

ფურიე ანალიზი

უწყვეტი ფურიე გარდაქმნა:

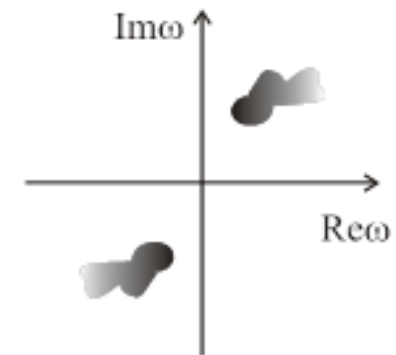
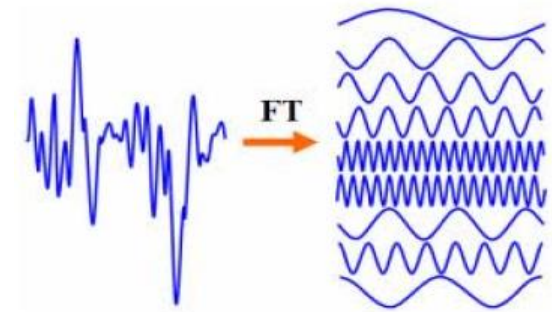
$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \exp(i\omega t) dt$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) \exp(-i\omega t) d\omega$$

$f(t)$ – რეალური;

$F(\omega)$ – კომპლექსური;

ფიზიკური ფუნქციის ფურიე სახის სიმეტრია: $F^*(-\omega) = F(\omega)$



ფურიე გარდაქმნის თვისებები

სუპერპოზიცია: $f_1(t) + f_2(t) \rightarrow F_1(\omega) + F_2(\omega)$

კონვოლუცია: $\int_{-\infty}^{+\infty} f_1(\tau) f_2(t - \tau) d\tau \rightarrow F_1(\omega) F_2(\omega)$

წრფივი დიფერენციალური განტოლების ფურიე ანალიზი

არაერთგვაროვანი დიფ. განტოლება:

$$\frac{d^2 f(t)}{dt^2} + \alpha \frac{df(t)}{dt} + \omega_0^2 f(t) = g(t)$$

ფურიე სახე:

$$-\omega^2 F(\omega) + i\alpha\omega F(\omega) + \omega_0^2 F(\omega) = G(\omega)$$

ალგებრული განტოლება (თუ მოხერხდა ინტეგრალის აღება):

$$G(\omega) = \int_{-\infty}^{+\infty} g(t) \exp(i\omega t) dt$$

☞ *N რიგის ერთგვაროვანი დიფ. განტოლება მუდმივი კოეფიციენტებით:*

$$a_n \frac{d^n f(t)}{dt^n} + a_{n-1} \frac{d^{n-1} f(t)}{dt^{n-1}} + \dots a_1 \frac{df(t)}{dt} + a_0 f(t) = 0$$

ფურიე გაშლის შემდეგად ვიპოვით განტოლების სპექტრს:

$$a_n (-i\omega)^n + a_{n-1} (-i\omega)^{n-1} + \dots - a_1 i\omega + a_0 = 0$$

ამონახსნის აღდგენა:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) \exp(-i\omega t) dt$$

☞ *N* რიგის ერთგვაროვანი დიფ. განტოლება ცვლადი კოეფიციენტებით:

$$a_n \frac{d^n f(t)}{dt^n} + a_{n-1} \frac{d^{n-1} f(t)}{dt^{n-1}} + \dots a \frac{df(t)}{dt} + a_0(t) f(t) = 0, \quad a_0(t) = \alpha t$$

(წრფივად) ცვლადი კოეფიციენტის ფურიე გაშლა: $\alpha t f(t) \rightarrow i\alpha \frac{dF(\omega)}{d\omega}$

$$\left[a_n (-i\omega)^n + a_{n-1} (-i\omega)^{n-1} + \dots + a_0 \right] F(\omega) + i\alpha \frac{dF(\omega)}{d\omega} = 0$$

N რიგის დიფ. განტოლება გადავწერეთ პირველი რიგის დიფ. განტოლებად.

ამონახსნის აღდგენა (შებრუნებული ფურიე გარდაქმნა):

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) \exp(-i\omega t) d\omega$$

მონაცემების დამუშავება: ჰილბერტის გარდაქმნის მეთოდით

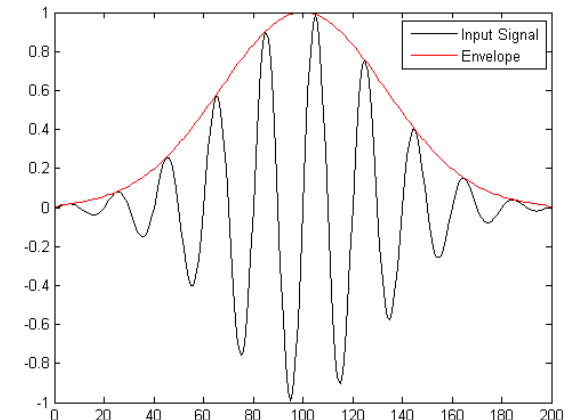
სიგნალი (ვექტორი): $s(t)$

- რას უდრის სიგნალის მყისი ამპლიტუდა და ფაზა?
- სიგნალის რა ნაწილია ჰარმონიული/პერიოდული, და რა ნაწილის აპერიოდული?

