MATLAB Signal Processing Report

Lambros Savva

January 2026

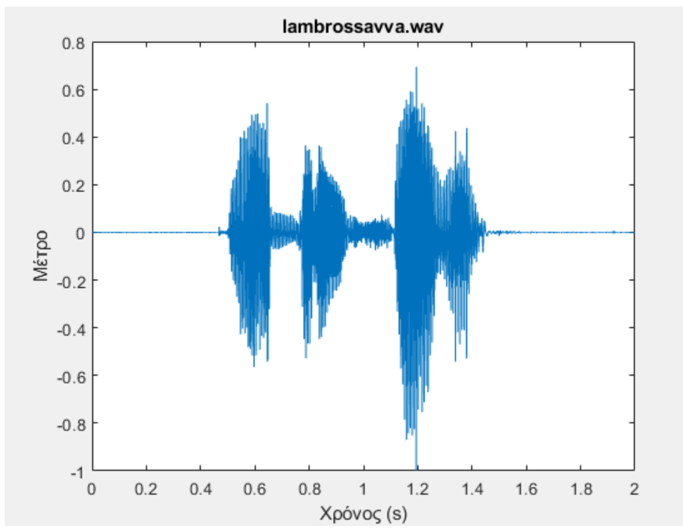
1 Voice Signal Analysis in Time and Frequency Domains

1.1 Voice Recording

Using the audiorecorder command with a sampling rate of 8000Hz at 16 bits and 1 channel, I recorded my name. Using the getaudiodata and sound commands, I listened to the recording (for confirmation). All of this was done in the ”audiowrite.m” file.

1.2 Signal Plotting

Using the plot command, I plotted the complete signal.



1.3 Energy

I calculated the energy as the convolution of the squared signal with the Hamming window using the conv command.

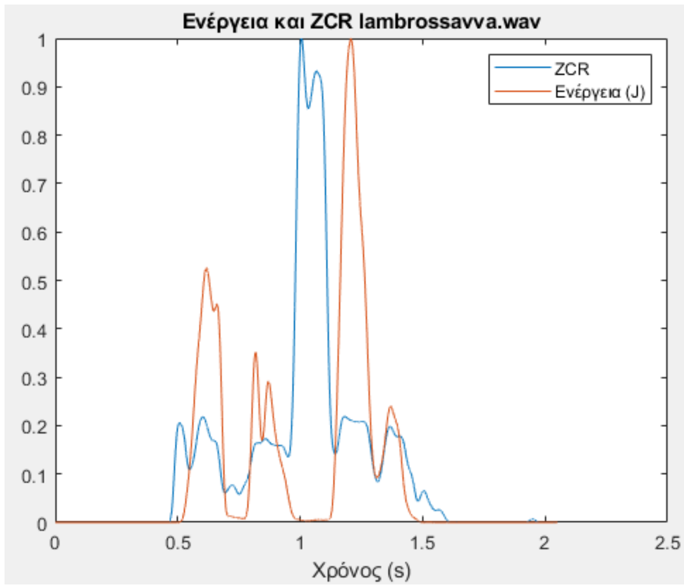
1.4 Zero Crossing Rate (ZCR)

First, I ”clipped” the signal, meaning that when the signal was quiet and oscillating around 0 without much energy, I set it equal to 0. This was done to prevent the ZCR from spiking between words.

Using the convolution of a square window of M samples (of the ”cleaned” signal) and the Hamming window, with M=400, I calculated the ZCR.

1.5 Energy & ZCR Plotting

After normalizing the signals to take values in [0,1], I plotted them on the same diagram.



