

# PROJEKTNI ZADATAK 1

Predmet: Osnove Algoritama i Struktura DSP1

Student: Milan Kapetanović RA 184/2018

# CILJ PROJEKTA

Cilj ovog projekta je bio da koristeći FIR i IIR filtre očistimo dva audio signala od šumova.

Dobili smo dva ulazna signala koji sadrže govor koji je pomešan sa šumom.

Projekat je podeljen u 2 kontrolne tačke:

## -Kontrolna tačka 1

U ovoj kontrolnoj tački imali smo za cilj da na prvom signalu primenimo FIR filter i posmatramo promene u signalu kroz različite dužine filtra.

## -Kontrolna tačka 2

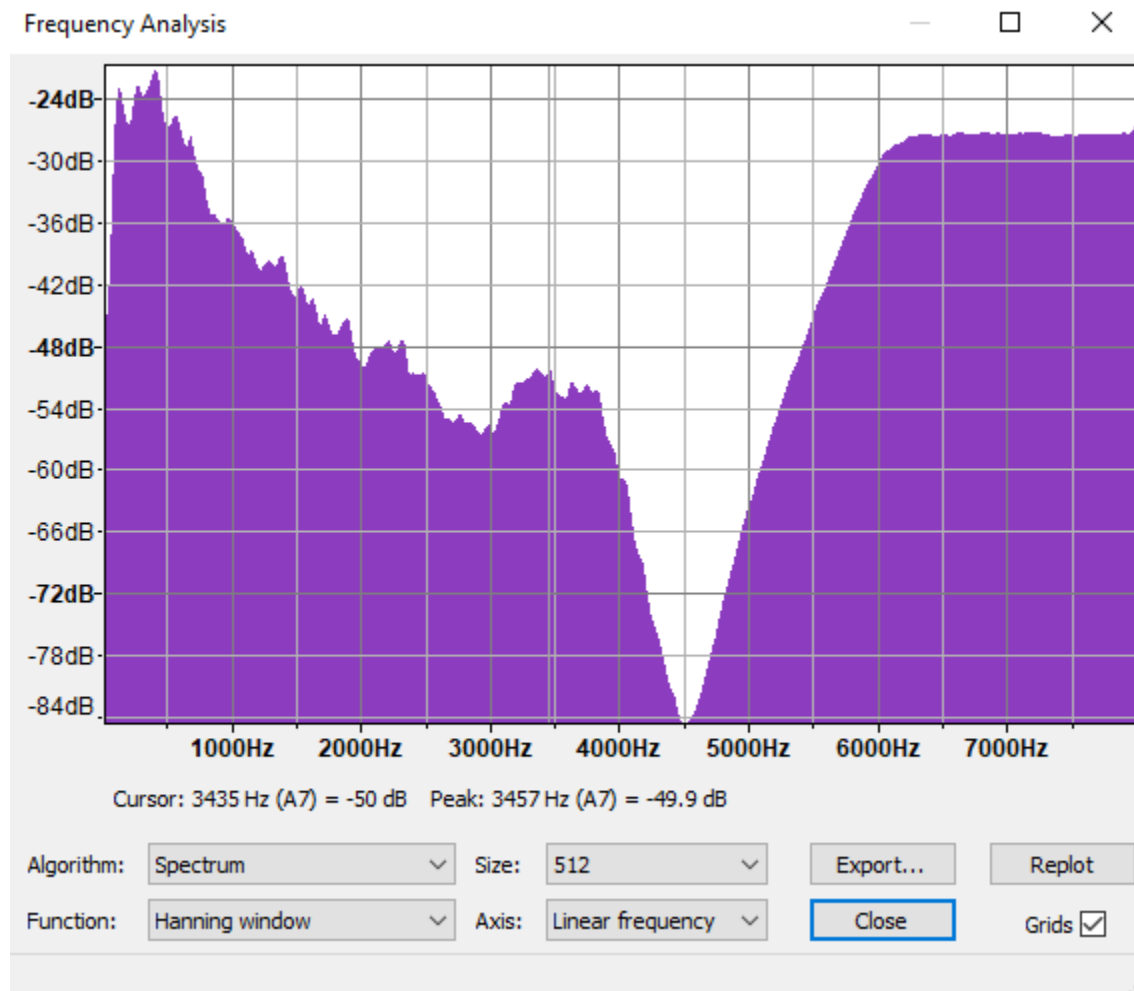
U ovoj kontrolnoj tački imali smo za cilj da na drugom signalu primenimo IIR filter i posmatramo promene u signalu kroz različite primene filtra.

