

ZPRACOVÁNÍ SIGNÁLU V MATLABU

Jan Černocký, FIT VUT Brno

1 Generování signálu, filtrace, frekvenční analýza

V této části budeme pracovat s uměle vygenerovaným signálem. Na vzorkovací frekvenci $F_s = 16000$ Hz vygenerujeme směs dvou harmonických signálů na frekvencích $f_1 = 440$ Hz a $f_2 = 1$ kHz. Využijeme přepočtu skutečných frekvencí na normované:

$$f' = \frac{f}{F_s}$$

a toho, že cosinusovka se generuje:

$$x[n] = C \cos(2\pi f' n)$$

Signál bude trvat 2 sekundy. Pro poslouchání budeme ukládat do souborů WAV a využijeme Linuxovský příkaz `play`; Matlab v Linuxu Bohu žel neumí přehrávat s libovolnou vzorkovací frekvencí :-)

```
% generovani signalu - smes 2 cosinusovek, trvani 2 sec.
Fs = 16000; f1 = 440; f2=1000;
n = 0:15999;
s = 0.49*cos(2 * pi * f1/Fs * n) + 0.49*cos(2 * pi * f2/Fs * n);
plot (s); wavwrite(s,16000,16,'s.wav');
```

Budeme frekvenčně analyzovat prvních 1024 vzorků tohoto signálu. Všimněte si, že se díváme jen na interval frekvencí do $F_s/2$ (horní polovina je symetrická a nezajímá nás).

```
% frekvencni analyza - vybereme 1024 vzorku
x = s(1:1024);
f = (0:511) / 1024 * Fs;
X = fft(x); X = X(1:512);
subplot (211); plot(f,abs(X)); subplot (212); plot(f,angle(X));
```

1.1 Filtrace

Z tohoto signálu chceme vybrat pouze složku na 1 kHz. V Matlabu máme pro návrh IIR filtrů k dispozici výkonnou funkci `ellip` (s pomocnou `ellipord`). Podívejte se do jejich helpů. Navrhovaný filtr bude pásmová propust. Pozor: zatímco všude se normalizuje vzorkovací frekvencí, u matlabovských funkcí pro návrhy filtrů normalizujeme Nyquistovou frekvencí ($F_s/2$) !

```
% navrh filtru tak, aby vybral 1 kHz:
Fs2 = Fs / 2; % timto budeme normalizovat pro navrh filtru
Wp = [900/Fs2 1100/Fs2];
Ws = [800/Fs2 1200/Fs2];
Rp = 3; Rs = 20;
[N, Wn] = ellipord (Wp, Ws, Rp, Rs)
[B,A] = ellip(N,Rp,Rs,Wn)
```

Prohlédneme frekvenční charakteristiku a rozložení nul a pólů navrženého filtru:

```
% koukani na frekv char
freqz(B,A,512,16000);
% a na poly a nuly
zplane (B,A)
```

... a budeme filtrovat signál:

```
% filtrovani signalu
sf = filter (B,A,s);
plot (sf); wavwrite(sf,16000,16,'sf.wav');
```

Úkoly:

1. Výstupní signál si přehrajte a frekvenčně analyzujte (opět 1024 vzorků). Zbavili jsme se 440 Hz ?
2. Ve spektru výstupu možná objevíte zbytek čáry na 440 Hz. Ověřte, že tato čára je utlumena přesně podle frekvenční charakteristiky filtru
 - odečtěte z $H(f)$ přenos na 440 Hz v dB.
 - převedte z dB zpět na normální číslo (jsme v amplitudách, takže $10^{dB/20}$).
 - vynásobte tímto číslem výšku původní čáry. Tuto velikost byste měli vidět na 440 Hz ve spektru výstupu.

2 Frekvenční analýza řeči

Nahrajte si řečový signál na $F_s = 8000$ Hz. Ustředněte (do proměnné `sm`) a rozdělte jej na rámce o délce 160 vzorků bez překrytí (matice `sr`). Pokud nefunguje Vaše funkce, použijte příloženou `frame.m`.