1.5.1 Observation

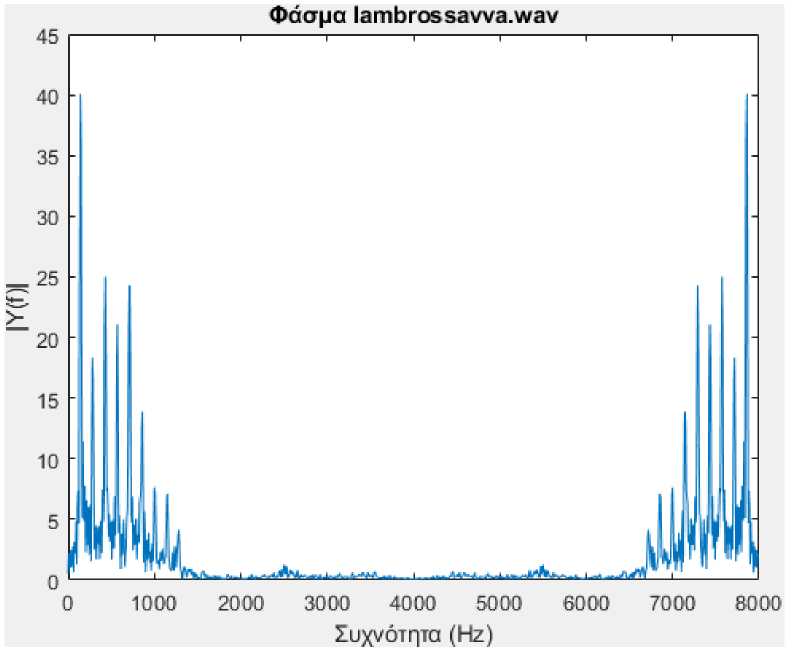
In voiced phonemes, the energy is greater than the ZCR, while in unvoiced phonemes, the opposite is true.

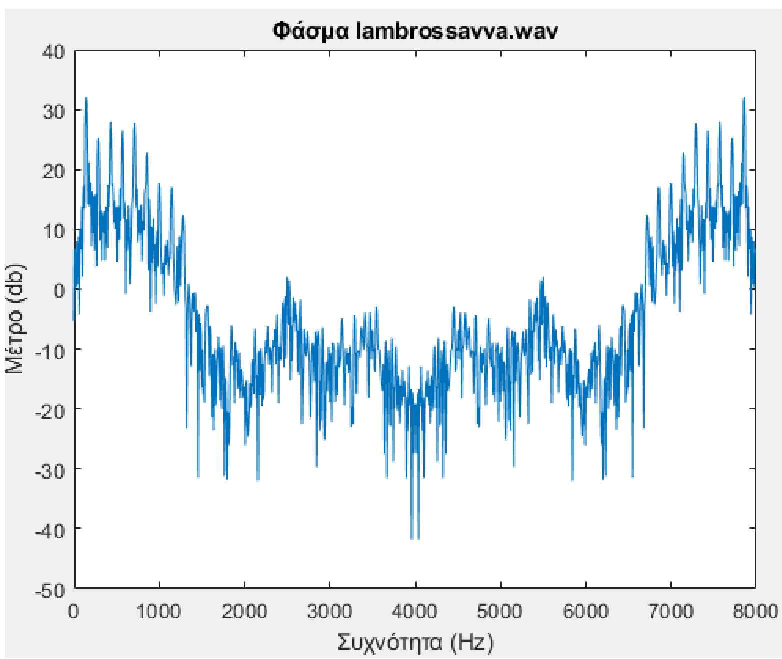
1.6 50ms Window

I selected the phoneme ”a” (L”a”mpros Savva) and visually found a period of $T = 7 \times 10^{-3}\ s$.

1.7 DFT

Using the Discrete Fourier Transform (DFT) with 1024 samples, I found and plotted the transform of the signal segment on both normal and logarithmic scales. By measuring the distance between peaks, I confirmed that the oscillation frequency was approximately $140 - 148\ Hz$.


Φάσμα lambrossavva.wav
|Y(f)|
Συχνότητα (Hz)
0
1000
2000
3000
4000
5000
6000
7000
8000
45
40
35
30
25
20
15
10
5


Φάσμα lambrossavva.wav
Μέτρο (db)
Συχνότητα (Hz)
40
30
20
10
0
-10
-20
-30
-40
-50
1000
2000
3000
4000
5000
6000
7000
8000