Nakon što su obe kontrolne tačke ispunjene dobijamo 2 nova audio signala u kojima su ili umanjeni šumovi (početni zadaci u obe kontrolne tačke) ili potpuno uklonjeni (poslednji zadaci u obe kontrolne tačke).

# Zadatak 1

- U ovom zadatku bilo je potrebno da iz signala “*signal1.wav*” pronađemo spektralni podopseg i njegovu gornju i donju granicu.

To smo uradili tako što ubacimo *signal1.wav* u Audacity i idemo **Analyze -> Plot Spectrum** i posmatramo odakle dokle ide signal.



Ovde vidimo da nas glavni signal ide od **0Hz** do **4500Hz** nakon toga se pojavljuje šum.  $f_1 = 0 \text{ Hz}$   $f_2 = 4500 \text{ Hz}$ .

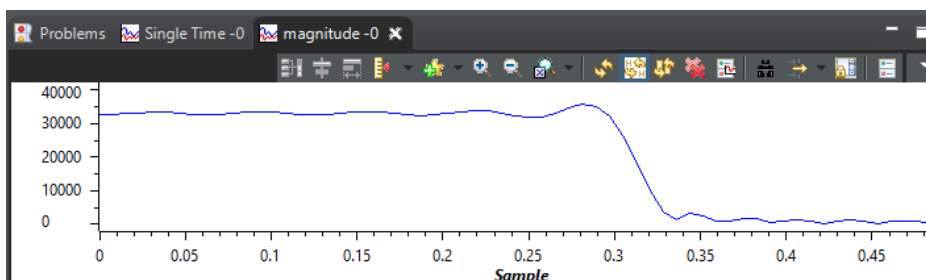
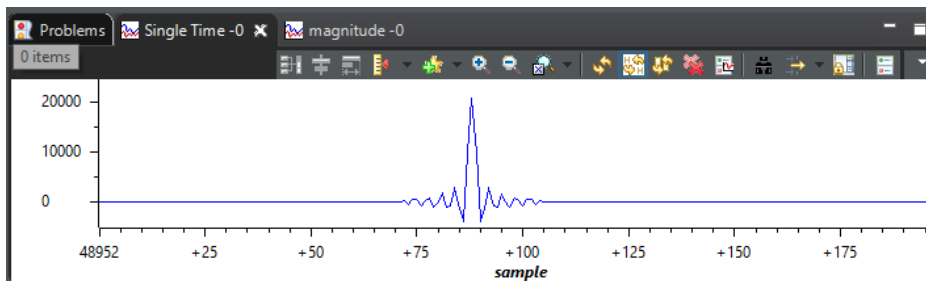
# Zadatak 2

U ovom zadatku bilo je potrebno da koristeći program **WinFilter** izgenerišemo koeficijente za FIR filter.

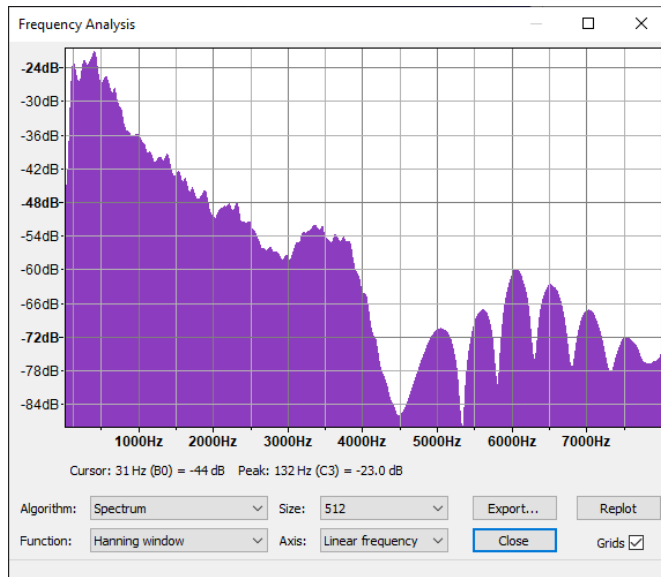
Te koeficijente ubacujemo u **coeff.h** i implementiramo ih u **main.c**

```
for(j = 0; j < AUDIO_IO_SIZE; j++)
{
    //TO DO: Call filter routine here. Use coefficients from coeffs.h
    outputBufferL[j] = fir_basic(inputBufferL[j], filter1_coeffs, history0, F1_COEFFS);
    outputBufferR[j] = fir_circular(inputBufferR[j], filter1_coeffs, history0, F1_COEFFS, p_state);
}
```

