

Εισαγωγή στην Ψηφιακή Επεξεργασία Σημάτων με
Python και Εφαρμογές σε Ακουστικά Σήματα

1η Εργαστηριακή Άσκηση

Ον/μα Συνεργατών:

Δημήτριος Βασιλειάδης ΑΜ: 03122111

Γρηγόριος Σταματόπουλος ΑΜ: 03122039

Περιεχόμενα:	1
1. Μέρος 1ο:	2
● 1.1	
● 1.2	
● 1.3	
● 1.4	
● 1.5	
● 1.6	
● 1.7	
2. Μέρος 2ο:	8
● 2.1	
● 2.2	
● 2.3	
● 2.4 Εφαρμογή σε σήμα μουσικής:	13
i. 2.1	
ii. 2.2	
iii. 2.3	
3. Μέρος 3ο:	16
● 3.1	
● 3.2	
● 3.3	
● 3.4	
● 3.5	
● 3.6	
● 3.7	
● 3.8	
● 3.9	
● 3.10	

Μέρος 1ο -Σύστημα Εντοπισμού Τηλεφωνικών Τόνων (Telephone Touch – Tones)

Εισαγωγή

Η εργασία αυτή ασχολείται με το σύστημα εντοπισμού τηλεφωνικών τόνων, το οποίο βασίζεται στον Διακριτό Μετασχηματισμό Fourier (DFT). Στόχος είναι η ανάλυση και υλοποίηση της διαδικασίας αναγνώρισης των τηλεφωνικών τόνων μέσω υπολογιστικών εργαλείων, καθώς και η ανάπτυξη αλγορίθμων για την ανίχνευση των αντίστοιχων πλήκτρων.

1.1 Δημιουργία των 10 τηλεφωνικών τόνων

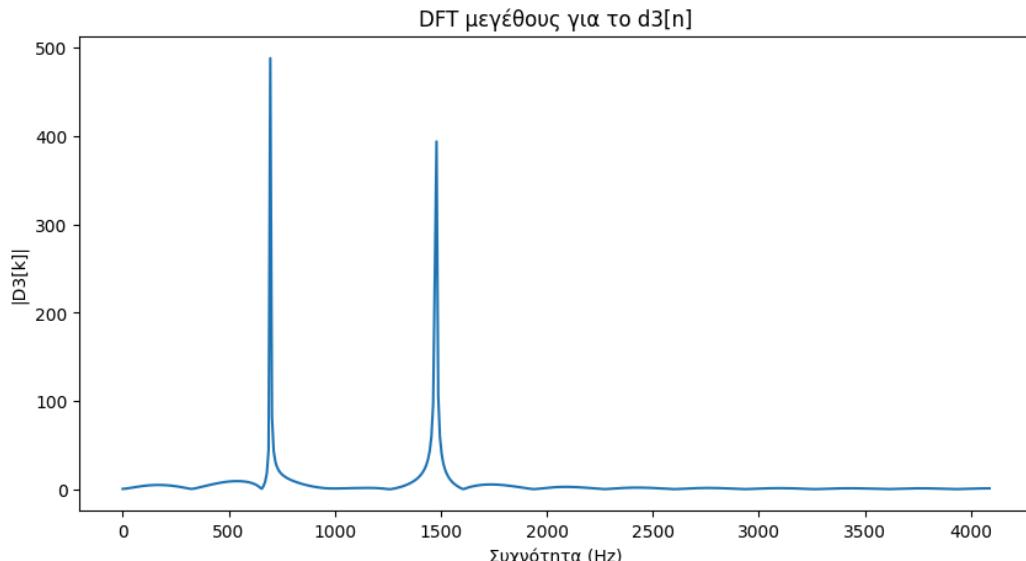
Για τη δημιουργία των τηλεφωνικών τόνων, χρησιμοποιούμε τις συχνότητες που παρέχονται στον Πίνακα 1 της εκφώνησης. Κάθε τόνος είναι το άθροισμα δύο ημιτονικών κυμάτων με διαφορετικές συχνότητες και έχει μήκος 1000 δειγμάτων.

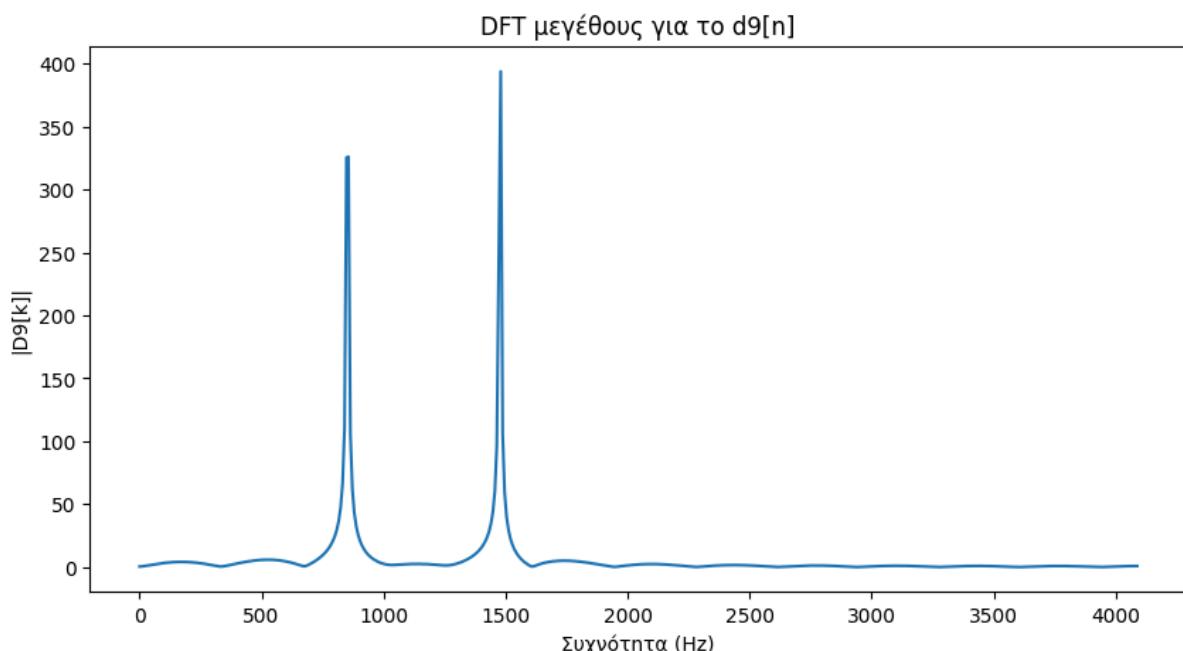
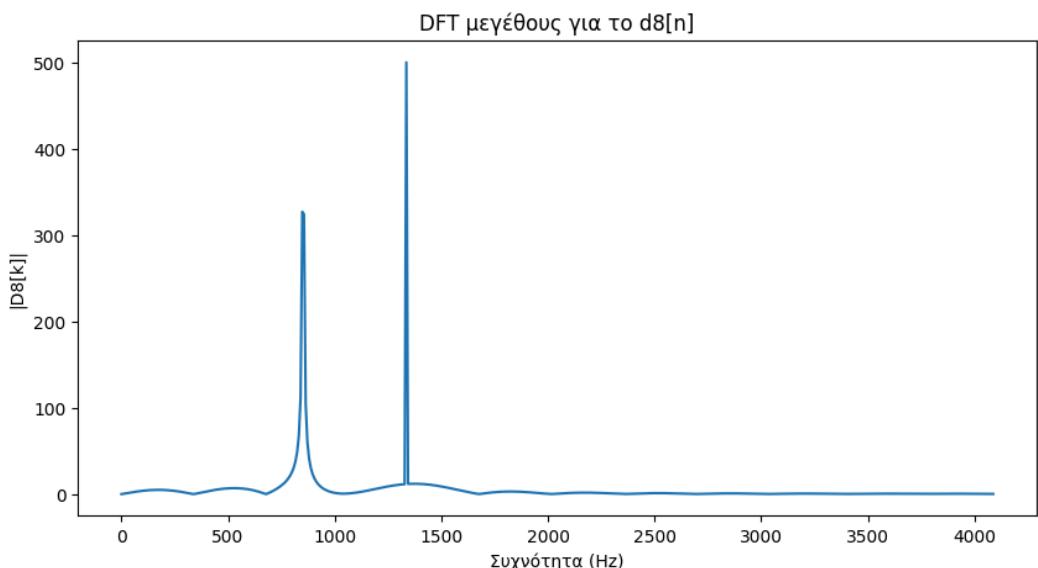
Ο κώδικας δημιουργεί τα ημιτονικά κύματα χρησιμοποιώντας τη συνάρτηση `sin()` της NumPy και τα αποθηκεύει σε μεταβλητές. Επιπλέον, χρησιμοποιείται η `Audio()` από το πακέτο `IPython.display` για την αναπαραγωγή των σημάτων.

1.2 Υπολογισμός του DFT για τόνους d3[n], d8[n] και d9[n]

Για τον υπολογισμό του DFT, εφαρμόζεται η `fft()` της NumPy με **N=1024** σημεία. Στη συνέχεια, υπολογίζουμε το μέτρο των μετασχηματισμένων σημάτων και δημιουργούνται οι αντίστοιχες γραφικές παραστάσεις μέσω της `matplotlib.pyplot`.

Παρατηρείται ότι κάθε γράφημα παρουσιάζει δύο κυρίαρχες κορυφές, οι οποίες αντιστοιχούν στις συχνότητες των τόνων που δημιουργήθηκαν, έτσι προσδιορίζουμε ότι κάθε τόνος προκύπτει από άθροισμα δύο ημιτόνων, ενα υψηλής και ένα χαμηλής συχνότητας.