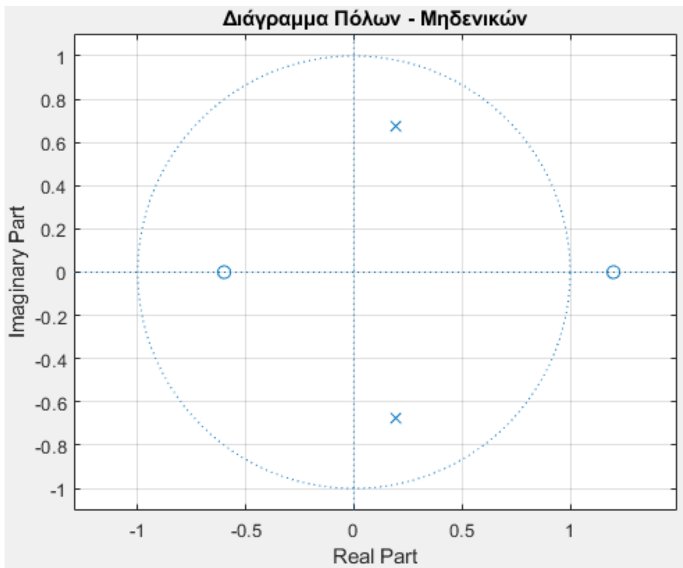
2 Filter Design and Application to Music Signals

2.1 Band-Pass Filter Design

2.1.1 Filter Design

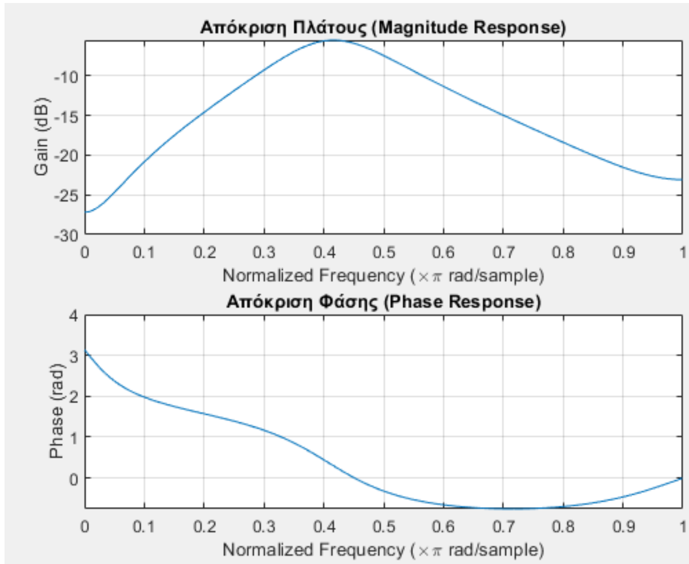
I designed a filter and represented it with a pole-zero diagram using the zplane command. It has poles at positions $0.196 \pm 0.672i$ and zeros at positions $\{1.2, -0.6\}$.

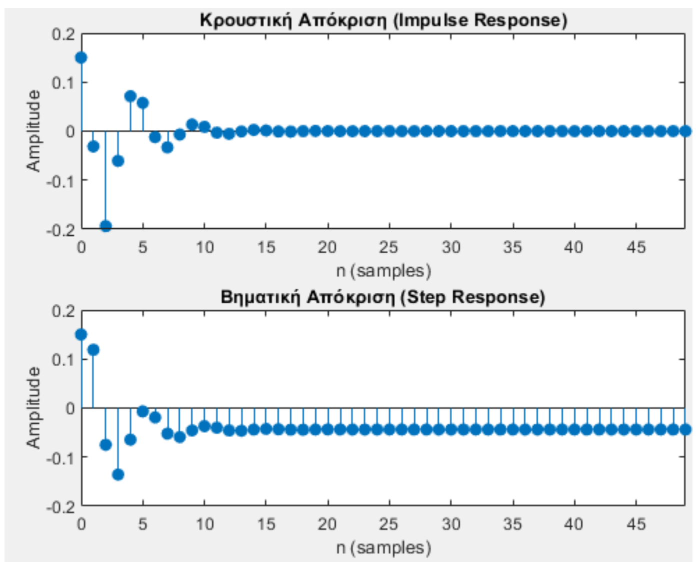
I also found the coefficient vectors a and b using zp2tf.



2.1.2 Responses

I plotted the magnitude response, phase response, impulse response, and step response diagrams.