„სიგნალის პროცესირების/ფოლტრაციის მეთოდები“

ფუნქციის ჰილბერტის გაფართოება:

$$D(t) \equiv S(t) + iS_H(t)$$



სადაც ფუნქციის ანალიზური გაფართოება მოცემულია ჰილბერტის გარდაქმნით:

$$S_H(t) = \sqrt{\frac{2}{\pi}} \int_{-\infty}^{+\infty} \frac{S(\tau)}{t - \tau} d\tau$$

კონვოლუცია: $S(t), \frac{1}{\pi t}$

ფურიე სახე: $S_H(\omega) = -is(\omega) \operatorname{sgn}(\omega)$

ჰარმონიული კომპონენტის ფაზის წანაცვლება $\pi/2$ -ით.

მაგალითად:

$$S(t) = A \cos(\omega t),$$

$$S_H(t) = iA \sin(\omega t)$$

რთული სიგნალის ფაზა და ამპლიტუდა:

$$D(t) \equiv S(t) + iS_H(t) = A(t)e^{i\varphi(t)}$$

$$A(t) = \sqrt{S^2(t) + S_H^2(t)}$$

ფურიე გარდაქმნით ვექტორის ჰილბერტის გარდაქმნის გამოთვლის პროცედურა ($\omega > 0$):

1. $S(t) \rightarrow s(\omega)$
2. $\text{Im}[s(\omega)] \leftrightarrow \text{Re}[s(\omega)]$: $s(\omega) \rightarrow s_H(\omega)$
3. $s_H(\omega) \rightarrow S_H(t)$

Matlab

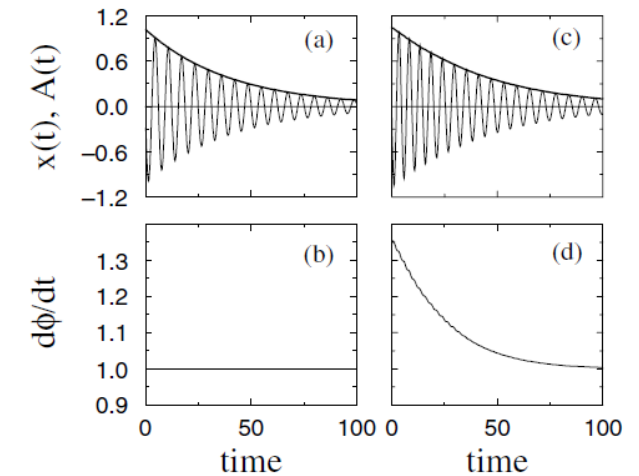
```
D = hilbert(S);
```

დავალება 10.1.

დაფინგის ოსცილატორი:

$$x''(t) + 0.05 x'(t) + x(t) + x(t)^3 = 0$$

იპოვეთ: $x(t)$ და რხევის ტრაექტორიის მომვლები (ამპლიტუდა) $A(t)$.

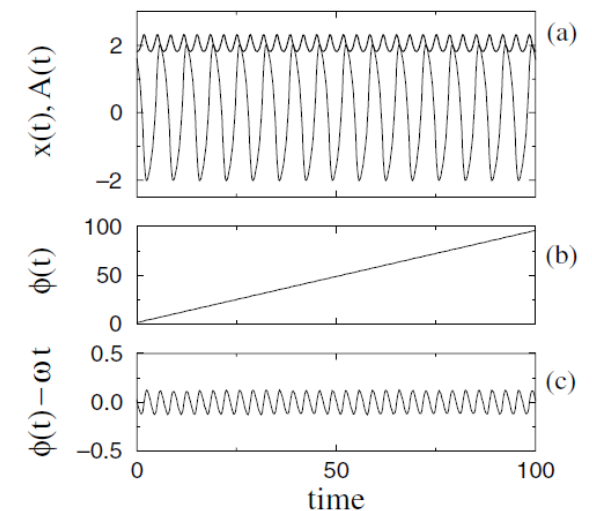


დავალება 10.2.

ვან დერ პოლის განტოლება:

$$x''(t) - (1-x^2) x'(t) + x(t) = 0$$

იპოვეთ: $x(t)$ და რხევის ტრაექტორიის მომვლები (ამპლიტუდა) $A(t)$.
დახაზეთ ამონახსნის ფაზური პორტრეტი (x, x');



დისკრეტული ფურიე გარდაქმნა

ფურიე მწკრივი:

$$X(k) = \sum_{j=1}^N x(j) \omega_N^{(j-1)(k-1)}$$

$$x(j) = \frac{1}{N} \sum_{k=1}^N X(k) \omega_N^{-(j-1)(k-1)}$$

$$\omega_N \equiv e^{2i\pi/N}, \quad N < \infty$$

ვექტორი: $x(1), x(2), \dots, x(N)$

ფურიე სახე: $X(1), X(2), \dots, X(N)$

`fftshift (0-2pi) -> (-pi,+pi)`

სტანდარტული დისკრეტული გაშლა: `ft`

სწრაფი ფურიე გაშლა (fast Fourier transform): `fft / ifft` ($N=2^n$, მაგ: 512, 1024, ...)