U ovoj for petlji koristimo za levi buffer **fir\_basic** dok za desni koristimo **fir\_circular**. Prosledjujemo koeficijente iz **coeff.h**. Nakon toga dobijamo Impulsni i Frekvencijski odziv.

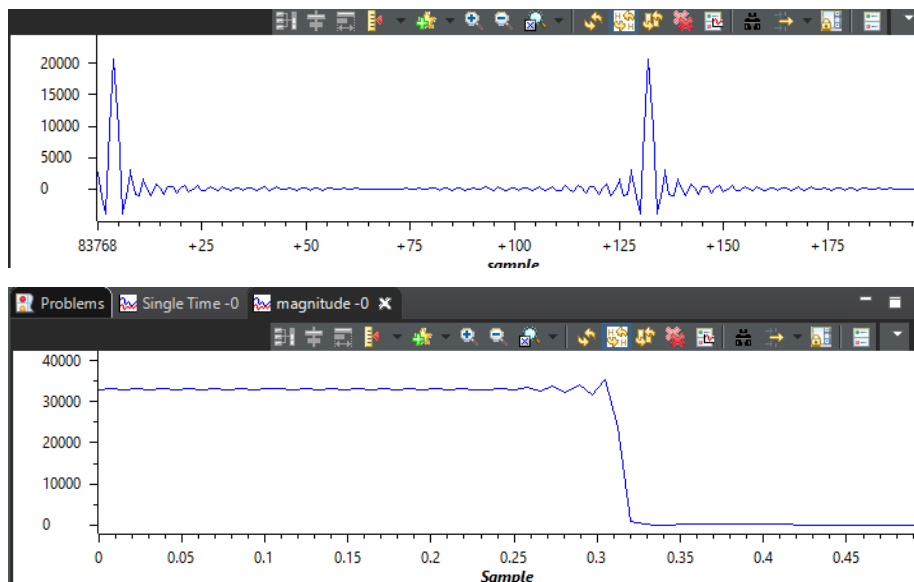


I dobijamo obrađeni signal “*Zadatak2.wav*”, u kojem je šum delimično uklonjen.