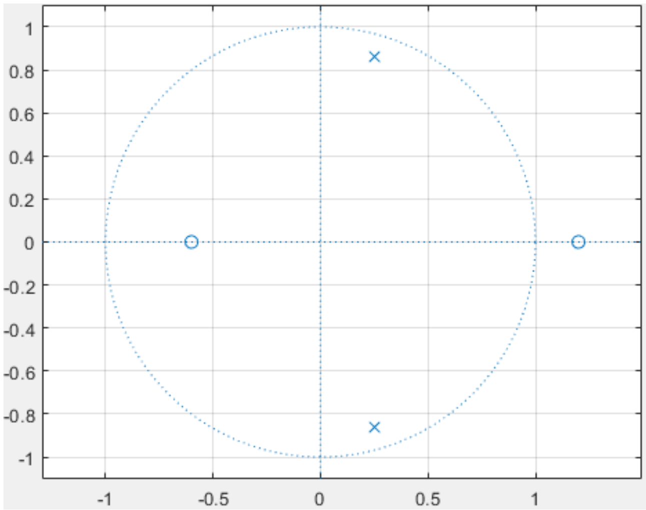


Magnitude and Phase Response

From the diagram, we can see that we have a band-pass filter. This filter is not linear.

2.1.3 Pole Movement

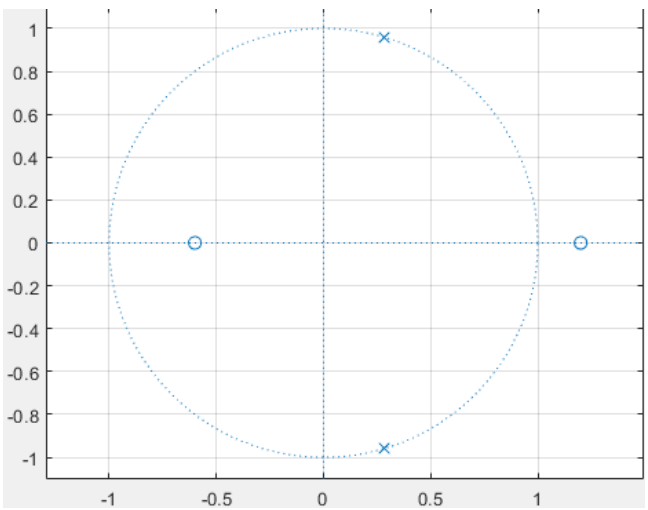
$0.252 \pm 0.864i$



We observe a normal filter whose poles have a distance from the center $|p| = 0.9$ and are located inside the unit circle. It is stable with a peak at 0.41.

Its response oscillates and settles.

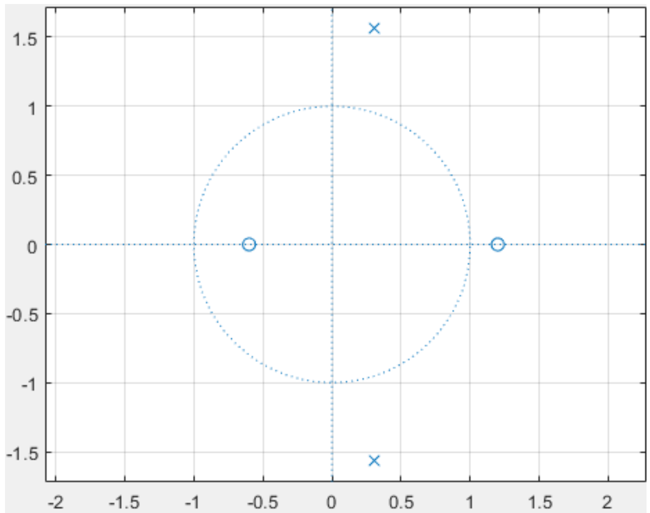
$0.28 \pm 0.96i$



We observe a marginal filter whose poles have a distance from the center $|p| = 1$ and are located exactly on the unit circle. It is marginally stable and "explodes" at 0.41.

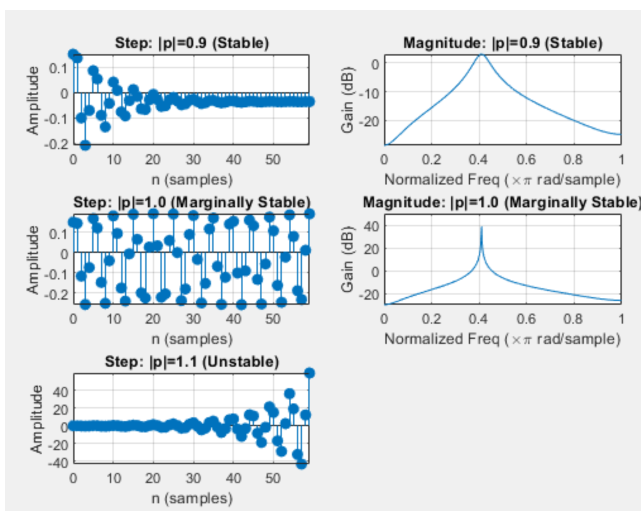
Its response oscillates but never settles.