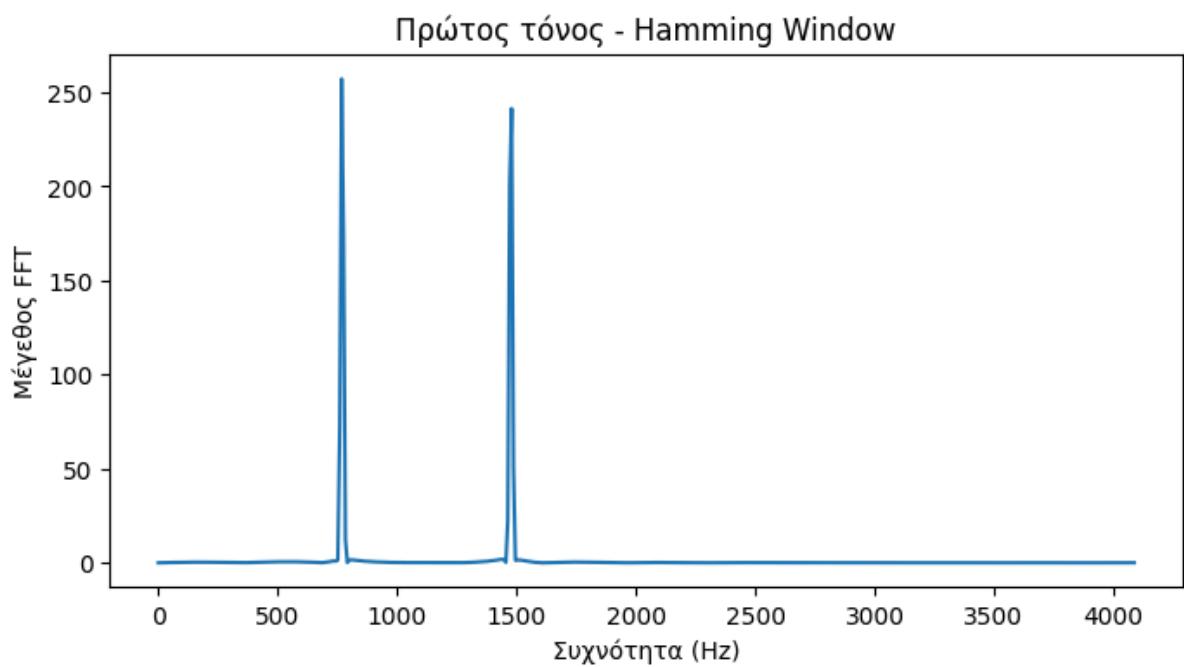
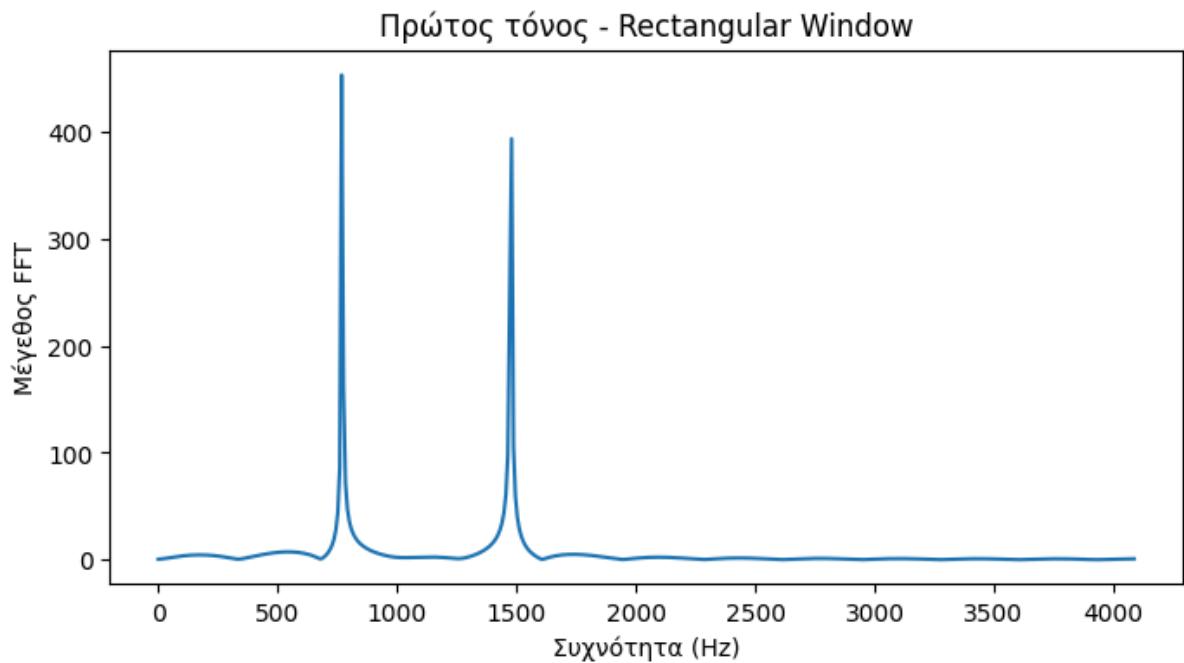
1.3 Δημιουργία αρχείου "tone_sequence.wav"

Το σήμα που δημιουργείται αντιστοιχεί στο άθροισμα των αριθμών μητρώου των μελών της ομάδας(03122111+03122039), με κάθε ψηφίο να αναπαρίσταται από τον αντίστοιχο τόνο. Τα ψηφία διαχωρίζονται μεταξύ τους με 100 μηδενικά δείγματα. Δημιουργήσαμε την ακολουθία ήχου “tone_sequence.wav”, ενώσαμε τα σήματα που αντιστοιχούν στους 2 αριθμούς μητρώου σε ένα μονοδιάστατο array με τη χρήση της np.concatenate και ακούσαμε το σήμα. Για την αποθήκευση του ηχητικού αρχείου χρησιμοποιήσαμε την συνάρτηση write() από τη βιβλιοθήκη soundfile.

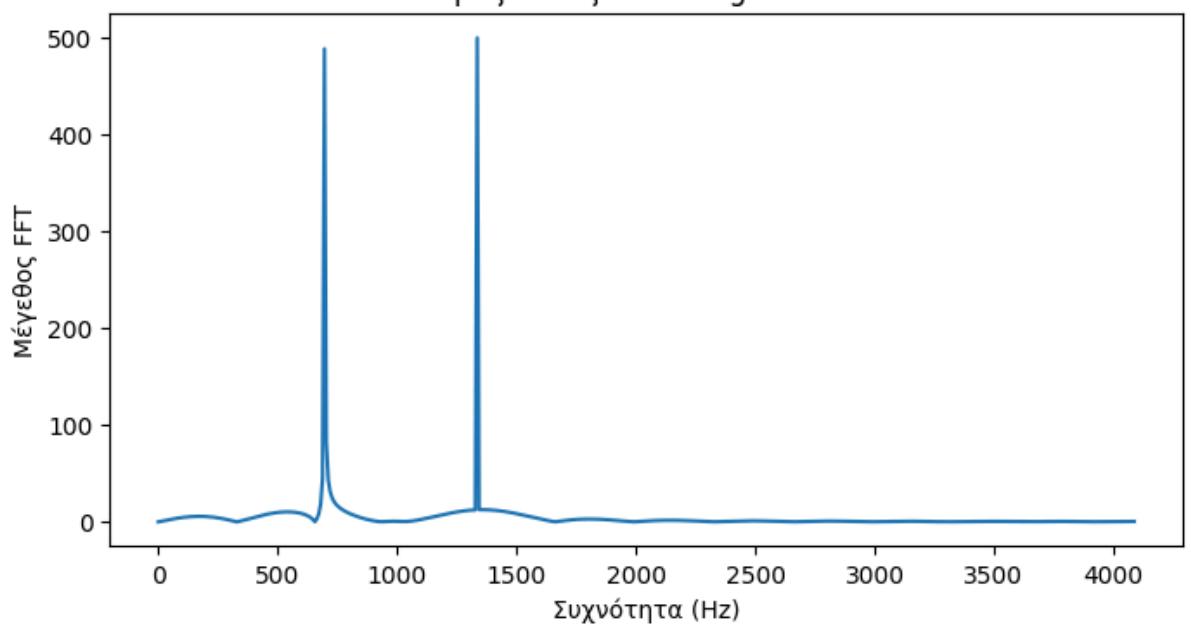
1.4 Υπολογισμός του DFT με παραθύρωση

Εφαρμόζουμε χρονικά παράθυρα στις σειρές δεδομένων και υπολογίζουμε τον γρήγορο μετασχηματισμό Fourier για $N=1024$ σημεία:

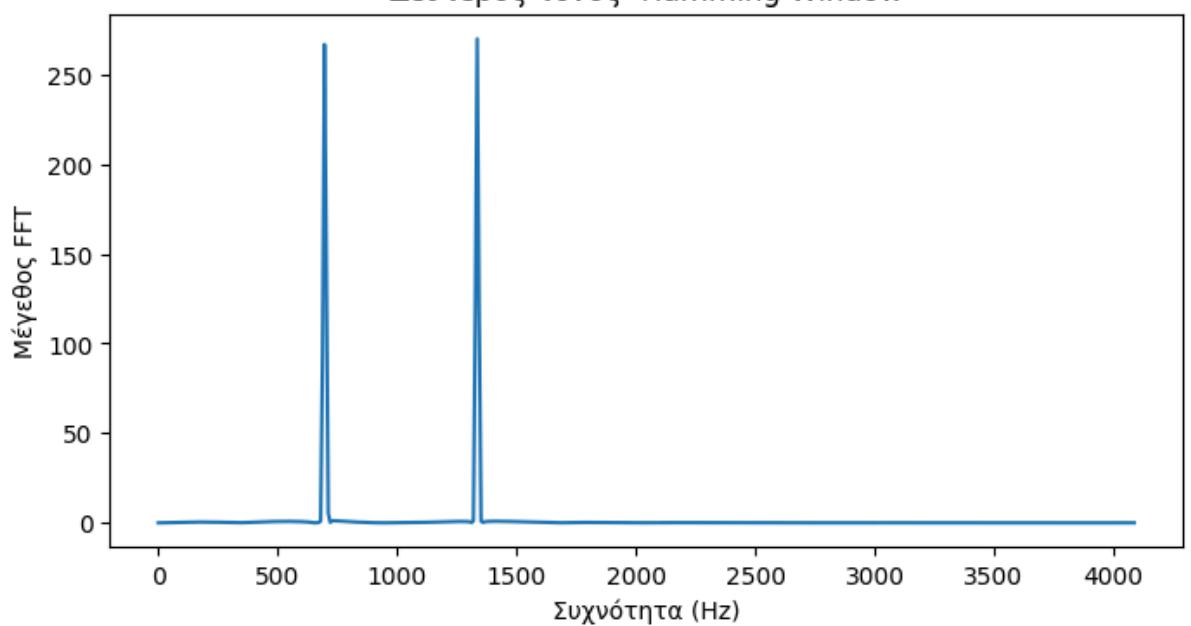
- Τετραγωνικό παράθυρο: Όπου τα δεδομένα κόβονται απότομα στο όριο του παραθύρου(αναδεικνύει τα side-lobes).
- Παράθυρο Hamming: Το οποίο εξομαλύνει τις άκρες του παραθύρου για καλύτερη φασματική ανάλυση.

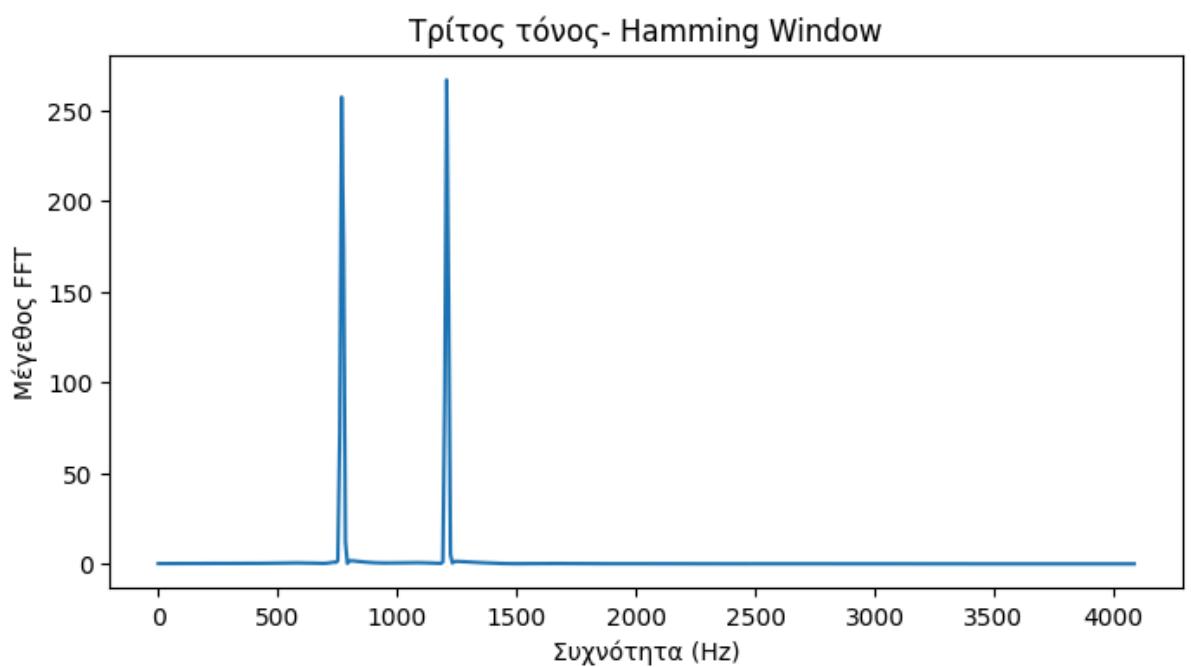
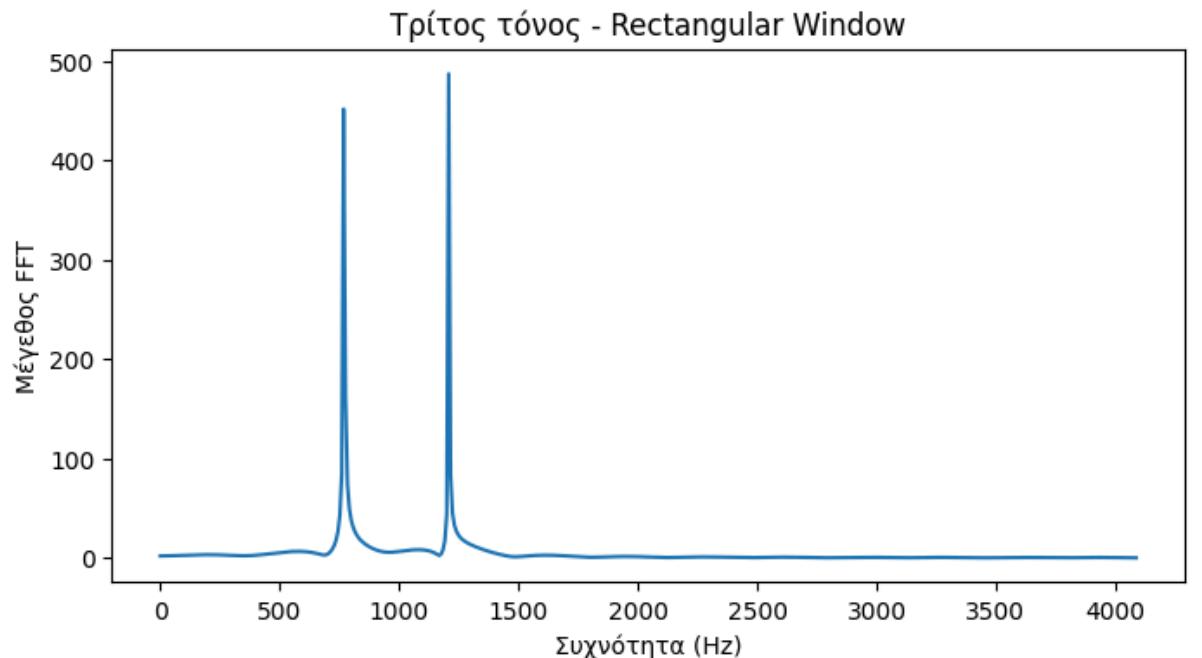


Δεύτερος τόνος - Rectangular Window



Δεύτερος τόνος- Hamming Window





Στον κώδικας χρησιμοποιούμε τις συναρτήσεις `window()` και `fft()` της NumPy για την εφαρμογή των παραθύρων και την υπολογιστική ανάλυση.

1.5 Εύρεση δεικτών k και αντιστοίχιση συχνοτήτων

Υπολογίζουμε η **λίστα** των δεικτών k που αντιστοιχούν στις κυρίαρχες συχνότητες του σήματος. Οι τιμές αυτές αντιστοιχίζονται με τις αρχικές συχνότητες του τηλεφωνικού πληκτρολογίου, επιτρέποντας τον εντοπισμό των πατημένων πλήκτρων. Παραθέτουμε τον πίνακα:

Υπολογισμός κοντινότερων δεικτών κ για τις δοθείσες διακριτές συχνότητες:

$\omega=0.5346$, $k\approx87$, $f_{\text{est}}=696.00 \text{ Hz}$

$\omega=0.5906$, $k\approx96$, $f_{\text{est}}=768.00 \text{ Hz}$

$\omega=0.6535$, $k\approx107$, $f_{\text{est}}=856.00 \text{ Hz}$

$\omega=0.7217$, $k\approx118$, $f_{\text{est}}=944.00 \text{ Hz}$

$\omega=0.9273$, $k\approx151$, $f_{\text{est}}=1208.00 \text{ Hz}$

$\omega=1.0247$, $k\approx167$, $f_{\text{est}}=1336.00 \text{ Hz}$

$\omega=1.1328$, $k\approx185$, $f_{\text{est}}=1480.00 \text{ Hz}$

1.6 Υλοποίηση της συνάρτησης ttdecode()

Η συνάρτηση ttdecode() δέχεται ως είσοδο το ηχητικό σήμα και επιστρέφει με τη σειρά που ακούγονται, τα ψηφία που περιέχει. Η διαδικασία περιλαμβάνει τα εξής βήματα.