Matlab

```
t0 = 0;
tfin = 100;
dt = 0.1;

t = (t0:dt:tfin);
T = tfin-t0;
N = length(t);

f = sin(2*pi*t) + 0.1*rand(1,N);
G = fft(f);

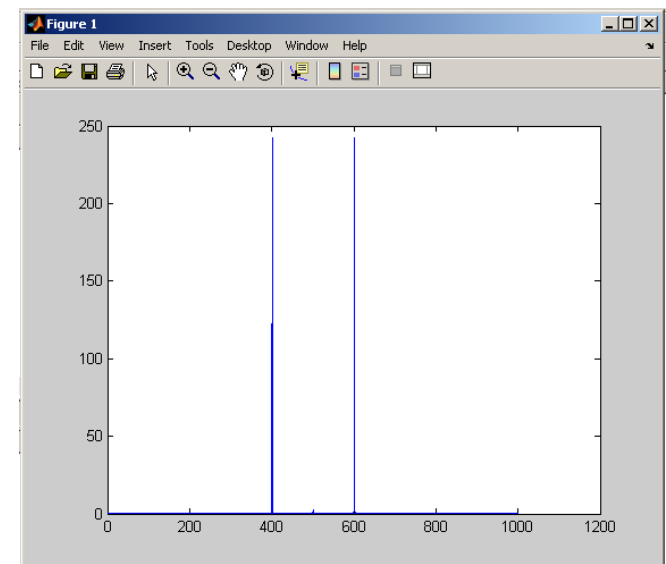
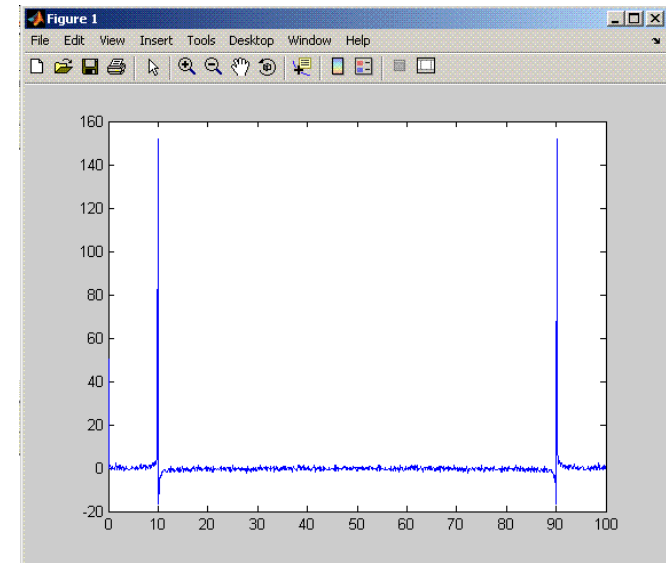
plot(t,real(G));
```

ენერგიის სპექტრალური სიმკვრივე

```
GG = fftshift(G);

Eomega = abs(GG).^2;

plot(Eomega);
```



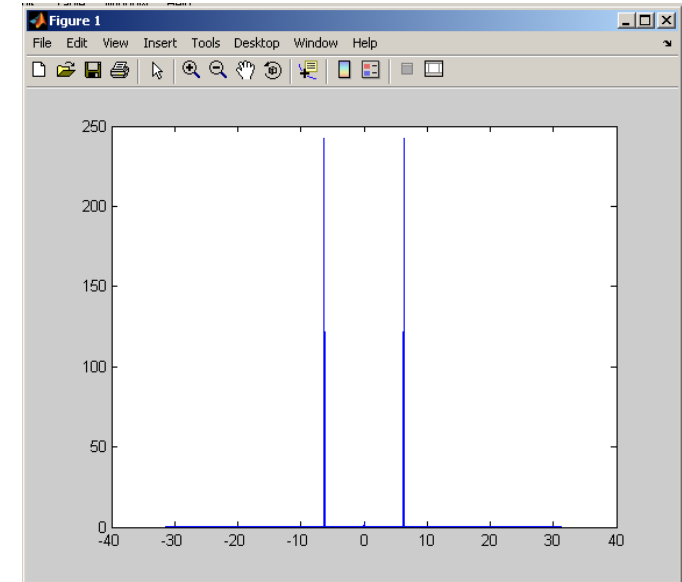
სიხშირები:

$$\Delta\omega_{max} = \frac{2\pi}{\Delta t}$$

$$\Delta\omega_{min} = \frac{2\pi}{T}$$

$$\omega = (-\Delta\omega_{max}/2 : \Delta\omega_{min} : \Delta\omega_{max}/2)$$

```
omega = (-pi/dt : 2*pi/T : pi/dt );
plot(omega,Eomega);
```



შეამოწმეთ:

Matlab ფუნქციები:

```
fft
fftshift
ifft
```

დავალება 10.3.