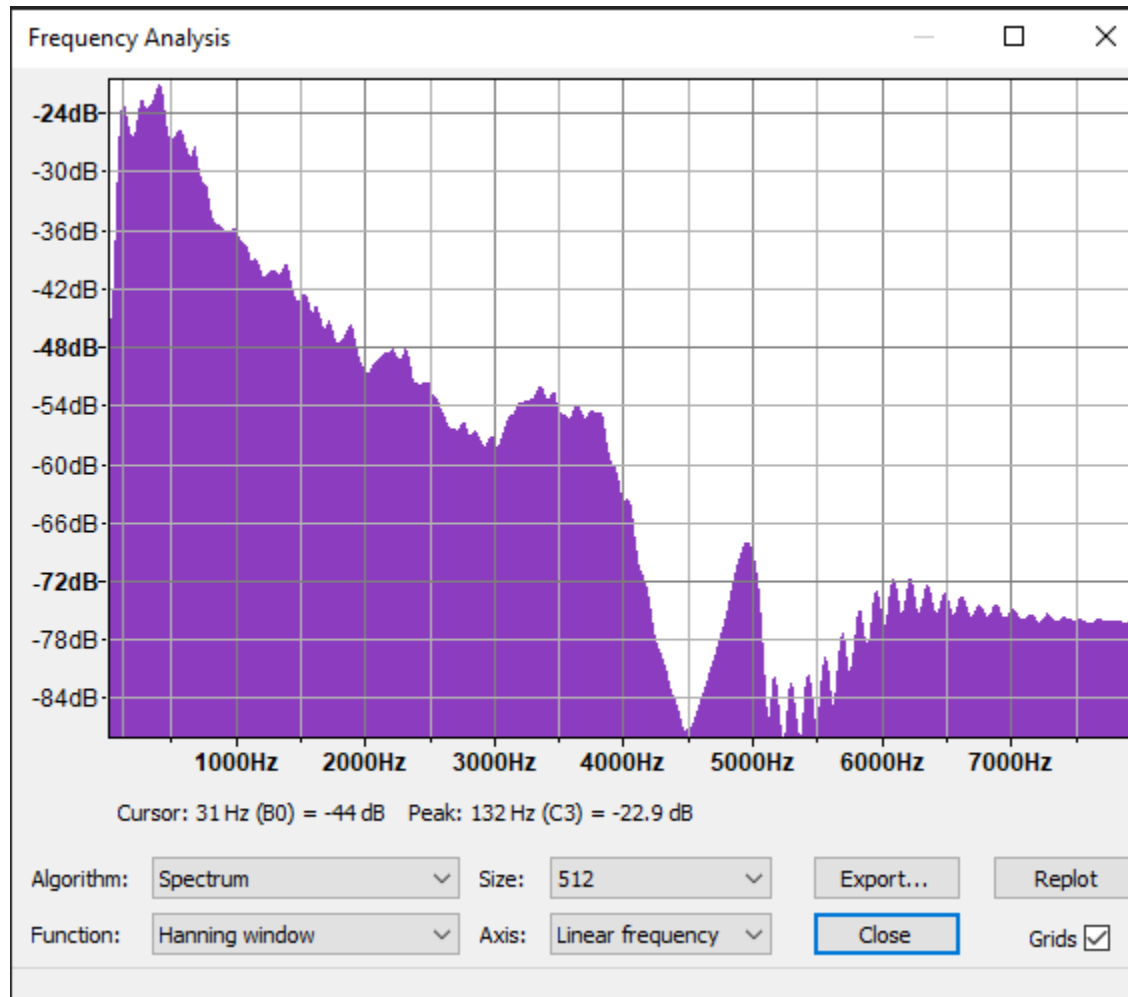


## Zadatak 3

Ovaj zadatak je u principu isti kao i prošli samo što povećavamo broj koeficijenata (**N tap**) u **WinFiltru** sa **33** (zadatak 2) na **121**. Umesto **F1\_COEFFS** pišemo **F2\_COEFFS**. Dobijamo malo drugačiji Impulsni i Frekvencijski odziv.