- # Βήματα:
- # (1) Διαχωρίζουμε το σήμα σε τμήματα τόνων (μήκους 1000) αγνοώντας τα μηδενικά
- # (2) Υπολογίζουμε FFT και την ενέργεια $|X[k]|^2$
- # (3) Βρίσκουμε τις δύο κορυφές στο φάσμα
- # (4) Χαρτογραφούμε τις δύο κορυφές στις row/col συχνότητες
- # (5) Επιστρέφουμε το αντίστοιχο ψηφίο

Για τα πιο σύνθετα σήματα, κατασκευάσαμε ακόμη μια συνάρτηση ttdecode_variable_gap() η οποία δεν λαμβάνει ως δεδομένο το **gap length**, αλλά δέχεται αντιθέτως ένα **threshold** ενέργειας το οποίο θέτουμε εμείς κατάλληλα και δρα ανάλογα για να βρει το αρχικό σήμα. Η συνάρτηση ελέγχεται με το σήμα που δημιουργήσαμε στο **Ερώτημα 1.3**, επιβεβαιώνοντας ότι η αποκωδικοποίηση λειτουργεί σωστά. (**6244150**)

1.7 Ανάλυση των σημάτων **easySig**, **mediumSig** και **hardSig**

Τα σήματα φορτώνονται από τα αρχεία `.npy` με τη χρήση της `npy.load()`. Στη συνέχεια, αναλύονται μέσω της ttdecode(), η οποία αναγνωρίζει τα ψηφία που περιέχονται στο κάθε σήμα.

Η διαδικασία αποδεικνύει την ικανότητα του συστήματός μας να αποκωδικοποιεί τηλεφωνικούς τόνους ακόμη και σε διαφορετικές συνθήκες πολυπλοκότητας θορύβου, μέσω του `ttdecode_variable_gap()` όπως μπορούμε να ακούσουμε στο `hardSig.npy`.

Τα αποτελέσματά μας είναι :

easy_sig -> 20136515, medium_sig -> 4748011820, hard_sig -> 0360937304

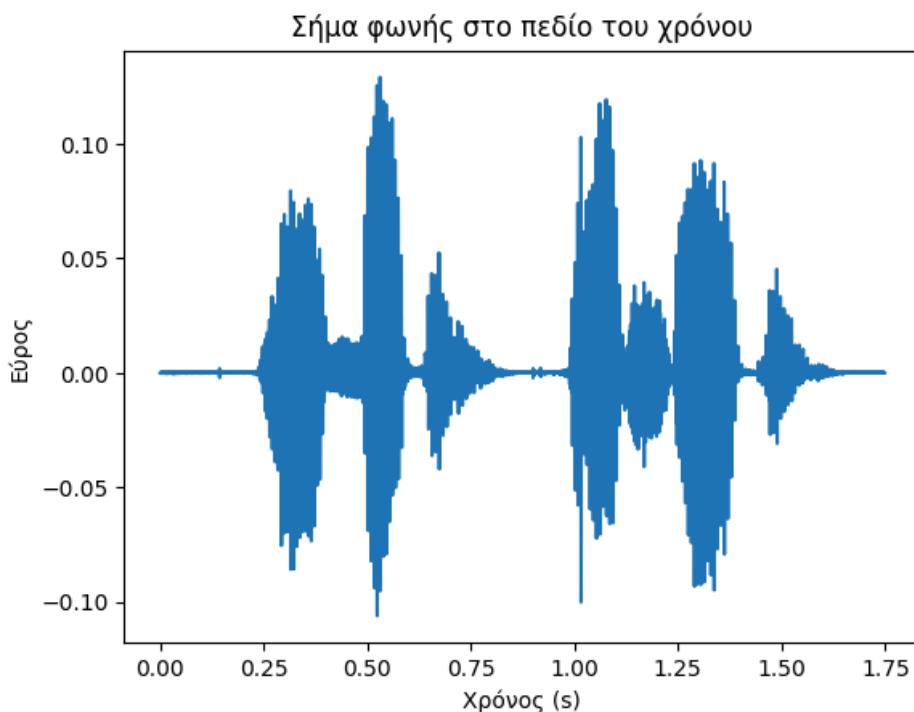
Μέρος 2ο - Χαρακτηριστικά Βραχέος Χρόνου Σημάτων Φωνής και Μουσικής

2.1

Φορτώνουμε το αρχείο ήχου “`speech_utterance.wav`” και προβάλλουμε το σήμα στο πεδίο του χρόνου:

Για να το διαβάσουμε χρησιμοποιήσαμε τη βιβλιοθήκη `soundfile` και με την εντολή `sf.read()` λαμβάνουμε το σήμα `speech` και τη συχνότητα δειγματοληψίας `sr_speech`.

Έπειτα πλοτάρουμε το δοσμένο σήμα στο πεδίο του χρόνου χρησιμοποιώντας άνω όριο το `np.arange()` του μήκους του σήματος διά τη συχνότητα δειγματοληψίας. Στο τέλος αναπαράγουμε το σήμα φωνής με την εντολή `display(Audio())`. Η απεικόνιση του σήματος στο πεδίο του χρόνου (διάρκειας 1,75) είναι:

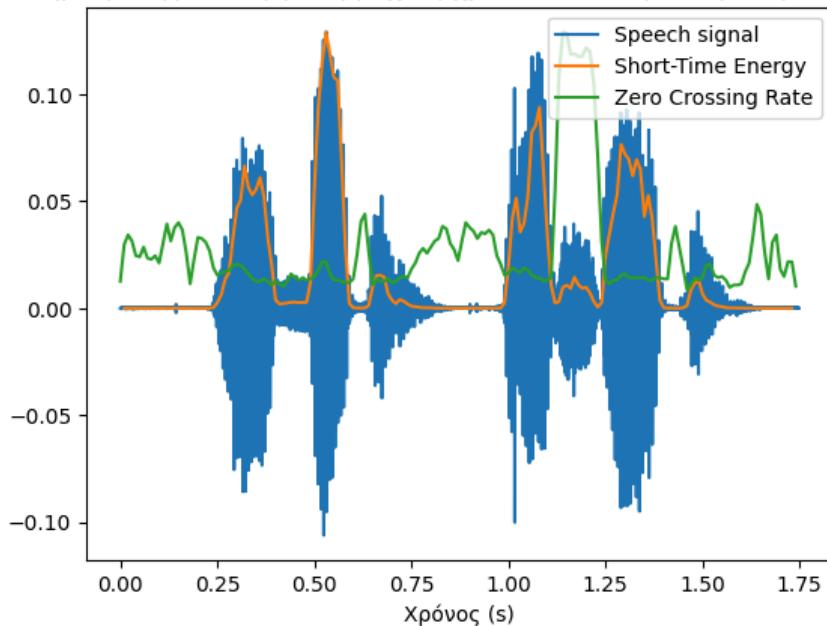


2.2

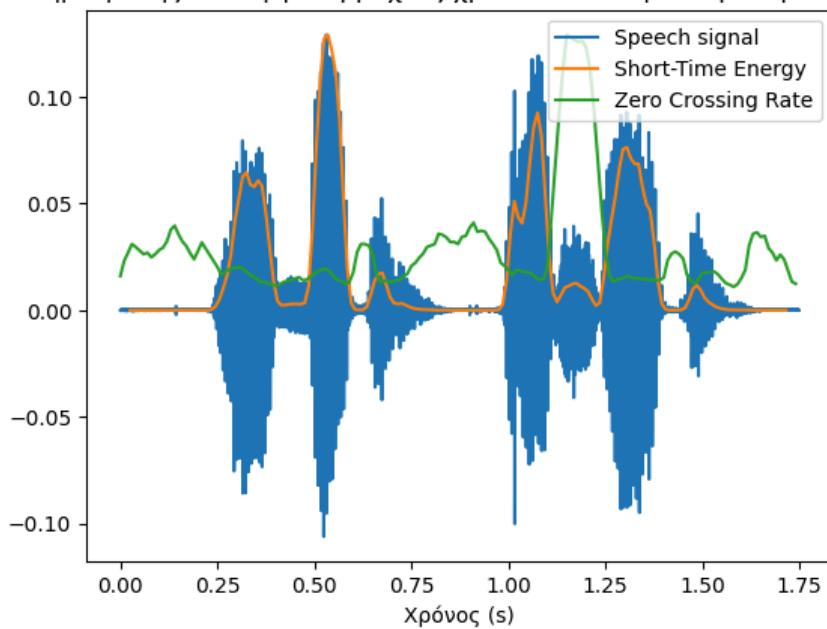
Υπολογίζουμε την ενέργεια βραχέος χρόνου και το ρυθμό εναλλαγής προσήμου του σήματος, βάσει των τύπων (1) και (2) και για τα οποία χρησιμοποιήσαμε **παράθυρα 20ms και 50ms**, ενώ αφήσαμε **hop duration στα 10ms**. Χρησιμοποιήσαμε παράθυρο Hamming για την παραγωγή και των δύο συναρτήσεων `ste()` και `zero_crossing_rate()`, τα οποία προκλήθηκαν με την χρήση της συνάρτησης `np.convolve()` του σήματος φωνής με τα

αντίστοιχα τροποποιημένα παράθυρα για αντίστοιχη χρήση . Τέλος, αναπαριστούμε για κάθε μήκος παραθύρου διαγράμματα στα οποία συνυπάρχουν **σήμα φωνής**, **Ενέργεια Βραγχέος Χρόνου και Ρυθμός Εναλλαγής Προσήμου**. Τα διαγράμματα τα **κανονικοποιούμε** ως προς το σήμα φωνής, ώστε να είναι πιο εύκολη η σύγκριση τους. Τα διαγράμματα είναι τα εξής:

Σήμα φωνής + Ενέργεια βραχέος χρόνου + ZCR για παράθυρο 20ms



Σήμα φωνής + Ενέργεια βραχέος χρόνου + ZCR για παράθυρο 50ms



Σχετικά με τα παράθυρα, μεγαλώνοντας το μήκος του παραθύρου παρατηρούμε πιο λεία αναπαράσταση, δηλαδή με λιγότερο οξείες διακυμάνσεις. Με την πρακτική αυτή μειώνεται η **λεπτομέρεια** στα διαγράμματα μας, αλλά στην πραγματικότητα αποκόπτεται θόρυβος οπότε η απεικόνιση της ενέργειας και του ρυθμού εναλλαγής προσήμου είναι πιο εύχρηστη.

