

ISS Projekt, autor: Patrik Holop (xholop01)

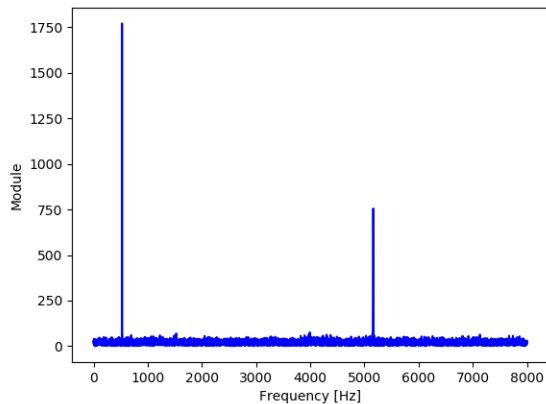
1. Základné informácie o súbore

Celý projekt bol vypracovaný v jazyku Python. Pomocou funkcie *soundfile.read* som zistil nasledujúce údaje.

Počet vzorkov (N)	16000
Vzorkovacia frekvencia (Fs)	16000 [Hz]
Dĺžka nahrávky	1[s]

2. Fourierova transformácia

Pomocou funkcie *scipy.fftpack.fft* som získal hodnoty spektra a zobrazil vektor od 0 do polovice vzorkovacej frekvencie (8000 [Hz])



3. Maximum modulu spektra

Najvyššiu hodnotu modulu zo spektra som získal pomocou funkcie *max*.

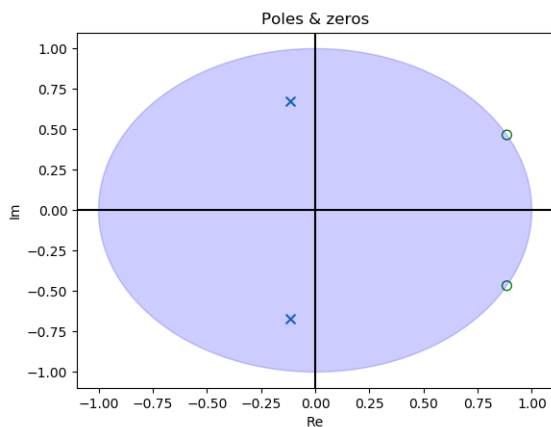
Frekvencia, na ktorej je max modulu spektra = 526 Hz

4. IIR Filter

Pre zadaný filter bolo treba zistiť prenosovú funkciu pomocou Z-tranformácie

$$H(z) = \frac{0.2324z^2 - 0.4112z + 0.2324}{z^2 + 0.2289z + 0.4662}$$

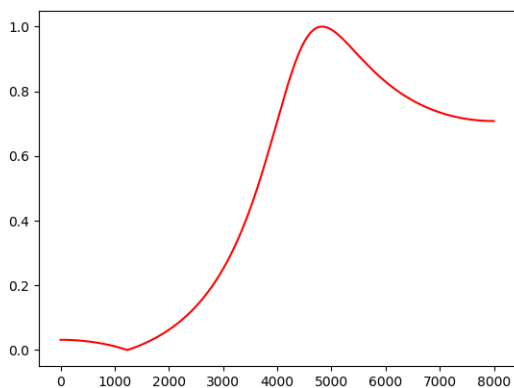
Z nej som pomocou funkcie *signal.zplane* získal graf núl a pólov.



Všetky póly sa nachádzajú vo vnútri jednotkovej kružnice, takže filter je **stabilný**.

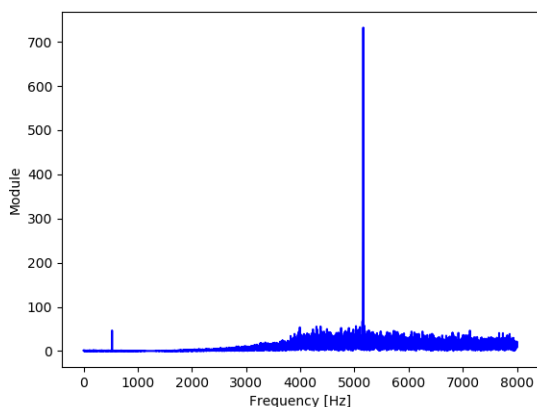
5. Kmitočtová charakteristika filtra

Kmitočtovú charakteristiku filtra je vidno na obrázku nižšie získanú pomocou funkcie *signal.freqz*. Os X reprezentuje hodnoty frekvencie v Hz a Y magnitúdu. Z grafu je vidno, že prepúšťa vyššie frekvencie a nižšie tlmí. Jedná sa o **hornú priepusť**.



6. Spektrum po filtrovaní

Po aplikovaní filtra pomocou funkcie *signal.lfilter* som získal hodnoty spektra rovnako ako v úlohe č. 2 nad filtrovaným signálom. Je vidno, že filter prepustil predovšetkým vyššie frekvencie.

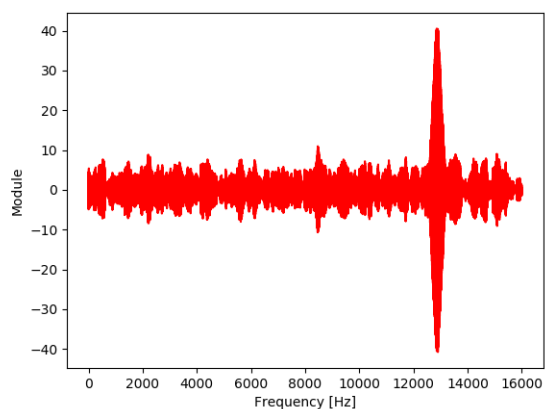


7. Maximum modulu po filtrovaní

Maximum modulu po filtrovaní sa zmenilo, lebo nižšiu frekvenciu filter neprepustil.