ჩაატარეთ მზის ლაქების რიცხვის სპექტრალური ანალიზი დროში (Matlab sunspot მონაცემები):

```
load sunspot.dat
year=sunspot(:,1);
wolf=sunspot(:,2);
```

იპოვეთ მზის ლაქების ვარიაციის პერიოდები;

დავალება 10.4.

ჩაატარეთ Matlab chirp მონაცემების სპექტრალური ანალიზი:

```
x = chirp(t,0,1,Fs/6);
```

დავალება 10.5.

ჩაატარეთ მოცემული ფუნქციის სპექტრალური ანალიზი:

$$f(t) = \sin(t) \exp(0.2t) + \cos(3t) \exp(-0.1t) + 0.2 \text{ rand}$$

1. Plot $E(\omega)$ vs. ω
2. Plot $\text{Re}(F)$ vs. $\text{Im}(F)$
3. $G1(\omega) = F(\omega) + \text{random (physical) noise in phase};$
4. $G2(\omega) = F(\omega) + \text{random (physical) noise in amplitude};$
5. plot on single graph: $f(t), f1(t), f2(t)$

Random noise: Max amplitude = $0.2 \cdot \text{Max}(F)$;

$F = A \cdot \exp(i \cdot \phi)$: A - Amplitude; ϕ - phase;

$f1 = \text{ifft } G1$

$f2 = \text{ifft } G2$

6. subplot(2,2,1); plot $f'(t)$ vs $f(t)$
7. subplot(2,2,2); plot $f1'$ vs $f1$
8. subplot(2,2,3); plot $f2'$ vs $f2$

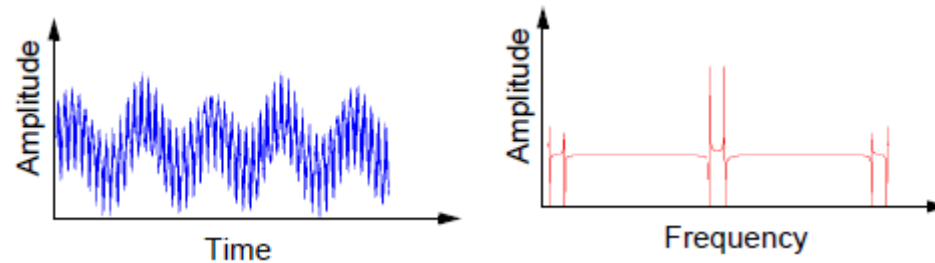
დანართი: ანალიზური ფუნქციის ფურიე გარდაქმნების ცხრილი

ფუნქცია	ფურიე სახე
$f(x)$	$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$
1	$\sqrt{2\pi} \cdot \delta(\omega)$
$\delta(x)$	$\frac{1}{\sqrt{2\pi}}$
e^{iax}	$\sqrt{2\pi} \cdot \delta(\omega - a)$
$\cos(ax)$	$\sqrt{2\pi} \cdot \frac{\delta(\omega - a) + \delta(\omega + a)}{2}$
$\cos(ax^2)$	$\frac{1}{\sqrt{2a}} \cos\left(\frac{\omega^2}{4a} - \frac{\pi}{4}\right)$
x^n	$i^n \sqrt{2\pi} \delta^{(n)}(\omega)$

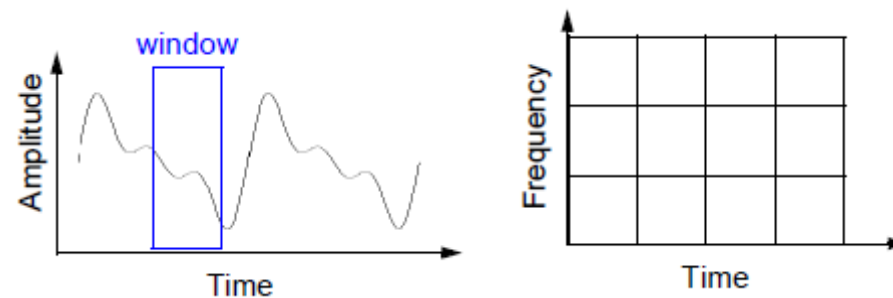
ფუნქცია	ფურიე სახე
$\frac{1}{x}$	$-i \sqrt{\frac{\pi}{2}} \operatorname{sgn}(\omega)$
$ x ^\alpha$	$\frac{-2 \sin(\pi\alpha/2) \Gamma(\alpha + 1)}{\sqrt{2\pi} \omega ^{\alpha+1}}$
$\operatorname{sgn}(x)$	$\sqrt{\frac{2}{\pi}} \frac{1}{i\omega}$
$u(x)$	$\sqrt{\frac{\pi}{2}} \left(\frac{1}{i\pi\omega} + \delta(\omega) \right)$
$\sum_{n=-\infty}^{\infty} \delta(x - nT)$	$\frac{\sqrt{2\pi}}{T} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{T}\right)$
$\log x $	$-\frac{\sqrt{\pi/2}}{ \omega } - \sqrt{2\pi} \gamma \delta(\omega)$
$(\mp ix)^{-\alpha}$	$\frac{\sqrt{2\pi}}{\Gamma(\alpha)} u(\pm\omega) (\pm\omega)^{\alpha-1}$