$0.308 \pm 1.056i$



We observe an unstable filter since the poles are outside the unit circle ($|p| > 1$).

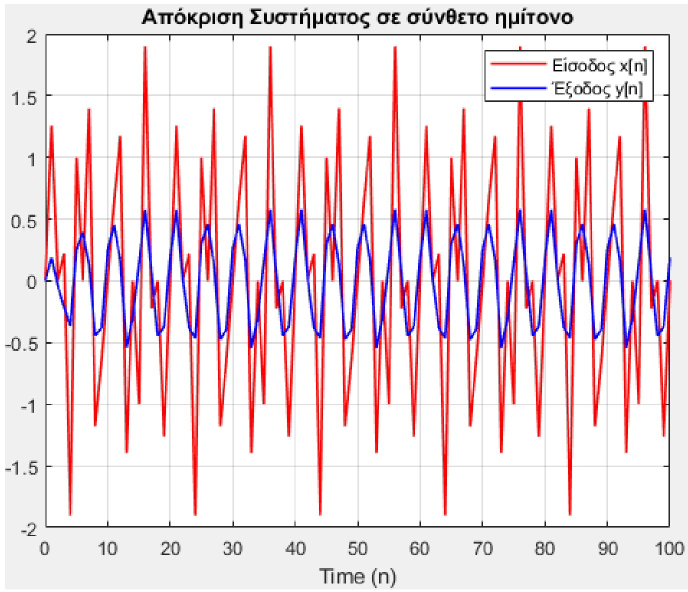
Its response diverges.



2.1.4 Signal Excitation

Using the filter, plot, and gen-sig commands, I used the signal $x[n] = \sin(0.4\pi n) + \sin(0.9\pi n)$ as the system input.

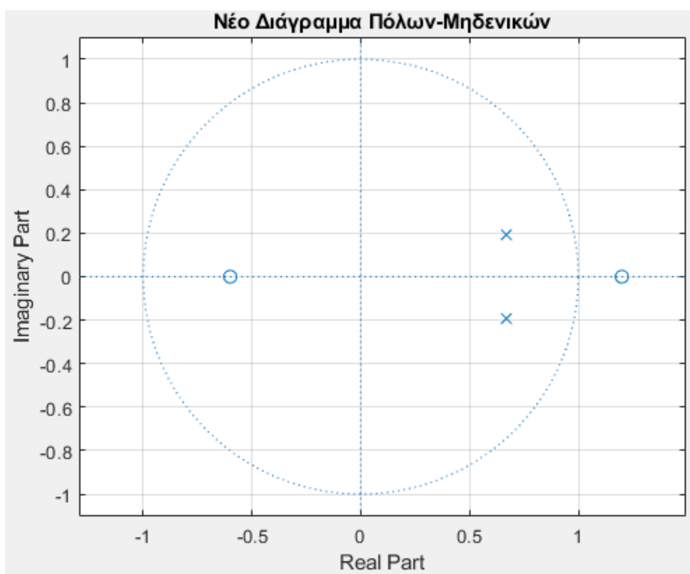
I observed how the filter smoothed the input signal while also significantly reducing its amplitude.

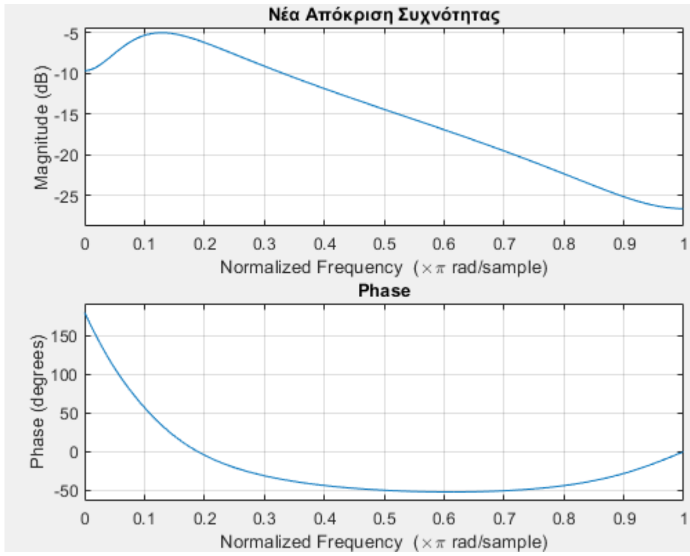


2.1.5 Filter Design with Different Poles

As before, I designed the same filter by changing the poles to positions $0.672 \pm 0.196i$.