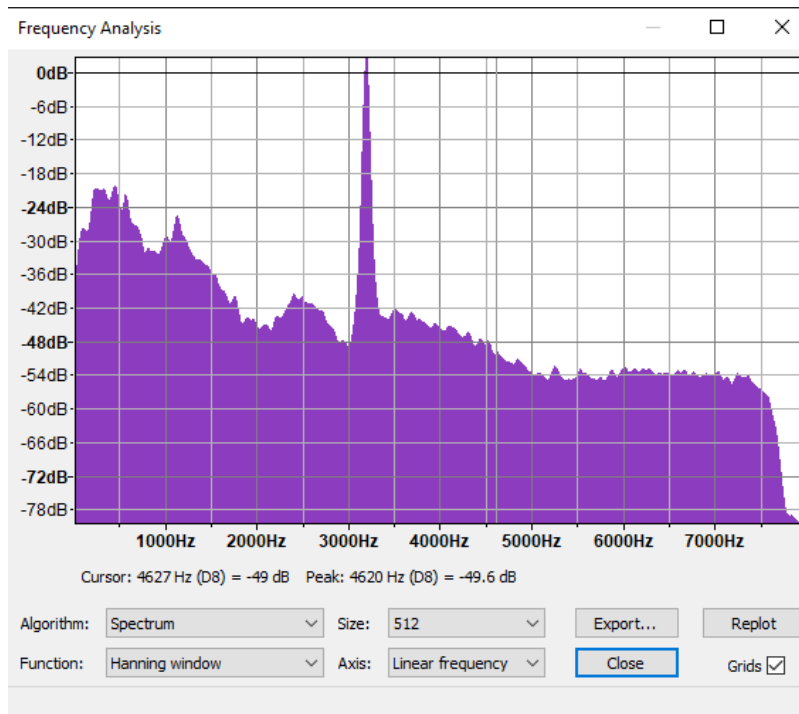


Nakon toga se kreira signal “*Zadatak3.wav*” i koristeći **Plot Spectrum** možemo videti da rastom broja koeficijenata šum se bolje potiskuje.



# Zadatak 4

Zadatak 4 traži da pronađemo frekvenciju sinusnog signala u audio signalu “*signal2.wav*”. Ponovo koristimo **Plot Spectrum**.



Najviši signal na grafu je sinusni signal i u našem slučaju on iznosi  $f1 = 3200\text{Hz}$ .

# Zadatak 5

U zadatku se prvo okrećemo formuli koja nam je zadata:

$$H(z) = \frac{1 - 2 \cos(2\pi f) z^{-1} + z^{-2}}{1 - 2r \cos(2\pi f) z^{-1} + r^2 z^{-2}}$$

Gde su gornji elementi  $a_0$ ,  $a_1$  i  $a_2$ . Donji elementi su  $b_0$ ,  $b_1$  i  $b_2$ .  $f$  nam predstavlja odnos izmedju  $f_1$  i  $f_s$ , koje nam je 16000Hz(**Sample Frequency**). Nakon toga delimo  $a_1$  i  $b_1$  sa 2 i množimo svaki element sa 32767. Dobijamo koeficijente

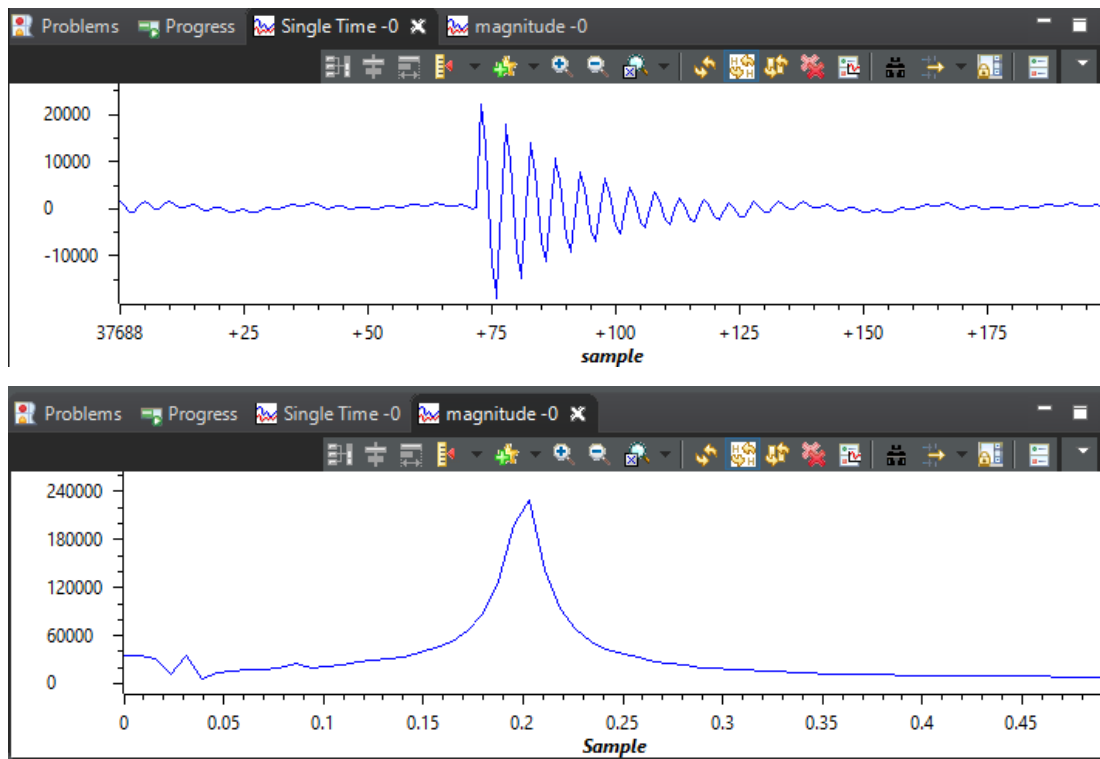
32767,  
-10145,  
32767,  
32767,  
-9638,  
29572