ვეივლეტ (wavelet) გარდაქმნა

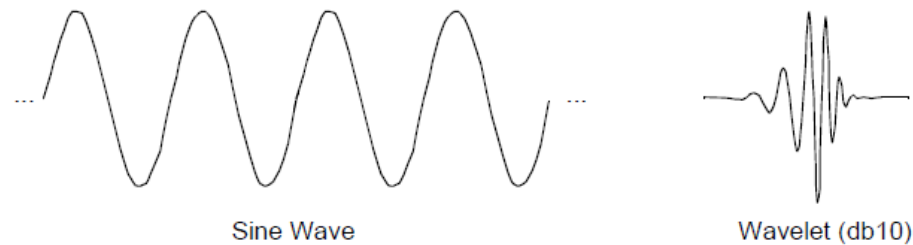
ფურიე გარდაქმნა აჩვენებს სიგნალის სპექტრს (სიხშირეებს), მაგრამ არა იმას დროის თუ რა მოწვევით ჩანს ეს სიხშირე.

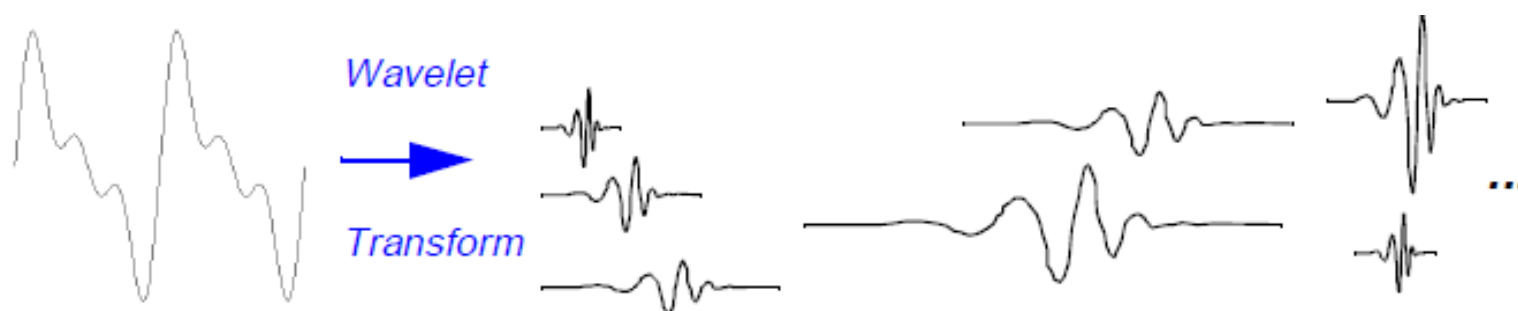


Short time Fourier analysis (Gabor)



ვეივლეტები:





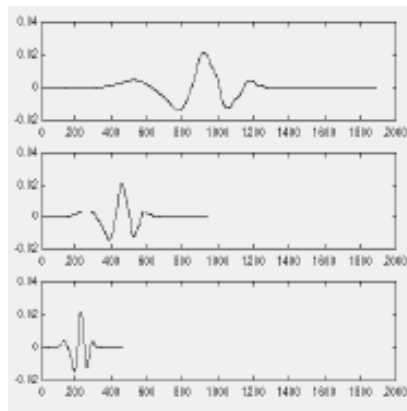
უწყვეტი ვეივლეტ გარდაქმნა (WT)

scale ვეივლეტის მასშტაბი
position ვეივლეტის პოზიცია

$$C(\text{scale}, \text{pos}) = \int_{-\infty}^{+\infty} f(t) \Psi(\text{scale}, \text{pos}, t) dt$$

$$[W_{\psi} f](a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \overline{\psi\left(\frac{x-b}{a}\right)} f(x) dx$$

მასშტაბი:

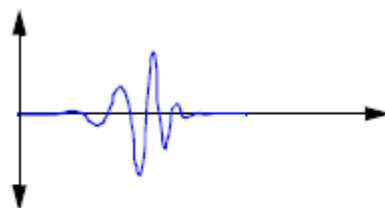


$$f(t) = \psi(t) \quad ; \quad a = 1$$

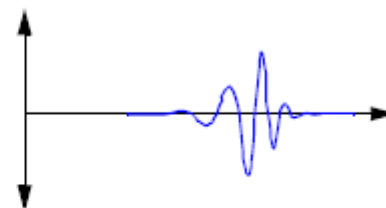
$$f(t) = \psi(2t) \quad ; \quad a = \frac{1}{2}$$

$$f(t) = \psi(4t) \quad ; \quad a = \frac{1}{4}$$

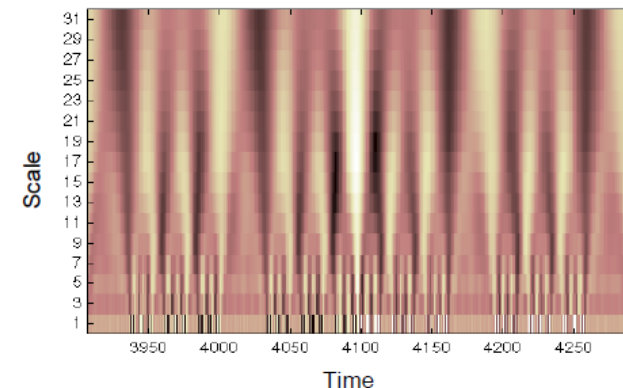
პოზიცია:



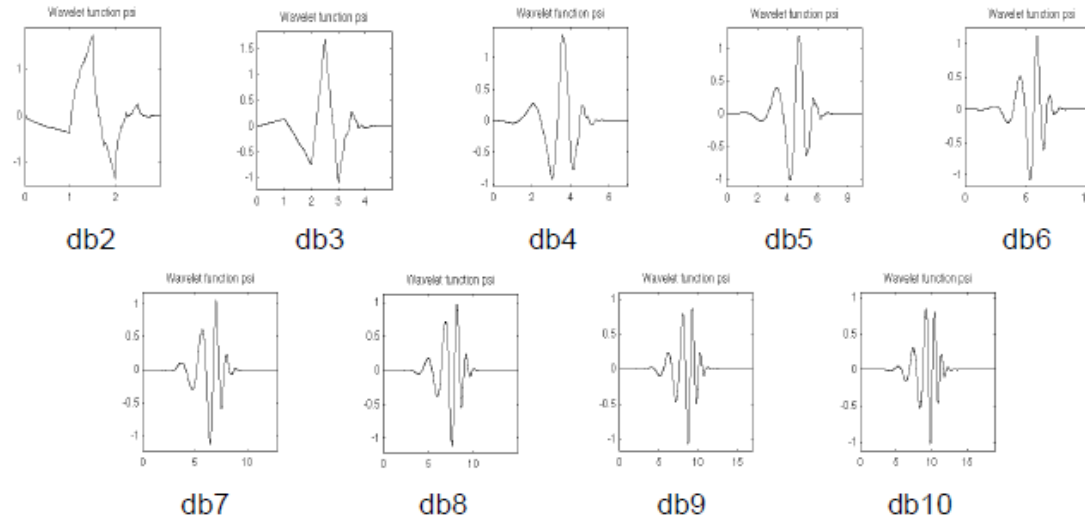
Wavelet function
 $\psi(t)$



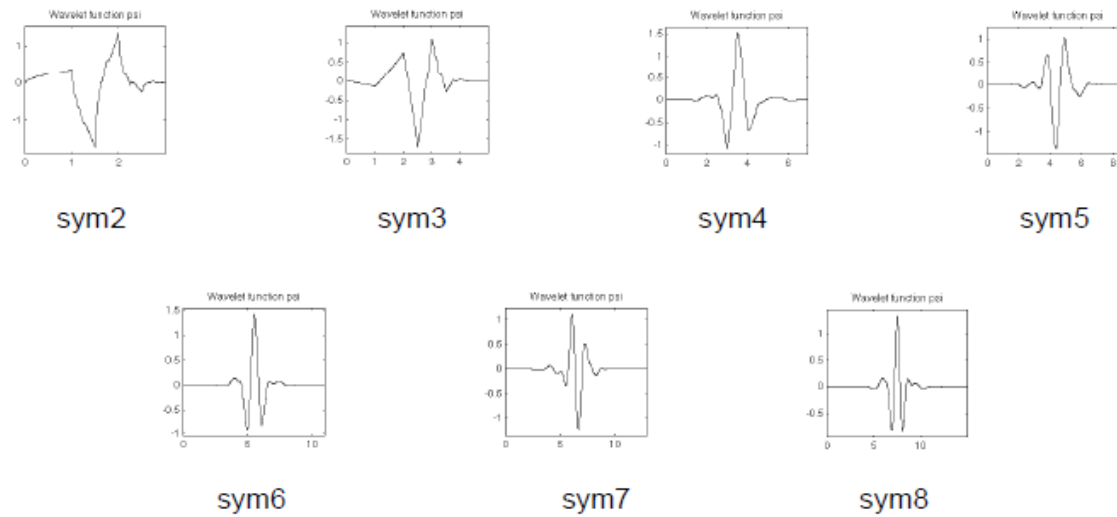
Shifted wavelet function
 $\psi(t-k)$



Daubechies wavelets (dbN)