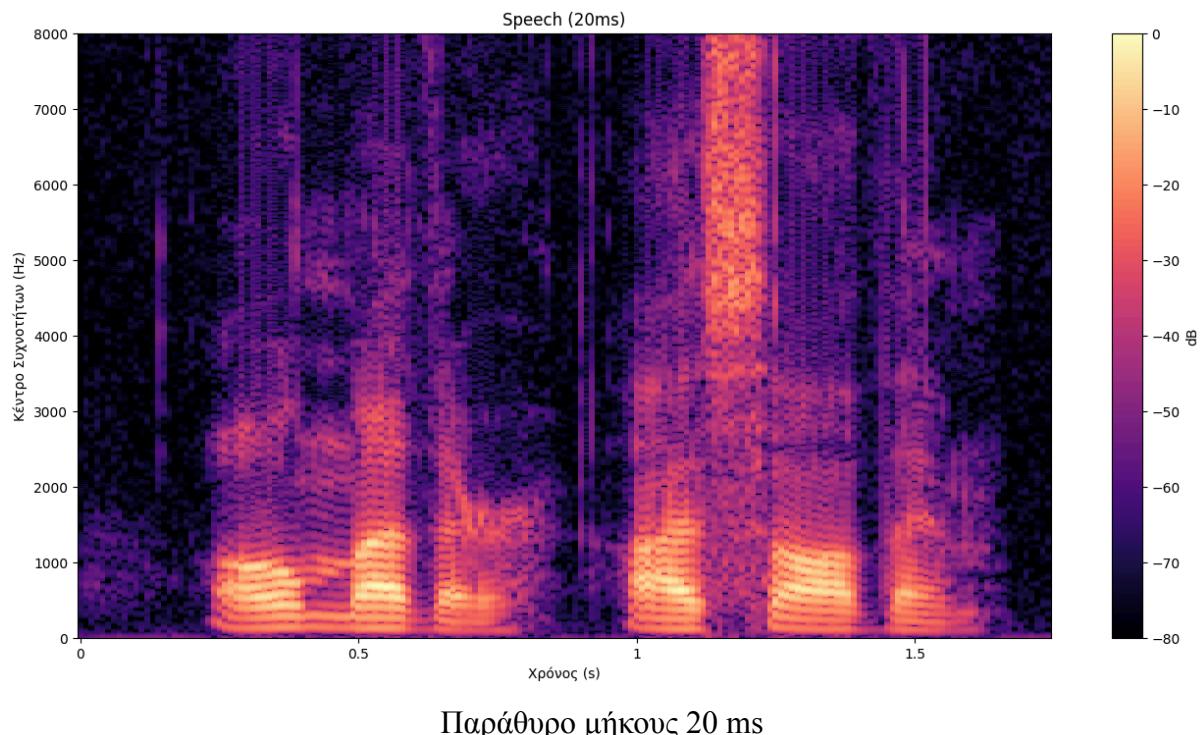
Σχετικά με τη φωνή και τη σιωπή, παρατηρούμε πως η ενέργεια βραχέος χρόνου του σήματος ομιλίας παρουσιάζει υψηλές τιμές στα τμήματα όπου η φωνή είναι δυνατότερη (π.χ. φωνήεντα ή τονισμένες συλλαβές) και χαμηλές τιμές κατά τις παύσεις ή τα άηχα σύμφωνα ενώ ο ρυθμός εναλλαγής προσήμου (ZCR) είναι μικρός σε ηχηρά /φωνηεντικά τμήματα και αυξάνεται σε άηχα τμήματα ή θορύβους (όπου το σήμα αλλάζει πιο συχνά πρόσημο).

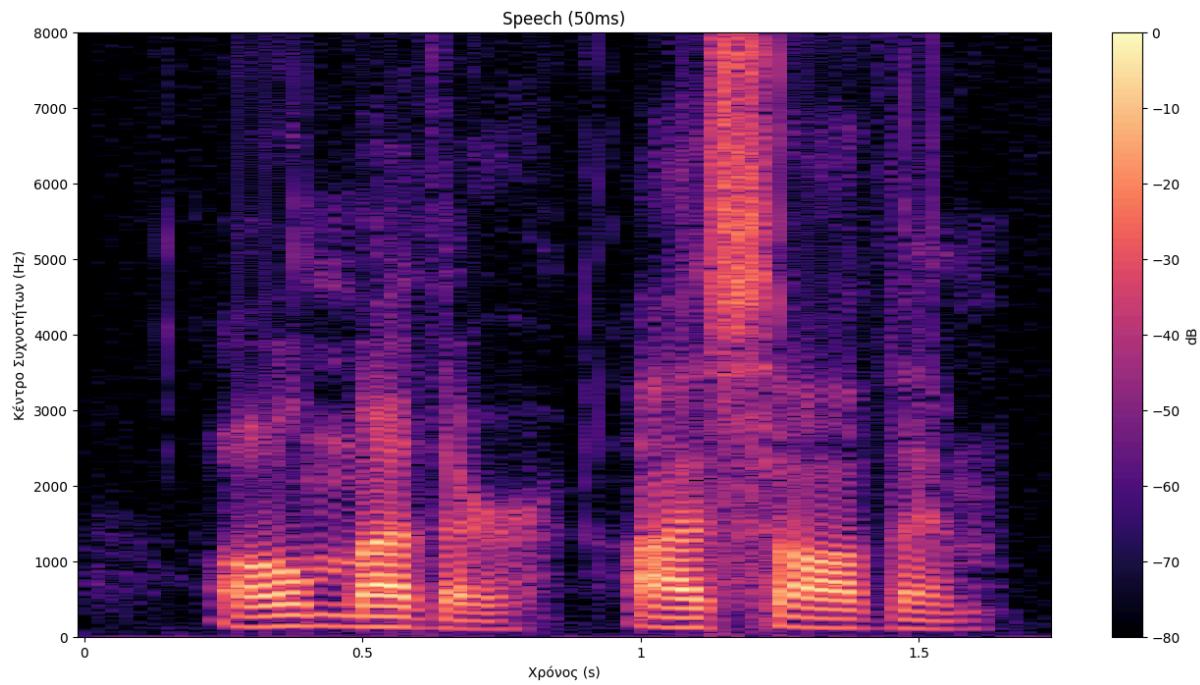
Επομένως, μπορούμε να συμπεράνουμε πως οι έμφωνοι ήχοι παρατηρούνται στις περιοχές όπου έχουμε ενέργεια σε peak ενώ ο ρυθμός εναλλαγής προσήμου είναι μηδενικός. Από την άλλη οι άφωνοι ήχοι παρατηρούνται όταν η ενέργεια είναι χαμηλή ενώ το ο ρυθμός εναλλαγής προσήμου σε peak.

2.3

Αρχικά υπολογίζουμε τον **STFT του σήματος** και τα μήκη των παραθύρων, μέσω της συνάρτησης που θεωρούμε εμείς ως `compute_spectrogram()`. Εκεί, **αρχικά** ανάλογα με το παράθυρο που δέχεται ως είσοδο και τη συχνότητα δειγματοληψίας βρίσκουμε το μήκος του παραθύρου και το είδος του. Στη συνέχεια, με τη βοήθεια της `συνάρτησης stft()` της `librosa` βρίσκουμε τον STFT και σχεδιάζουμε το **πλάτος** του (`abs()`) σε λογαριθμική **κλίμακα** (`log10()`) από τη `numpy`. Τέλος μετατρέπουμε το πλάτος σε dB μέσω της βιβλιοθήκης `amplitude_to_db()` της `librosa`.

Έτσι τώρα έχουμε **διαθέσιμα** και για το σήμα φωνής, αλλά και μουσικής όπως θα δούμε στη συνέχεια, τα χαρακτηριστικά που χρειάζεται να βάλουμε ως **είσοδο** στη συνάρτηση `specshow()` της `librosa display` για να παρουσιάσουμε το **φασματογράφημά** μας.





Παράθυρο μήκους 50ms

Πειραματιζόμενοι με το εύρος του παραθύρου για τιμές **20-50 ms** και **N=2048 σημεία** παρατηρούμε ότι όσο **μεγαλύτερο το εύρος του παραθύρου** τόσο πέφτει η **χρονική ανάλυση**, δηλαδή το φασματογράφημα γίνεται πιο **ομαλό** και **χάνεται η ακρίβεια** στην καταγραφή κάποιων γρήγορων αλλαγών.

Στα παραπάνω φασματογραφήματα, παρατηρούμε τη **φασματική δομή του φωνητικού σήματος**. Οι **σκοτεινότερες** περιοχές αντιστοιχούν σε **υψηλότερη ένταση** σε συγκεκριμένες συχνότητες. Διακρίνονται **οριζόντιες ζώνες** έντονης ενέργειας στις **χαμηλότερες** συχνότητες, που αντιστοιχούν στις φορμάντες των φωνηέντων. Αντίθετα, στα σημεία με σύμφωνα όπως το /sh/, η ενέργεια **μετατοπίζεται** προς **υψηλότερες συχνότητες** (πιο φωτεινές περιοχές στα **υψηλά Hz**).

