L4. Kontinuirana wavelet transformacija (CWT)

Uvod

Na vježbama ćete se upoznati s implementacijom CWT-a u MATLAB-u. Implementacija je dio MATLAB-ovog Wavelet Toolbox paketa. Primijenit ćete CWT analizu na stvarnim signalima i vizualizirati rezultate. Usporedit ćete ih s STFT-om.

CWT

Kontinuirana wavelet transformacija (CWT) definirana je sljedećim izrazom:

$$X(\tau, a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} x(t)\, \psi^*\left(\frac{t-\tau}{a}\right) dt. \qquad (1)$$

Primjeri wavelet funkcija:

$$\begin{aligned}
&\text{kompleksni Morletov wavelet: } \psi(t) = C \cdot e^{-\frac{t^2}{2\sigma^2}} e^{-j\omega_0 t}, \\
&\text{realni Morletov wavelet: } \psi(t) = e^{-\frac{t^2}{2}} \cos 5t, \\
&\text{sombrero (\textit{eng. Mexican hat}): } \psi(t) = C\left(1-t^2\right) e^{-\frac{t^2}{2}}, \\
&\text{kompleksni Shannonov wavelet: } \psi(t) = C \cdot \frac{\sin(\omega_b t)}{\omega_b t} e^{-j\omega_0 t},
\end{aligned} \qquad (2)$$

i mnoge druge.

Za numeričko izračunavanje kontinuirane wavelet transformacije u MATLAB-u služi funkcija cwt(). Funkcija je dio Wavelet Toolboxa. Slijedi skraćeni opis korištenja (help cwt):

CWT Real or Complex Continuous 1-D wavelet coefficients.

CWT Real or Complex Continuous 1-D wavelet coefficients.

COEFS = CWT(S, SCALES, 'wname') computes the continuous wavelet coefficients of the vector S at real, positive SCALES, using wavelet whose name is 'wname'. The signal S is real, the wavelet can be real or complex.

COEFS = CWT(S, SCALES, 'wname', 'plot') computes and, in addition, plots the continuous wavelet transform coefficients.

Vektor S predstavlja signal, vektor SCALES sadržava sve skale za koje želimo izračunati CWT, a 'wname' je naziv wavelet funkcije iz biblioteke.

Podatke o grupama i obiteljima wavelet funkcija raspoloživih u Wavelet Toolboxu možemo dobiti funkcijom waveinfo(). Podržane su sljedeće grupe:

1. Crude wavelets
2. Infinitely regular wavelets.
3. Orthogonal and compactly supported wavelets.
4. Biorthogonal and compactly supported wavelet pairs.
5. Complex wavelets

U svakoj grupi realizirano je više obitelji wavelet funkcija. Korisnik može definirati i vlastite wavelet obitelji pomoću wavemngr().

Diskretne aproksimacije wavelet funkcija možemo dobiti MATLAB funkcijom wavefun(). Primjer (za wavelete iz grupa 1 i 5):

```
[psi, xval] = wavefun('wname');          % 'wname' označava kratki naziv waveleta
figure, plot(xval, psi);             % prikaz wavelet funkcije
```

Iz grupe 1. na vježbama koristimo: realni Morletov wavelet ('morl'). Iz grupe 5. koristimo kompleksni Morletov wavelet ('cmorFb-Fc'). Ove wavelet funkcije su odabrane zbog sličnosti s STFT funkcijama razlaganja, ako je lokalizirajući otvor Gaussov. Više podataka o implementaciji i svojstvima odabranih waveleta možemo dobiti funkcijom waveinfor():

```
waveinfo('morl')
waveinfo('cmor')
```

Implementacija realnog Morletovog waveleta $\psi(t) = e^{-\frac{t^2}{2}} \cos 5t$ u MATLAB-u ne dozvoljava slobodne parametre: širina Gaussovog otvora je fiksirana, a isto vrijedi i za frekvenciju kosinusne funkcije. Kod analize realnih signala nameće nam se pitanje na kojoj frekvenciji je centar energije od $\psi(t)$? Moramo primijeniti izraz:

$$\omega_\Psi^+ = \frac{\int_0^{+\infty} \omega \left|\Psi(\omega)\right|^2 d\omega}{\int_0^{+\infty} \left|\Psi(\omega)\right|^2 d\omega}.$$

Za različite iznose skale znamo da će centri iznositi $\omega_c = \omega_\Psi / a$. Uočimo da $\omega_\Psi \neq 5$. U MATLAB-u se centralna frekvencija $f_c = \omega_c / 2\pi$ za različite iznose skale numerički može dobiti funkcijom scal2frq().

