(spektrumtomb));
mesh(csonkoltido,ftengely,log10(spektrumtomb((x2+1):(2*x2),1:y2)));
view(2);
```

### Teszt eredmények:



### Fejlesztési javaslat

Subplottal alá lehetne tenni az időjelet, és a többi függvényt.