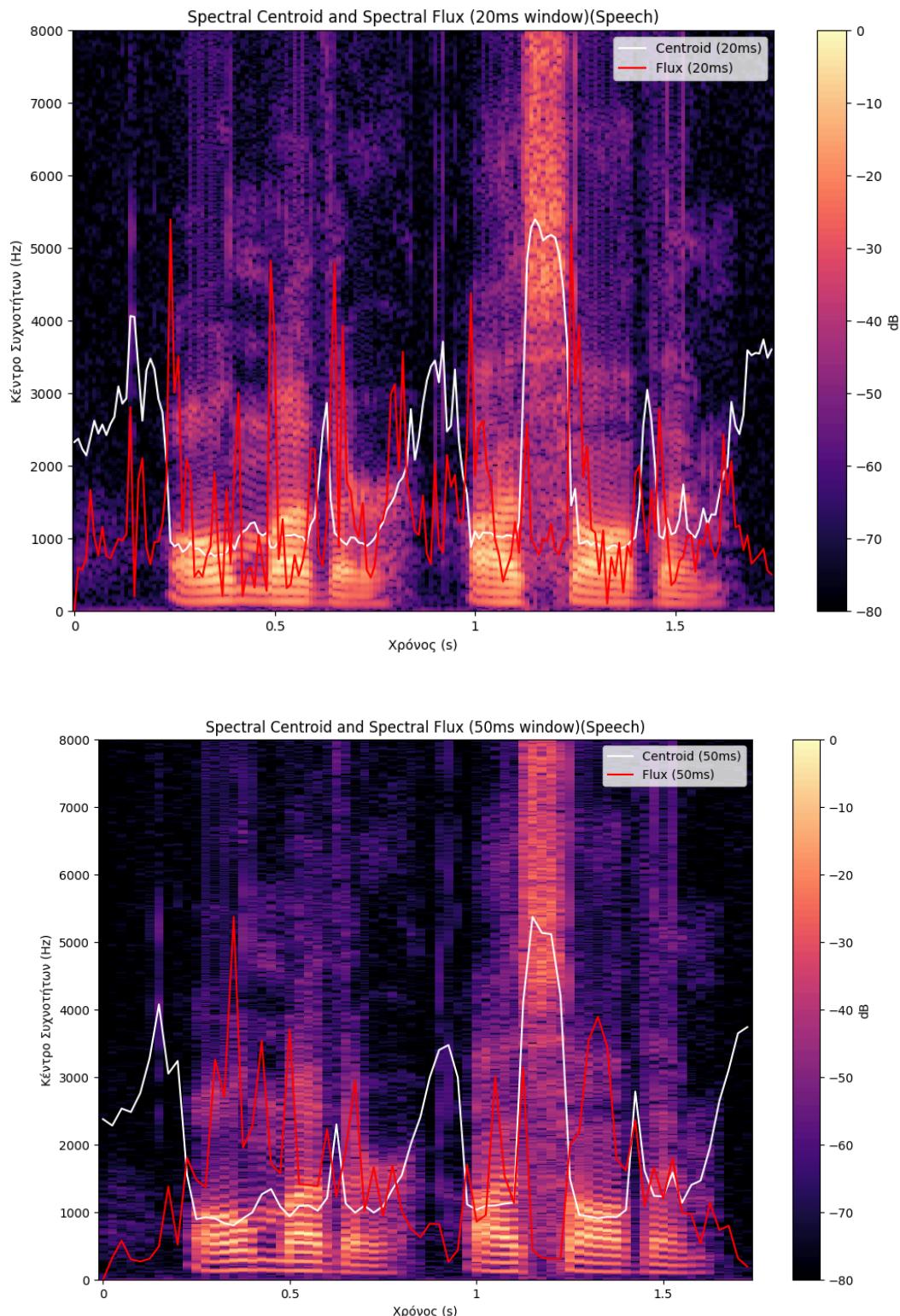
Στη συνέχεια χρησιμοποιούμε το STFT για να πάρουμε τις τιμές του **φασματικού κέντρου(Spectral Centroid)** και της **φασματικής ροής**, σύμφωνα με τους δοσμένους τύπους, τα οποία φαίνονται παρακάτω, για 20ms και 50ms παράθυρο. Προσπαθήσαμε να τα **κανονικοποιήσουμε** στο μέγεθος του φασματικού κέντρου.

Συμπεράσματα - Παρατηρήσεις:

Για το **φασματικό κέντρο** του σήματος ομιλίας: Παρατηρούμε ότι στις περιοχές όπου κυριαρχούν **χαμηλές συχνότητες** (π.χ. φωνήντα με χαμηλό τόνο), το **κέντρο βάρους βρίσκεται χαμηλότερα** (π.χ. κάτω από **1000 Hz**). Αντίθετα, όταν εμφανίζονται ήχοι με περισσότερη **υψηλή συχνότητα** (όπως τα σφυρίγματα /sh/), το φασματικό κέντρο **ανεβαίνει σημαντικά**.

Για τη **φασματική ροή**, δηλαδή ο ρυθμός μεταβολής του φάσματος: Παρατηρούμε **ακίδες** στη φασματική ροή αντιστοιχούν σε **απότομες αλλαγές στο φάσμα**. Για παράδειγμα, στα **όρια μεταξύ φθόγγων** (όπου αλλάζει από φωνήν σε σύμφωνο ή σιωπή) εμφανίζονται

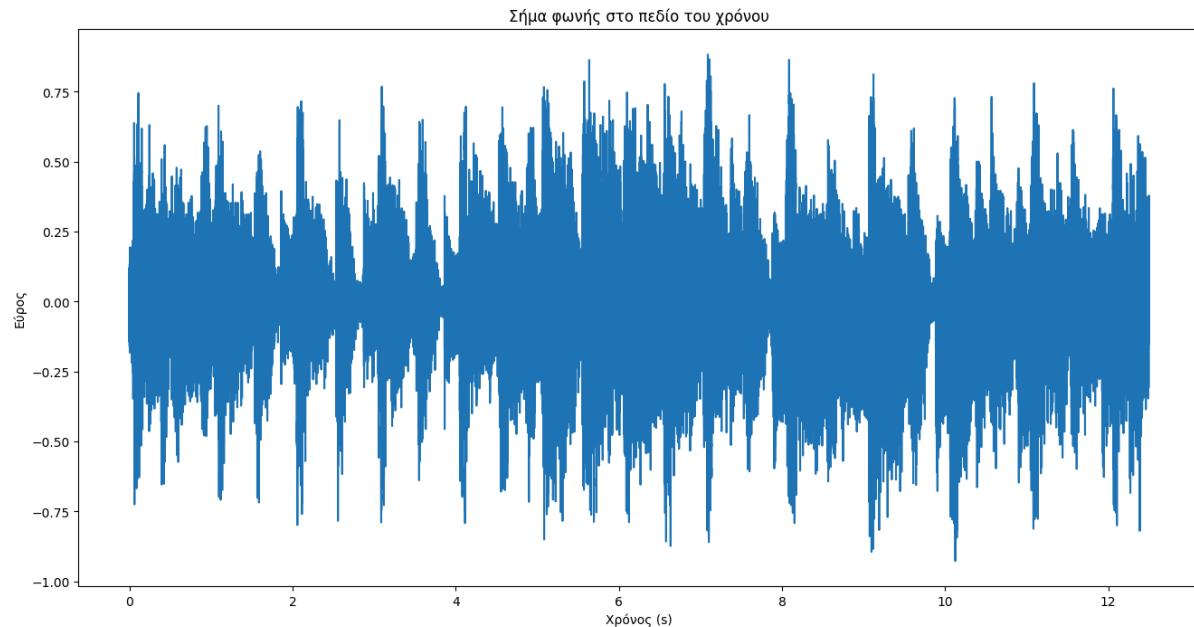
υψηλές τιμές φασματικής ροής, υποδεικνύοντας μια ηχητική μετάβαση ή **έναρξη νέου ήχου** Τέλος, τα γραφήματα του παραθύρου 20ms παραμένουν πιο "αιχμηρά" και αναλυτικά.



Μέρος 2ο - Ηχητικό Σήμα Μουσικής

2.4

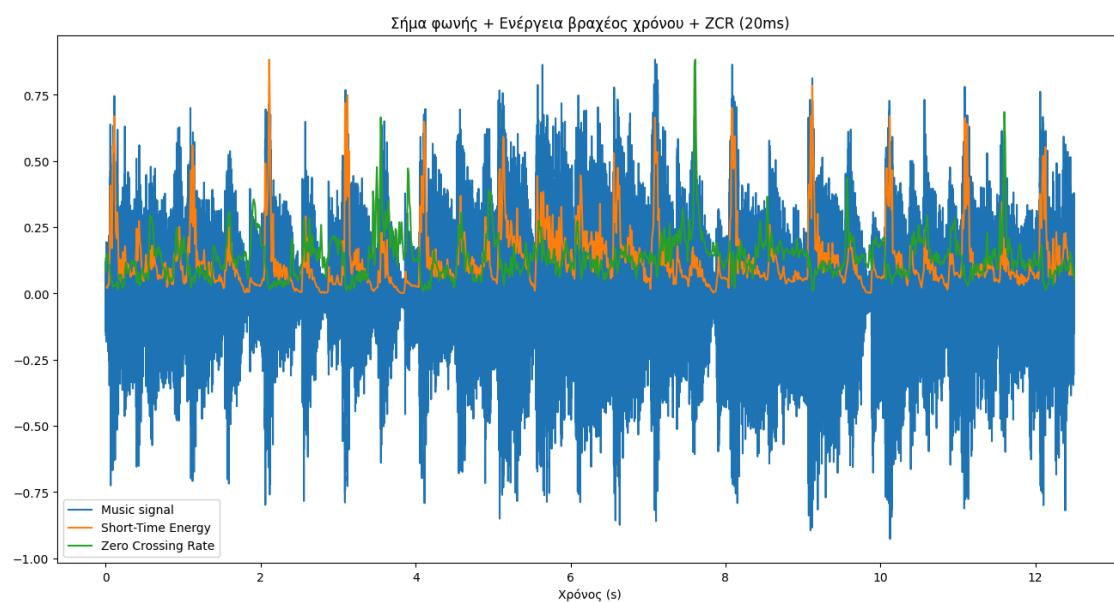
Επαναλαμβάνουμε τη διαδικασία για το δοσμένο σήμα μουσικής “music.wav”, το οποίο αρχικά διαβάζουμε και εξάγουμε συχνότητα δειγματοληψίας $sr_music = 48\text{kHz}$, το πλοτάρουμε (και το αναπαράγουμε). Η **απεικόνιση του σήματος μουσικής στο πεδίο του χρόνου** είναι:



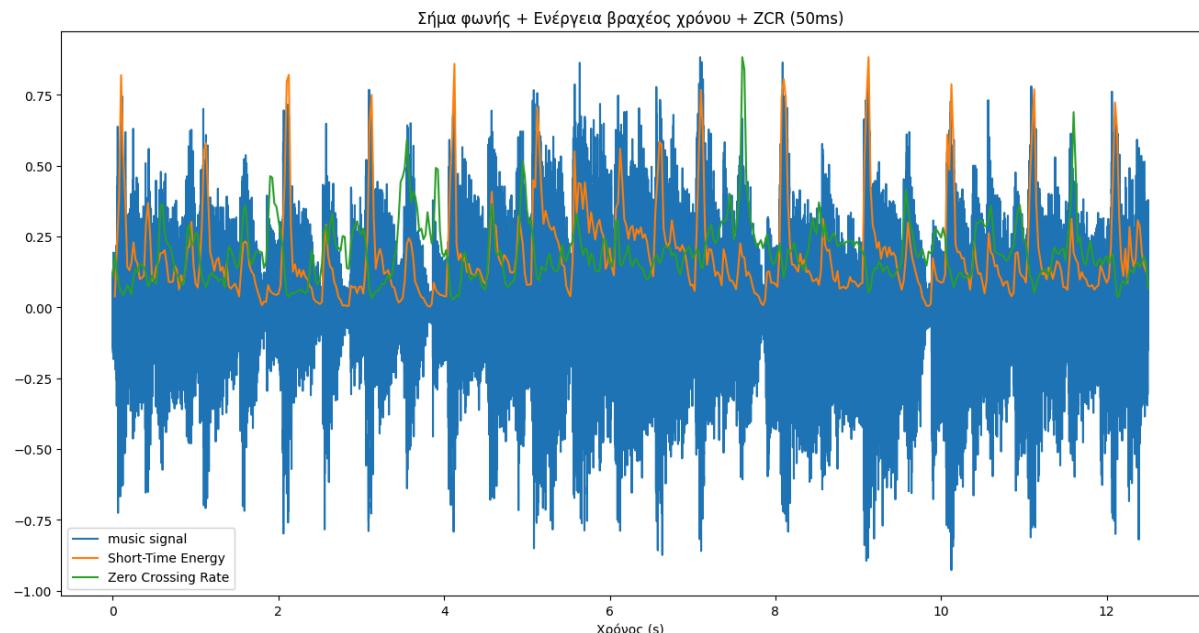
Παρατηρούμε το διάγραμμα του σήματος στο πεδίο του χρόνου. Παρουσιάζει διαφορετική μορφή από εκείνη της ομιλίας. Βλέπουμε ένα **πιο συνεχές και περιοδικό σήμα**, καθώς περιέχει **συνεχές νότες και σταθερό ρυθμό**.

Όμοια με το ερώτημα 2.2, υπολογίζουμε την Ενέργεια βραχέος χρόνου και τον Ρυθμό εναλλαγής προσήμου σε κοινό διάγραμμα με το σήμα μας:

Για 20ms παράθυρο:



Για 50ms Παράθυρο:

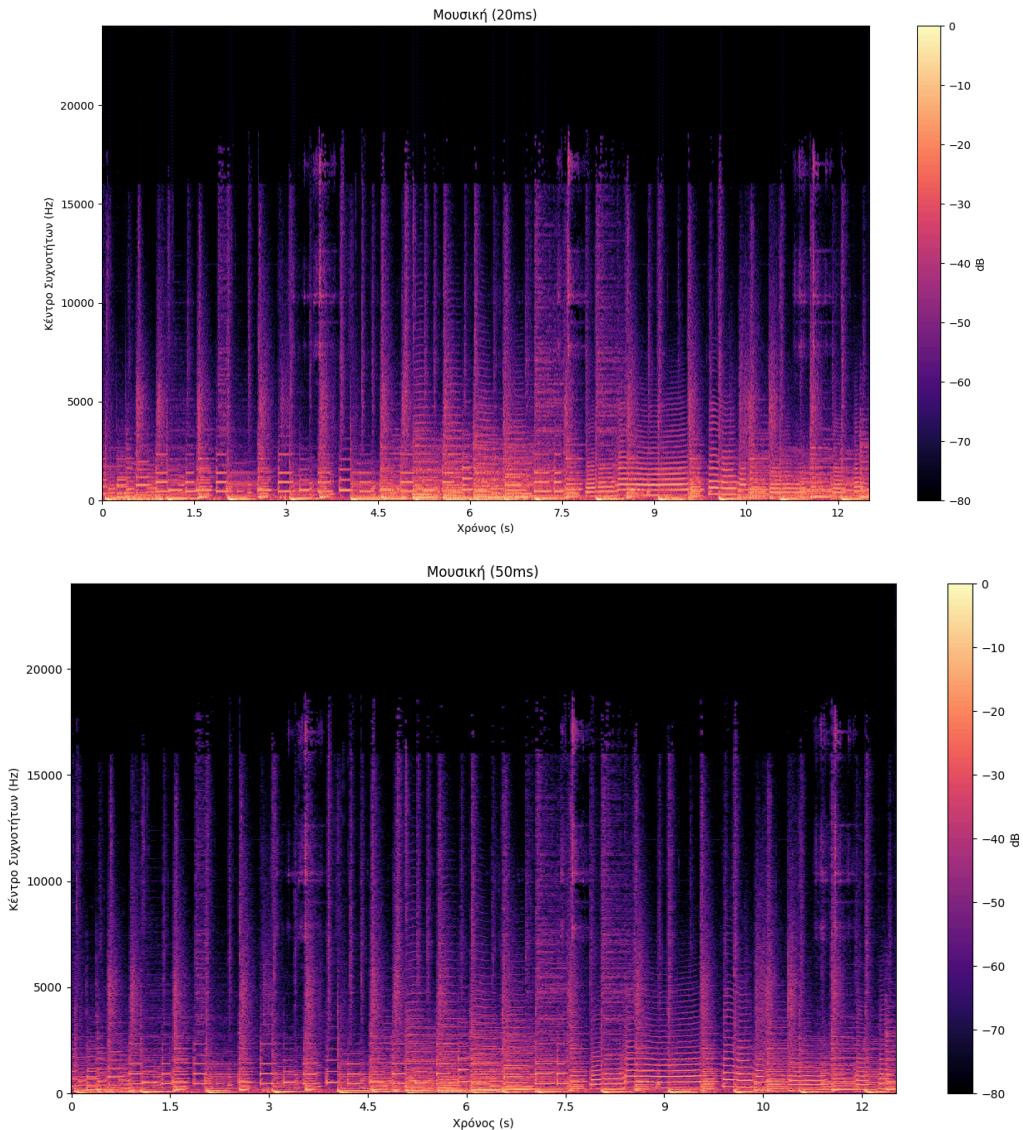


Από τα διαγράμματα ενέργειας βραχέος χρόνου + ZCR, έχουμε τα εξής συμπεράσματα/παρατηρήσεις:

- Παρατηρήσαμε πάλι ότι όσο μεγαλώνουμε το εύρος του παραθύρου τόσο πιο λείες γίνονται οι γραφικές παραστάσεις
- Η ενέργεια βραχέος χρόνου παρουσιάζει μια πιο τακτική εναλλαγή ανάλογα με τον ρυθμό της μουσικής. Στο μουσικό απόσπασμα με σταθερό ρυθμό, παρατηρούμε πως οι κορυφές της ενέργειας επαναλαμβάνονται περιοδικά.
- Ο ρυθμός εναλλαγής προσήμου τείνει να εξαρτάται από το είδος των οργάνων: ένα τονικό μουσικό όργανο (π.χ. έγχορδο) παράγει κυματομορφές με χαμηλό ZCR, ενώ τα κρουστά όργανα αυξάνουν το ZCR, το οποίο επιβεβαιώνει και η βιβλιογραφία.