In this example, the filter's passband moved from 0.41 to 0.1, much lower.

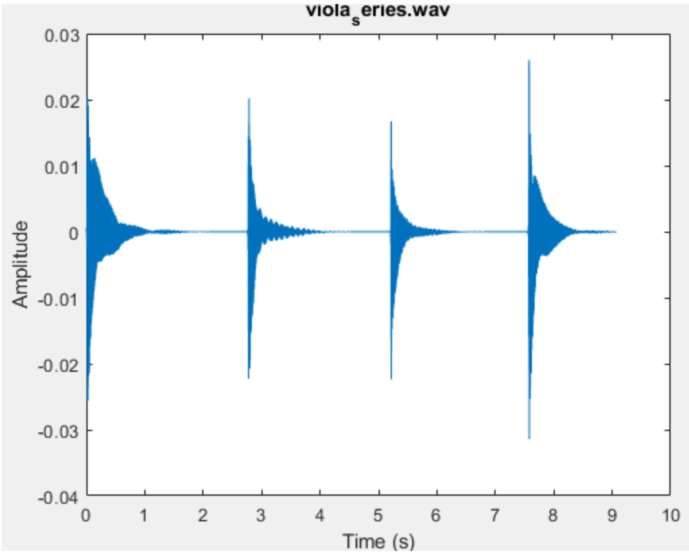


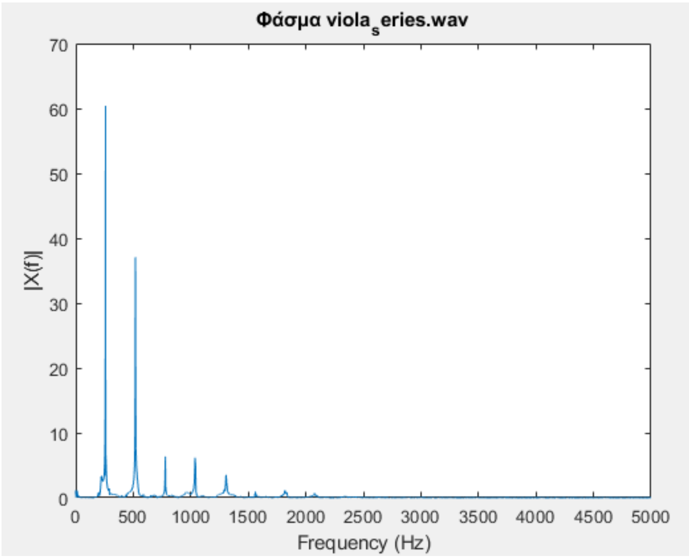

Νέα Απόκριση Συχνότητας
Magnitude (dB)
-5
-10
-15
-20
-25
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)
Phase
Phase (degrees)
150
100
50
0
-50
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)

2.2 Filter Application to Music Signals

At this point, I loaded the file "viola_series.wav" and plotted it graphically. I then observed in its spectrum that there were clear peaks approximately every 260Hz.

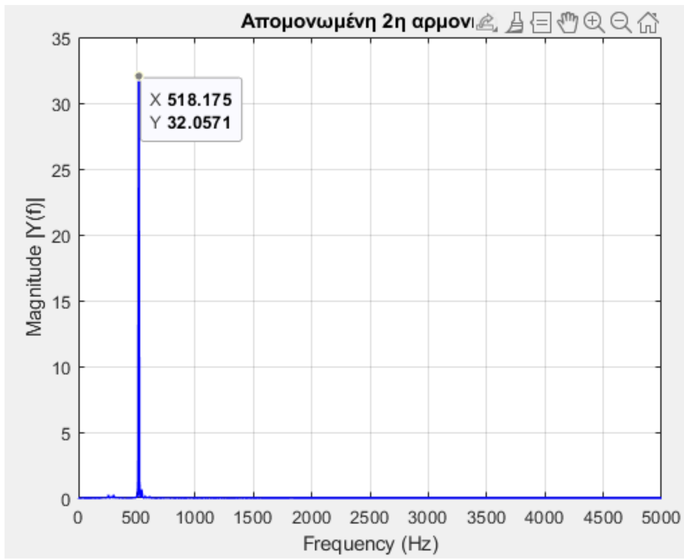
I also note that the peaks correspond to integer multiples of the fundamental frequency, but their magnitude decreases as frequency increases.