Symlets



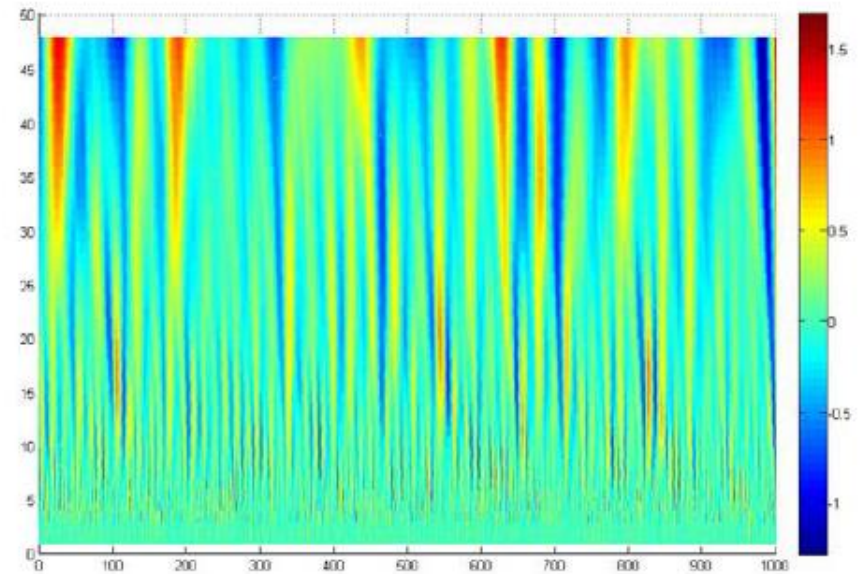
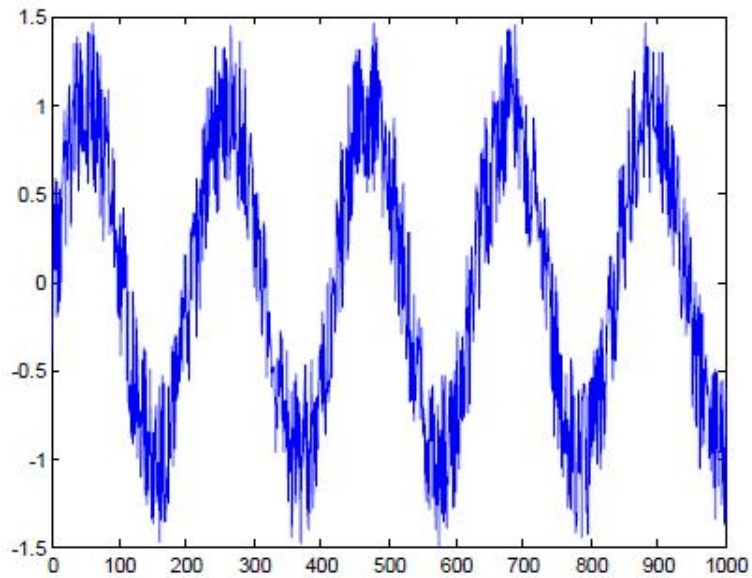
Wavelet მაგალითი

```
load noissin;

figure(1);
plot(noissin);

c = cwt(noissin,1:48,'db4');

figure(2);
mesh(c);
view(0,90);
```

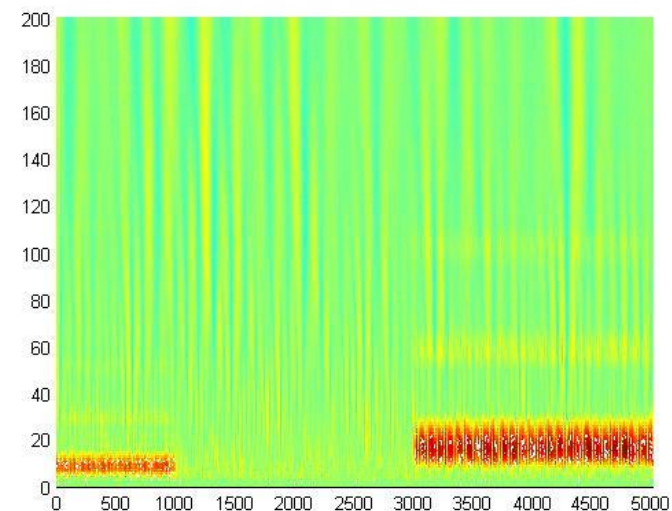
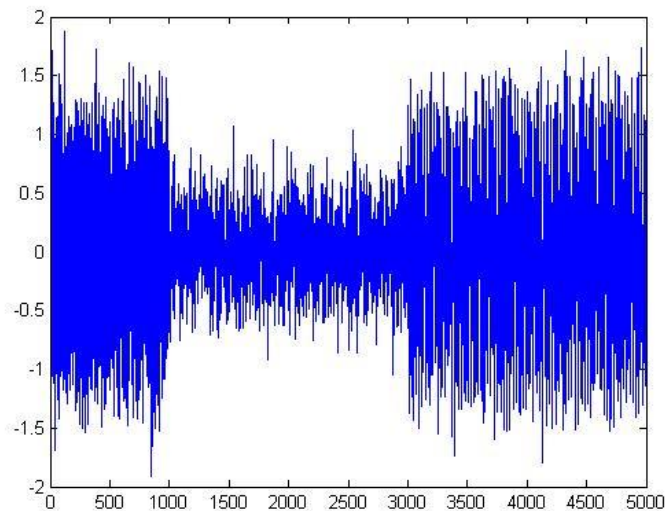