Nakon uvrštavanja u **coeff.h** pravimo novu petlju koristeći funkcionalnosti IIR filtra.

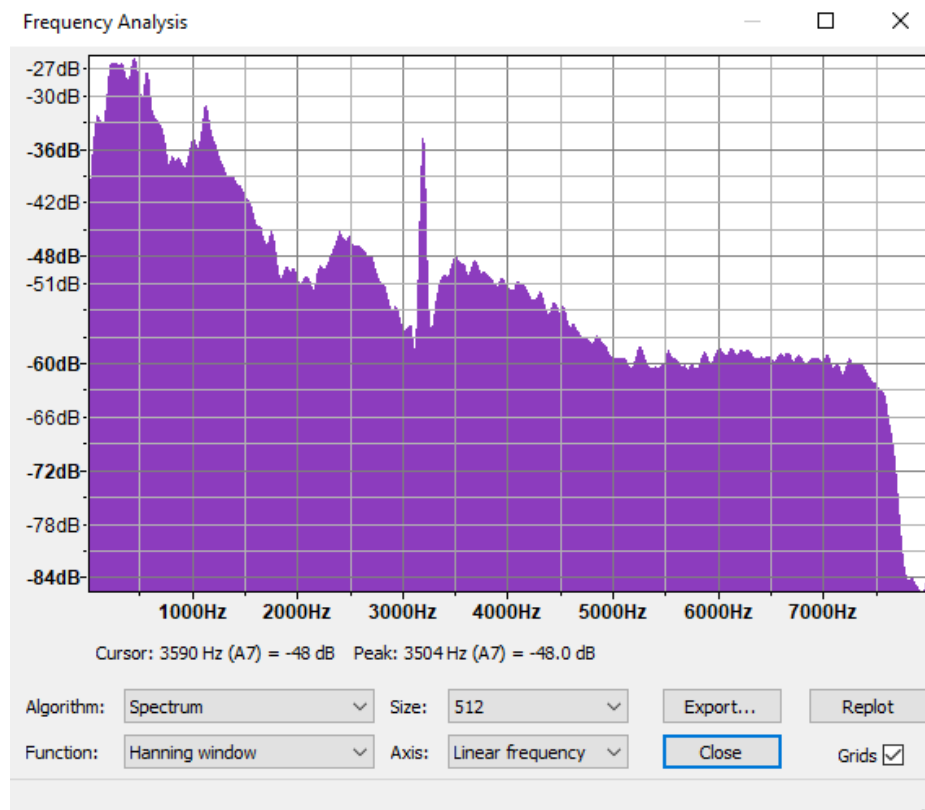
```
//Zadatak 5
for(j = 0; j < AUDIO_IO_SIZE; j++)
{
    //TO DO: Call filter routine here. Use coefficients from coeffs.h
    outputBufferL[j] = second_order_IIR(inputBufferL[j], iir_notch_coeffs, historyI, historyO);
}
```



Nakon ovog koda dobijamo Impulsne i Frekvencijske odzive:



I na kraju dobijamo obrađeni signal “*Zadatak5.wav*” i kada pogledamo **Plot Spectrum**:



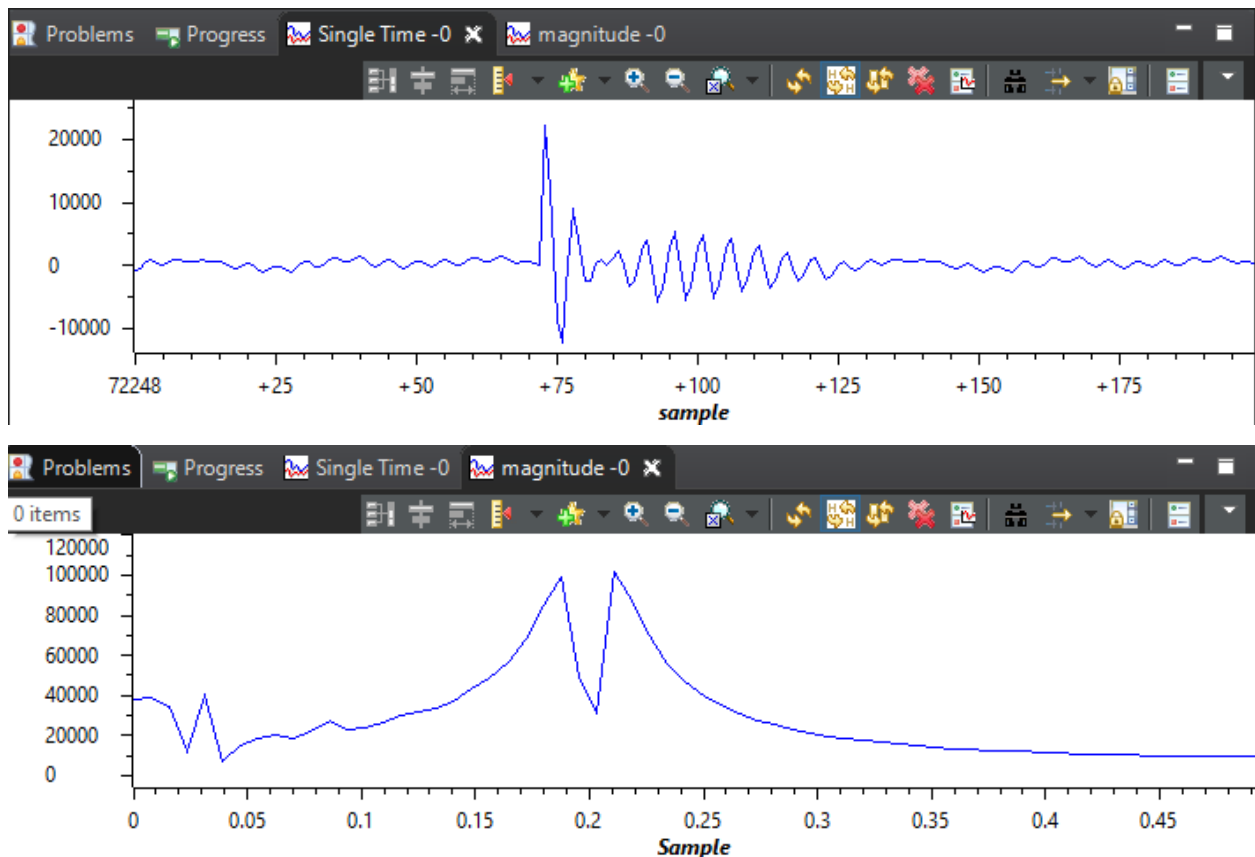
Vidimo da je šum većinski uklonjen.