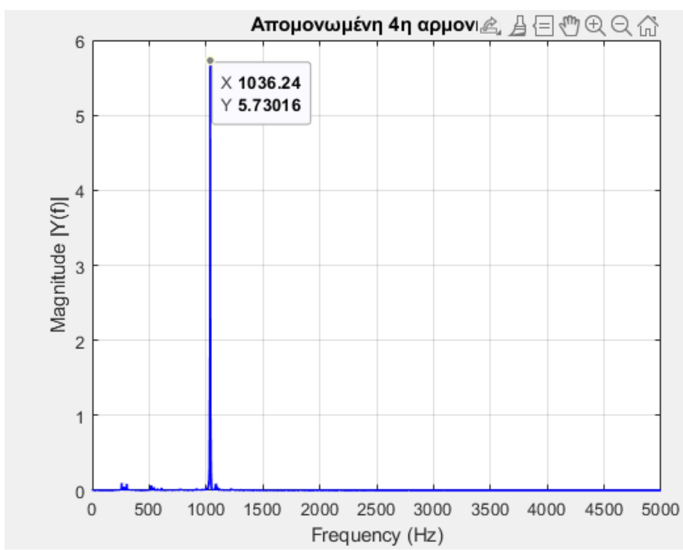




Using a filter with poles positioned to isolate the 2nd harmonic and with a distance from the center of 0.99 (very precise filter), we can isolate the harmonic by reducing it from 519Hz to 518Hz.

Similar results for the 4th harmonic.

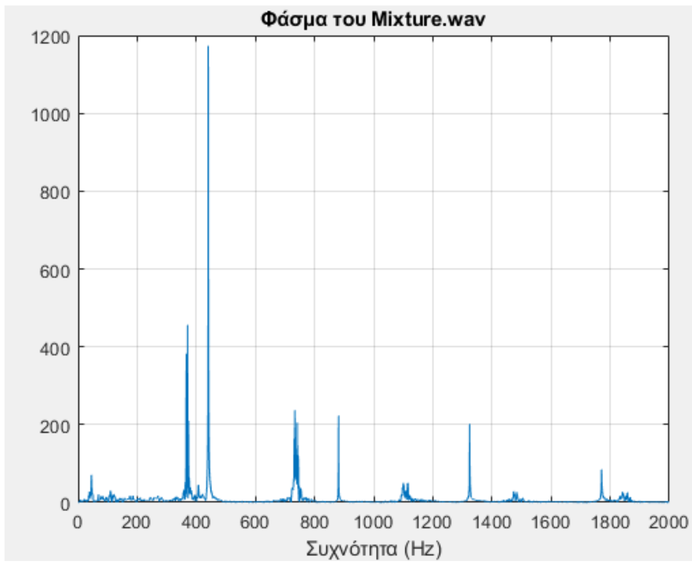




3 Musical Note Separation

3.1 Double Peaks

After loading and listening to the signal, it sounds like a combination of 2 instruments. After viewing its spectrum, we observe double peaks. Peaks at harmonics of 370Hz and 440Hz. Similarly to before, as frequency increases, the signal magnitude decreases.