Matlab: wscalogram

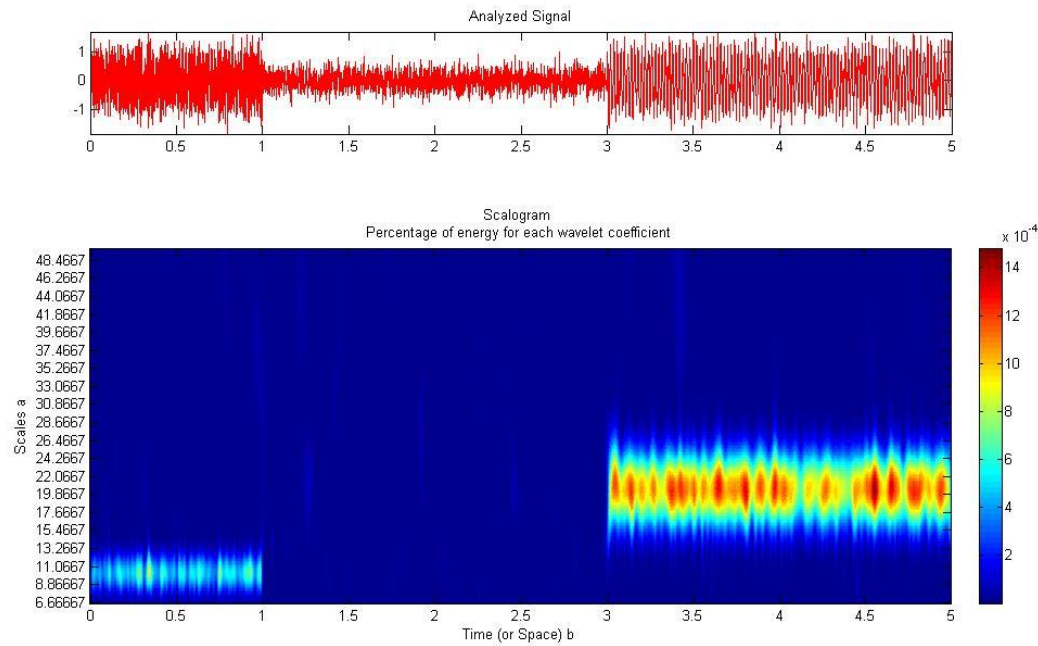
```
Fs = 1000;
fc = centfrq('cmor1-1');
% a = fc/(freq*dt)

freqrange = [20 150];
scalerange = fc./(freqrange*(1/Fs));
t = linspace(0,5,5e3);
x = cos(2*pi*100*t).*(t<1)+cos(2*pi*50*t).*(3<t)+0.3*randn(size(t));
```

```
C=cwt(x',1:200,'db4');
mesh(C);
view(0,90);
```



```
scales = scalerange(end):0.2:scalerange(1);
Coeffs = cwt(x,scales,'cmor1-1');
SCImg = wscalogram('image',Coeffs,'scales',scales,'ydata',x,'xdata',t);
```



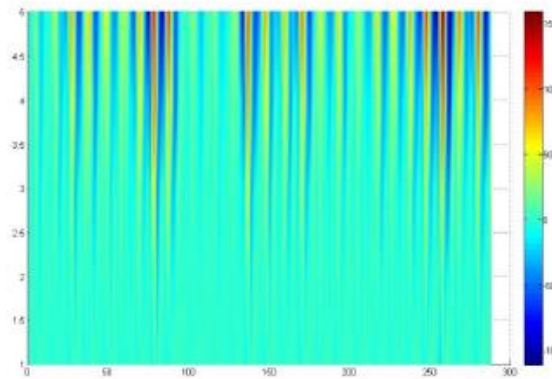
```
load sunspot.dat
year=sunspot(:,1);
wolf=sunspot(:,2);

SCALES=50;

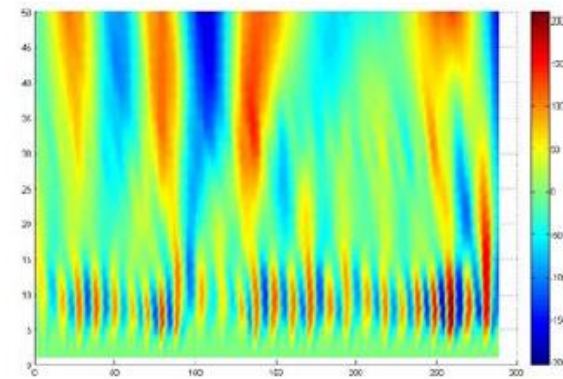
c = cwt(wolf,1:SCALES,'db4');

surf(c);
view(0,90);
shading interp;
```

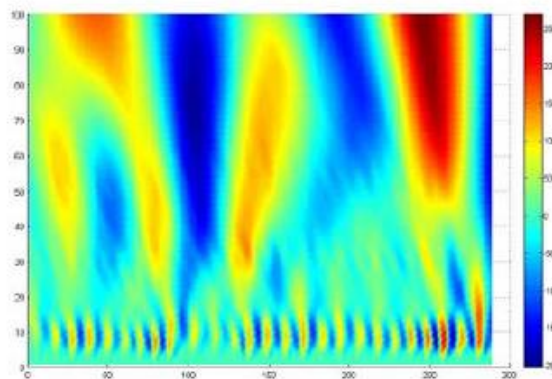
10



50



100



200