V signálu vyberte jeden znělý rámec (nejlepší je zoomovat na waveformu signálu, odečíst číslo vzorku, podělit 160 a přičíst 1 – tak dostanete příslušné číslo rámce). Vyberte jej (např. pro rámec 197):

```
x = sr(:,197);
```

a podle přednášek proveďte frekvenční analýzu:

1. se zobrazením plného pásma od 0 do F_s .
2. se zobrazením od 0 do $F_s/2$.
3. s doplněním nulami (zero-padding) pro zvýšení rozlišení ve frekvenci.
4. spektrální hustotu výkonu.
5. její logaritmickou verzi.

Na přednášce se dozvíte, že spektrum řeči je určeno záklaním tónem (jemná struktura) a postavením artikulačního ústrojí (hrubá struktura). Vidíte je na spočítaných spektrech?

3 Hand-made spektrogram

Na minulém cvičení jste pouze jako 'black-box' spouštěli funkci `specgram`. Nejedná se o nic jiného než o odhad spektrální hustoty výkonu pro jednotlivé rámce – viz `help specgram`. Budeme schopni si takový spektrogram spočítat ručně:

```
Nr = size(sr,2); % pocet ramcu
% alokujeme pro specgram (jednotliva spektra budou ve sloupcich)
sg = zeros (512,Nr);
for n = 1:Nr,
    x = sr(:,n); %plot(x); pause;
    X = fft([x' zeros(1,1024-160)]); X = X(1:512); Gdft= 1/160 *abs(X) .^ 2;
    sg(:,n) = Gdft';
end
imagesc (10 * log10(sg)); axis xy;
```

Úkoly

1. Poslední řádka je zobrazení – zjistěte, proč musíme použít funkci `axis`.
2. Do druhého obrázku vypočtete spektrogram pomocí matlabovské funkce:
`figure(2); specgram(sm)`
Zjistěte, zda se liší. Pokud ano, zkuste si "pohrát" s dalšími parametry funkce `specgram` (viz `help`) tak, aby byl stejný (hodnoty na osách neřešte...).

4 Wah-wah

je populární efekt pro elektronické nástroje - jedná se o přeladitelnou pásmovou propust. Úkolem je navrhnout efekt, který periodicky přeladuje propust od 500 Hz do 1500 Hz. Perioda má trvat 1 s. Nejprve navrhne jako test propust se středem v 1 kHz a podíváme se na její kmitočtovou charakteristiku:

```
% navrh jednoho filtru: - pasmova propust od 900 do 1100, 3db ripple,
% 20db atten - nic moc parametry :-)
Fs=8000; Fs2 = Fs / 2; % timto budeme normalizovat pro navrh filtru
Wp = [900/Fs2 1100/Fs2];
Ws = [800/Fs2 1200/Fs2];
Rp = 3; Rs = 30;
[N, Wn] = ellipord (Wp, Ws, Rp, Rs)
[B,A] = ellip(N,Rp,Rs,Wn)
freqz(B,A,512,8000);
```

Tímto filtrem vyfiltrujte celý signál `sm` a poslechněte si jej. Můžete zkusit i frekvenční analýzu některých rámců pomocí FFT.

Úkoly:

1. Zbytek úlohy je vyřešen v souboru `wahwah.m`. Nejprve tento soubor spusťte “přes mrtvoly”, zobrazte a poslechněte si výstupní signál `yy`.
2. Všimněte si, jak je řešen výpočet střední frekvence pro každý rámeček. Uložte si hodnoty `cf` do vektoru a prohlédněte si jej.
3. Proč je pro indexování správných koeficientů filtru použita funkce `'modulo'` ?
4. “Lepení” rámečků k výslednému signálu pomocí `yy = [yy y']`; je značně časově náročné, protože Matlab pokaždé přealokovává... Navrhněte něco chytřejšího.