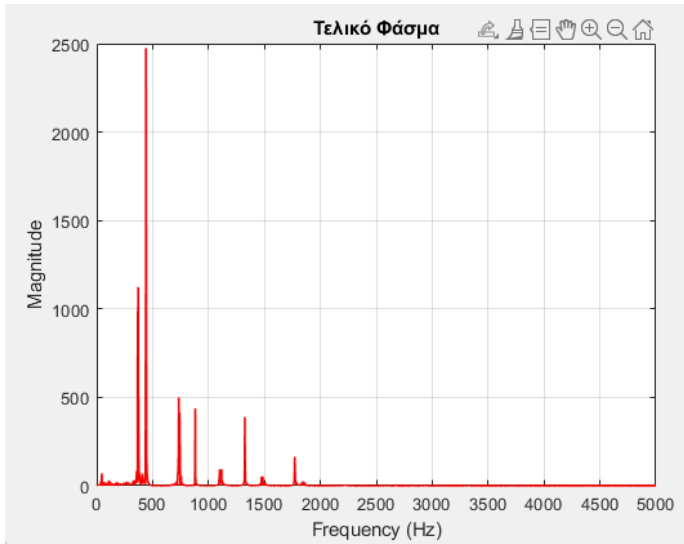


We have peaks approximately at the expected frequencies, which are: 370Hz, 740Hz, 1110Hz, 1480Hz and 440Hz, 880Hz, 1320Hz, 1760Hz.

3.2 Combination

Using high-precision filters, after combining the isolated frequencies, the signal was no longer 2 instruments but something that resembled both, although it was longer in duration.

Comparing with the original signals, we see that in the final combined signal we have the sound of the piano and the warmth of the brass instrument. However, the final sound is not as clear as the sound of the individual instruments.


Τελικό Φάσμα
Magnitude
Frequency (Hz)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

3.3 Combination 2

Combining the notes from the 2 instruments in "mixture2.wav", I observe that in the spectrum we don't have "pairs" of peaks but single peaks. We also observe that some, although at higher frequencies, are larger. Finally, the sound is more pleasant and smooth. All of this is due to the fact that one fundamental frequency is a harmonic of the other, and thus they add in the spectrum and in reality make a difference of one octave on the staff. They are essentially the same note.

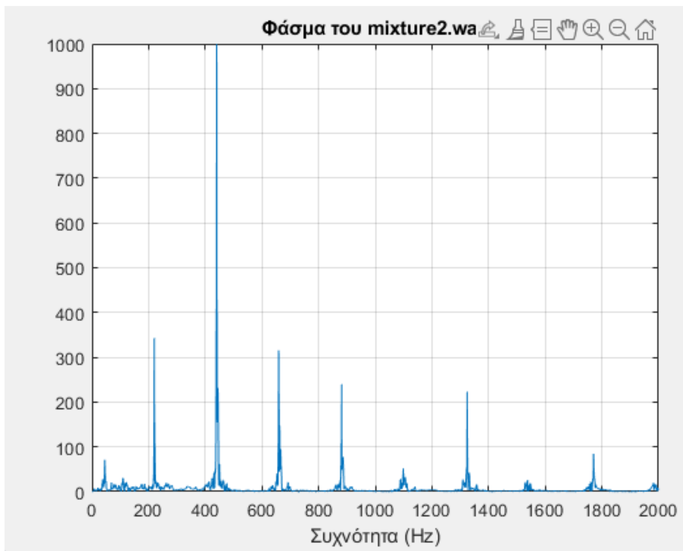


Figure 1: Before filtering

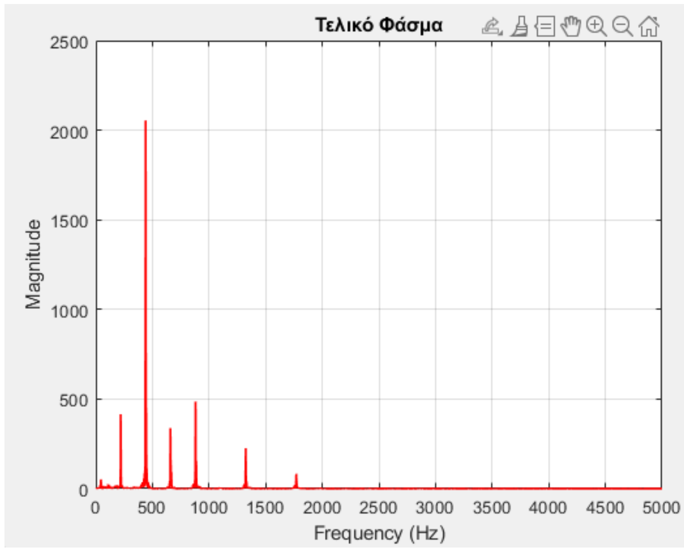


Figure 2: After filtering