Frekvencia, na ktorej je maximum modulu spektra = 5164 [Hz]

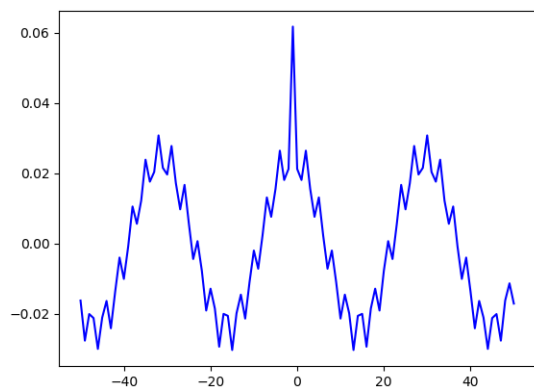
8. Detekovanie pridaného šumu



Pomocou funkcie `numpy.correlate` som hľadal koreláciu so signálom $[1,1,-1,-1]$ (80 sekvencií). V grafe je vidno výraznú koreláciu v konkrétnej časti signálu.

Pridaný šum začína počnúc 12857 [Hz] a 0.8035 [s].

9. Autokorelácia



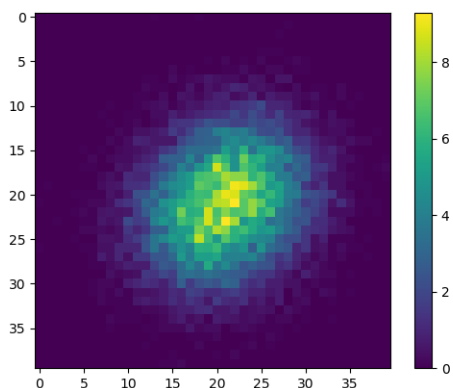
Pre vychýlený odhad koeficientov autokorelačnej funkcie v rozsahu k od -50 do 50 (os X) som použil funkciu `signal.correlate`.

10. Autokorelačný koeficient

Z vyššie vykreslenej funkcie som jednoduchým adresovaním v poli získal

R[10]: -0.011068

11. Združená funkcia hustoty rozdelenia pravdepodobnosti



Vykreslený graf som získal pomocou funkcie `plt.hist2d`, ktorý rozdelil graf na 40x40 štvorcov a získal jednotlivé početnosti pre každý štvorek na základe pola pôvodných dát $X[n]$ a posunutých $X[n+10]$. Tieto hodnoty ešte bolo potrebné vydeliť vzorkovacou frekvenciou a obsahom štvorca, aby sme získali výsledné hodnoty.

12. Overenie funkcie

Integrál sme museli aproximovať sumou. Súčet objemov častí pod grafom po vynásobení obsahom štvorčeka bol rovný 1.

Jedná sa o správnu združenú funkciu hustoty rozdelenia pravdepodobnosti.

13. Autokorelačný koeficient

Hodnotu som vypočítal na základe vzťahu $R[10] = \frac{\sum_{n=0}^{n-1} x[n] \cdot x[n+10]}{n} = -0.011068$

Výsledný autokorelačný koeficient je rovnaký.

Bonusová úloha

Pri riešení bonusovej úlohy som skúšal niekoľko variánt prístupu k úlohe. Najjednoduchším spôsobom bolo vytvorenie náhodného generovania koeficientov v intervaloch $(b_i - \epsilon, b_i + \epsilon)$, kde ϵ bola povolená chyba, osvedčila sa hodnota 0.02. Tento postup nevedol k celkovej energii chyby nižšej ako 820.

Druhý prístup bolo vytvoriť jednoduchý *Perceptron*, ktorý bude konvergovať k minimu funkcie chyby. Trénovacie dáta boli prekrývajúce sa šesticie zo vstupného súboru. Hoci bol výpočet rýchly a jednoznačne konvergoval (testoval som niekoľko počiatočných konfigurácií), dosiahnutá chyba 870 taktiež nebola uspokojivá.

Tretí prístup bolo vytvorenie jednoduchého evolučného algoritmu, ktorý v každej generácii generoval 500 riešení, ktoré boli vyhodnocované na základe predošlých generácií. Dosiahnutá chyba bola 780, čo bol pokrok, ale nedostatočný.

Posledná verzia využíva algoritmus *LPC* (Linear prediction filter coefficients), ktorý sa snaží dosiahnuť podobný výsledok ako samotné zadanie $x[n] = -a[2]x[n-1] - a[3]x[n-2] - \dots - a[p+1]x[n-p]$. Napriek tomu, že sa jedná o predimplementovanú funkciu, jednoznačne porazila vyššie spomenuté algoritmy s chybou 738.

Výsledný filter:

$$X[n] = 0.204957 \cdot X[n-1] + 0.1373999 \cdot X[n-2] + 0.335082258 \cdot X[n-3] + 0.043341 \cdot X[n-4] - 0.100567 \cdot X[n-5]$$

Porovnanie s ostatnými filtermi:

Popis	Rovnica	Chyba	Výsledok
Iba nuly	$X[n]=0$	988.6	Môj filter je o 25,35% lepší
Posledný prvok	$X[n]=X[n-1]$	1298.2	Môj filter je o 43,08% lepší
Priemer posledných 2	$X[n]=0.5X[n-1]+0.5X[n-2]$	1024	Môj filter je o 27,93% lepší
Môj filter	viz. výsledný filter	738	