Implementacija kompleksnog Morletovog waveleta $\psi(t) = C \cdot e^{-\frac{t^2}{2\sigma^2}} e^{-j\omega_0 t}$ u MATLAB-u ima prednost jer dozvoljava podešavanje širine Gaussovog otvora i frekvencije kompleksne harmonijske funkcije:

Definition: a complex Morlet wavelet is
*cmor(x) = (pi*Fb)^{-0.5}*exp(2*i*pi*Fc*x)*exp(-(x^2)/Fb)*
depending on two parameters:
Fb is a bandwidth parameter
Fc is a wavelet center frequency
Wavelet name   cmor"Fb"-"Fc"

Parametri koji određuju širinu i frekvenciju zadaju se uz ime wavelet funkcije. Primjeri: 'cmor1-1', 'cmor32-1', cmor'8-2' i sl.

Lako je pokazati da je za kompleksni wavelet centar energije točno je jednak $\omega_\Psi = \omega_0$, odnosno u implementaciji u MATLAB-u $f_c = Fc$ Hz.

To su razlozi zbog kojih očekujemo da će primjena kompleksnog Morletovog waveleta u analizi realnih signala, a u svrhu usporedbe sa STFT-om imati prednost nad ostalim obiteljima wavelet funkcija.

Glazbena ljestvica

Na vježbama ćemo analizirati glazbene signale i pokušati identificirati tonove, odnosno notni zapis.

Koje frekvencije pripadaju tonovima glazbene ljestvice? Pitanje je samo naizgled jednostavno. Vrlo detaljne odgovore možete pronaći na Internetu, korištenjem poznatih pretraživača uz ključne riječi „frequencies of the music scale“.

Još je Pitagora ponudio rješenje za glazbenu ljestvicu. Polazište mu je bilo da je oktava omjer 2:1, a kvinta 3:2, pa je zaključio da se svi tonovi mogu dobiti iz omjera (3/2)*(3/2)*(1/2) = 9/8. Iako je njegov rezultat bio sjajan, ipak je uključivao pogrešku i sljedeća je oktava počinjala tonom otprilike 1.36% više frekvencije od potrebne.

U srednjem vijeku Vicenzo Galilei (1606.-1649.) je korigirao ljestvicu na omjere 1, 9/8, 5/4, 4/3, 3/2, 5/3, 15/8, 2. Ona je zvučala bolje od Pitagorine, ali su kompozicije odsvirane s različitim početnim tonom zvučale drugačije. Čak je trebalo promijeniti po koju notu. Takva se ljestvica u engleskoj literaturi često sreće pod nazivom *just intonation*.

Zarlino (1558.) i Stevin (1608.) predlažu podjelu oktave u 12 polutonova s identičnim omjerima frekvencija, a za popularizaciju novog sustava vrlo zaslužan je J. S. Bach (1685.-1753.).

Danas prevladava navedeni sustav istih omjera (*eng. equitempered*), gdje omjer frekvencija susjednih polutonova iznosi $\sqrt[12]{2} = 1.0594631...$ Naravno, u akordima (suzvučjima više tonova) se više ne javljaju razlomački odnosi frekvencija, kako je to bio slučaj kod prethodnih ljestvica.

Po definiciji, frekvencija tona **A** u trećoj oktavi (A3) iznosi **440 Hz**.

Ostale polutonove ljestvice izračunavamo kao:

Calculation for Equal-Tempered tuning [A3 = 440Hz]							
Hertz		**Octave=0**	**Octave=1**	**Octave=2**	**Octave=3**	**Octave=4**	**Octave=5**
0	A	55.000	110.000	220.000	440.000	880.000	1,760.000
1	A#/Bb	58.270	116.541	233.082	466.164	932.328	1,864.655
2	B	61.735	123.471	246.942	493.883	987.767	1,975.533
3	C	65.406	130.813	261.626	523.251	1,046.502	2,093.005
4	C#/Db	69.296	138.591	277.183	554.365	1,108.731	2,217.461
5	D	73.416	146.832	293.665	587.330	1,174.659	2,349.318
6	D#/Eb	77.782	155.563	311.127	622.254	1,244.508	2,489.016
7	E	82.407	164.814	329.628	659.255	1,318.510	2,637.020
8	F	87.307	174.614	349.228	698.456	1,396.913	2,793.826
9	F#/Gb	92.499	184.997	369.994	739.989	1,479.978	2,959.955
10	G	97.999	195.998	391.995	783.991	1,567.982	3,135.963
11	G#/Ab	103.826	207.652	415.305	830.609	1,661.219	3,322.438
12	A	110.000	220.000	440.000	880.000	1,760.000	3,520.000