# Zadatak 6

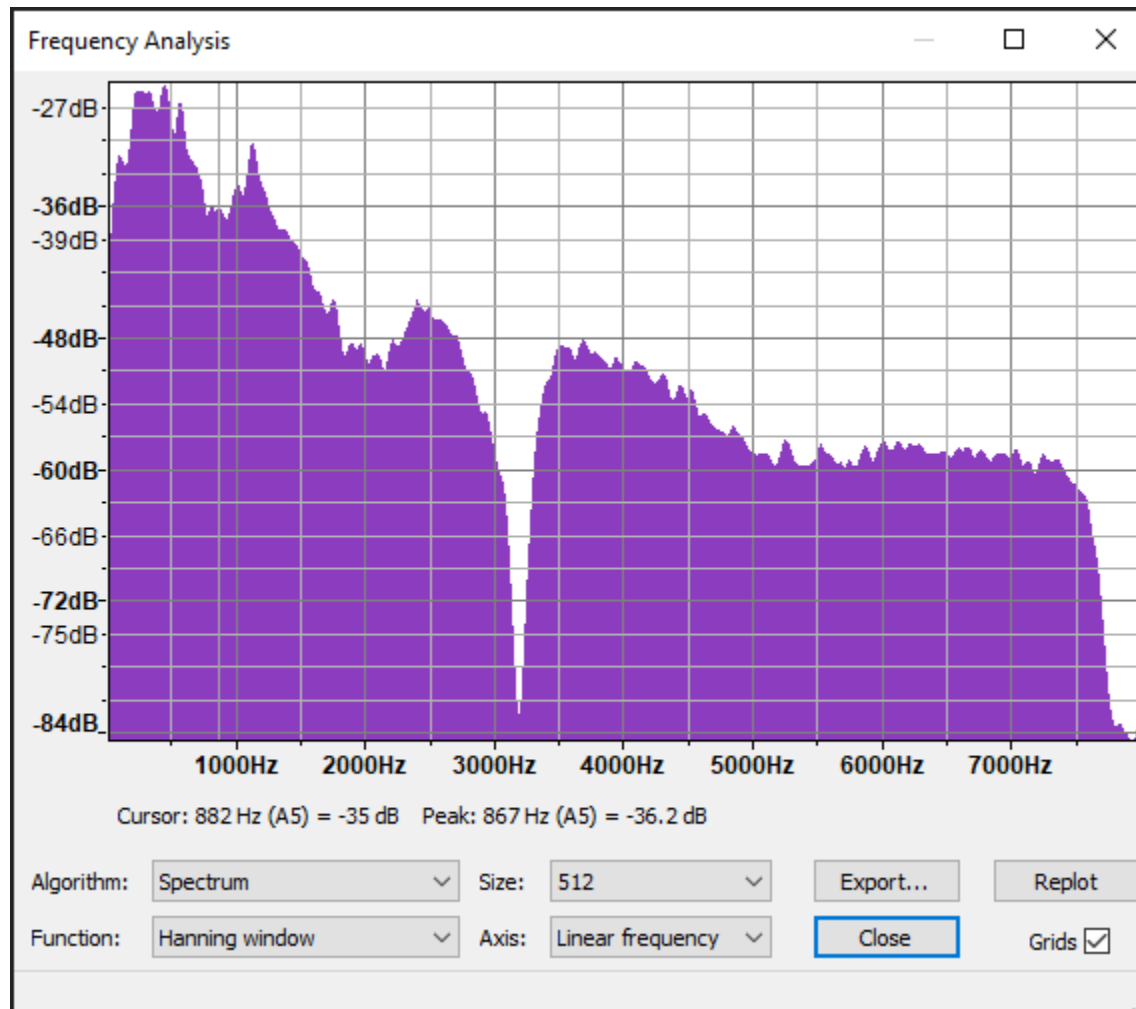
Zadatak zahteva da primenimo 3 **IIR filtra** drugog reda kako bismo dobili filter 6. reda sa istim koeficijentima kao u petom zadatku. To smo ostvarili ovako:

```
//Zadatak 6
for(j = 0; j < AUDIO_IO_SIZE; j++)
{
    //TO DO: Call filter routine here. Use coefficients from coeffs.h
    sesti_zadatak_priv1[j] = second_order_IIR(inputBufferL[j], iir_notch_coeffs, x_history, y_history);
    sesti_zadatak_priv2[j] = second_order_IIR(sesti_zadatak_priv1[j], iir_notch_coeffs, x_history1, y_history1);
    outputBufferL[j] = second_order_IIR(sesti_zadatak_priv2[j], iir_notch_coeffs, x_history2, y_history2);
}
```

Uveli smo 2 dodatne promenljive kako bismo najjednostavnije povezali 3 puta **IIR filter** drugog reda. Što dovodi do *ovakvih* odziva:



Nakon toga dobijamo obrađeni signal “Zadatak6.wav” i proveravajući **Plot Spectrum** vidimo:



Šum je **potpuno uklonjen** a ostale frekvencije su ostale netaknute.