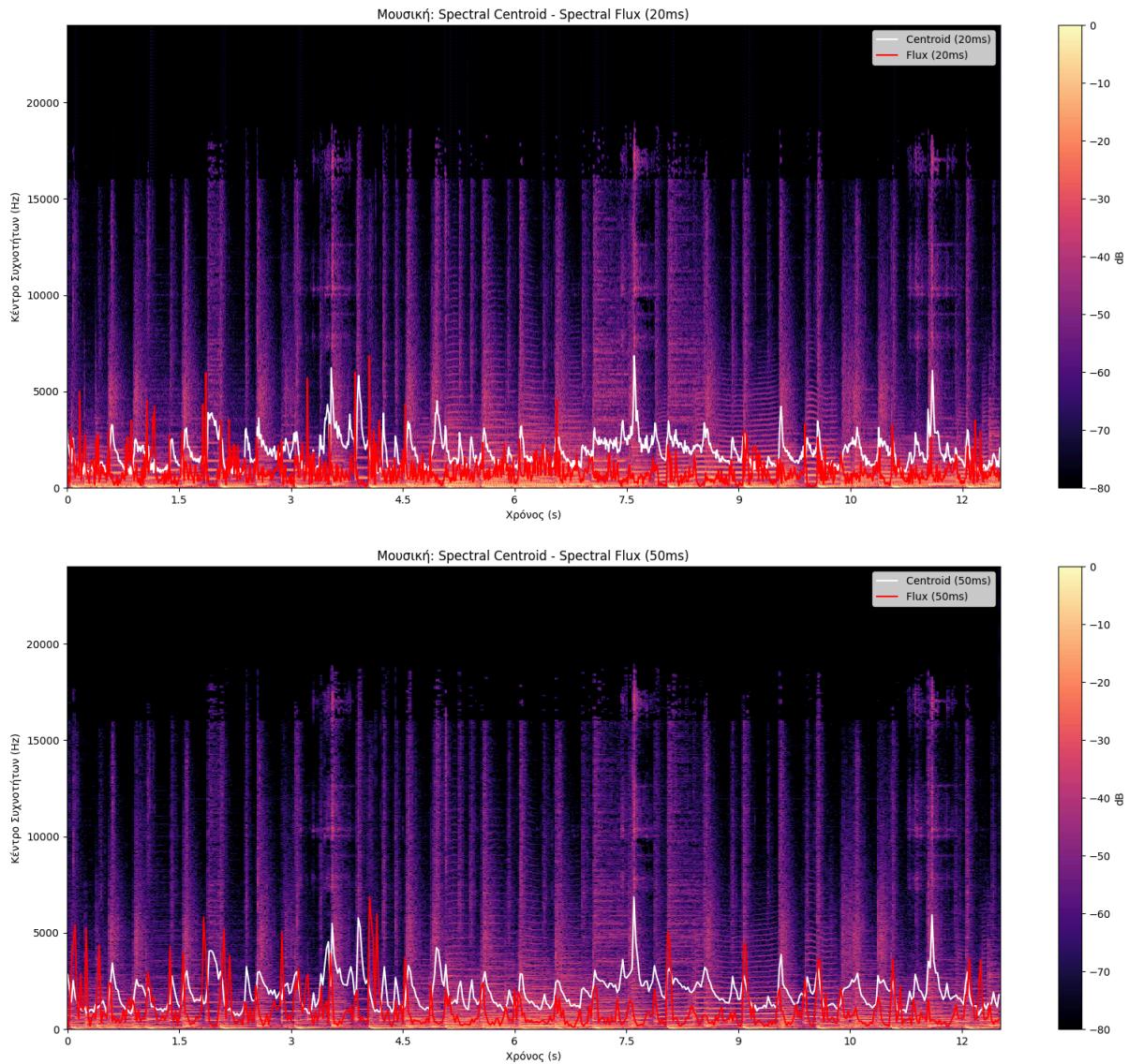
Στην συνέχεια υπολογίζουμε όμοια με το ερώτημα 2.3:

Υπολογισμός STFT (log φασματογράφημα)+ Φασματικό κέντρο (SC), Φασματική ροή (SF) για αρχείο μουσικής σε κοινό διάγραμμα:



Διακρίνουμε διάφορες γραμμές/ζώνες σταθερής συχνότητας που αντιστοιχούν σε αρμονικές των μουσικών νοτών. Ακόμη, κρουστά ή απότομες αλλαγές, εμφανίζονται ως σχήματα που εκτείνονται σε πολλές συχνότητες για μικρό χρόνο. Τελικά, σε σύγκριση με την ομιλία, το μουσικό φάσμα είναι πιο οργανωμένο γύρω από διακριτές νότες και αρμονικές.

Τέλος, αξιοποιώντας το STFT υπολογίσαμε τις τιμές του φασματικού κέντρου και της φασματικής ροής οι οποίες απεικονίζονται στα παρακάτω διαγράμματα:



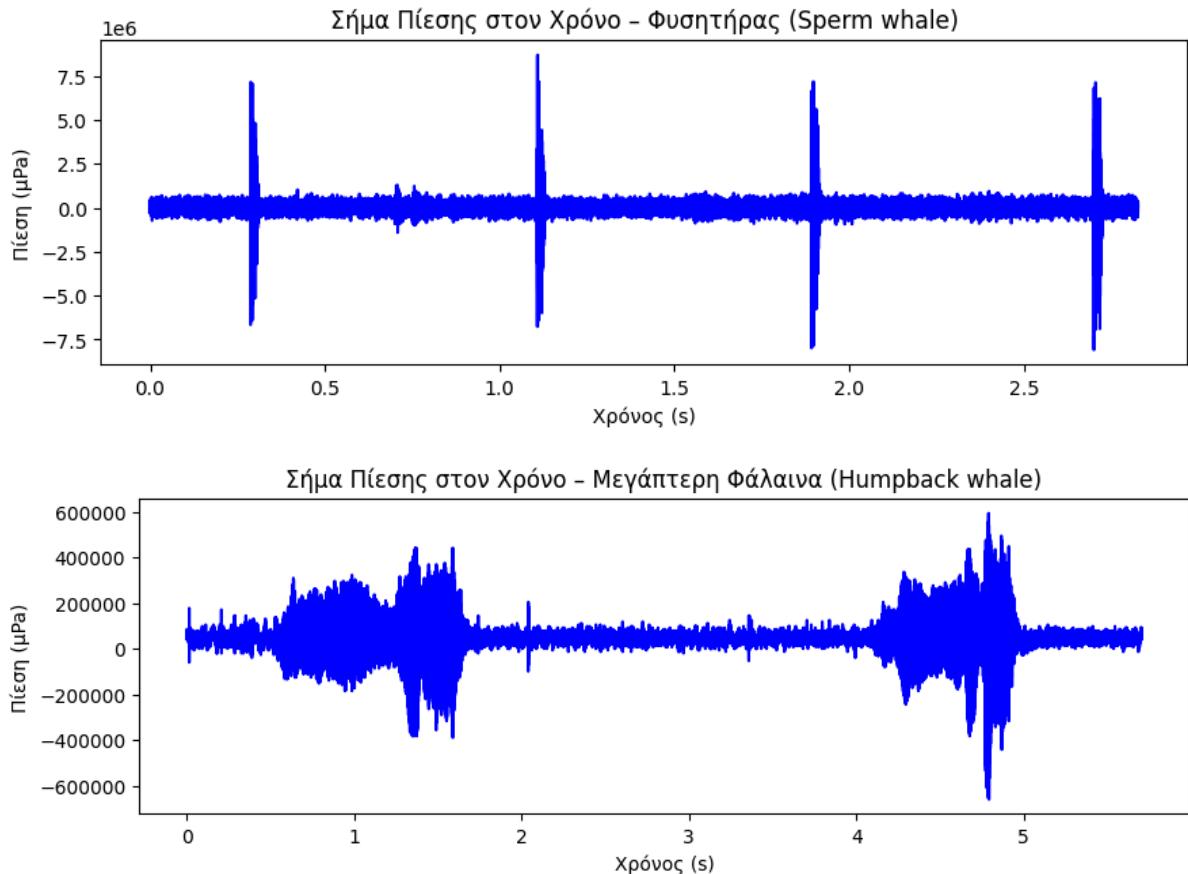
Οπότε, παρατηρούμε πως η φασματική ροή παρουσιάζει κορυφές σε **σημεία αλλαγών δηλαδή στην έναρξη νέων νοτών ή ρυθμικών χτύπων**. Το μουσικό απόσπασμα είναι πλούσιο σε εναλλαγές νοτών με πολλές αντιθέσεις και ρυθμικούς χτύπους, και αυτό έχει ως αποτέλεσμα να υπάρχουν **περιοδικά έντονα σημεία φασματικής ροής** και σε άλλα να παραμένει **σχετικά σταθερή**.

Μέρος 3ο - Ψηφιακή Επεξεργασία και Φασματική Ανάλυση Βιοσημάτων Κητωδών

3.1 Μετατροπή σημάτων Τάσης σε σήματα Πίεσης

Αρχικά φορτώνουμε τα σήματα “85005006.wav” και “9220100Q.wav” με τη βιβλιοθήκη soundfile και την συνάρτηση sf.read(). Η μαθηματική μετατροπή σε πίεση αναφέρεται στην εκφώνηση, και εμείς την υλοποιούμε ως συνάρτηση με τους κατάλληλους παράγοντες. Έπειτα, ακούσαμε τα σήματα που εκπέμπουν τα δύο κητώδη και παρατηρούμε οτι Το σήμα

του φυσητήρα εμφανίζει σύντομους παλμούς (clicks), ενώ η φάλαινα εμφανίζει συνεχή αρμονικά μοτίβα (τραγούδι). Τυπώνουμε επίσης τις συχνότητες δειγματοληψίας για το εκάστοτε σήμα. και με τις γνωστες συναρτήσεις της matplotlib.pyplot εμφανίζουμε τα 2 plots.



3.2 RMS of Pressure and SP Level

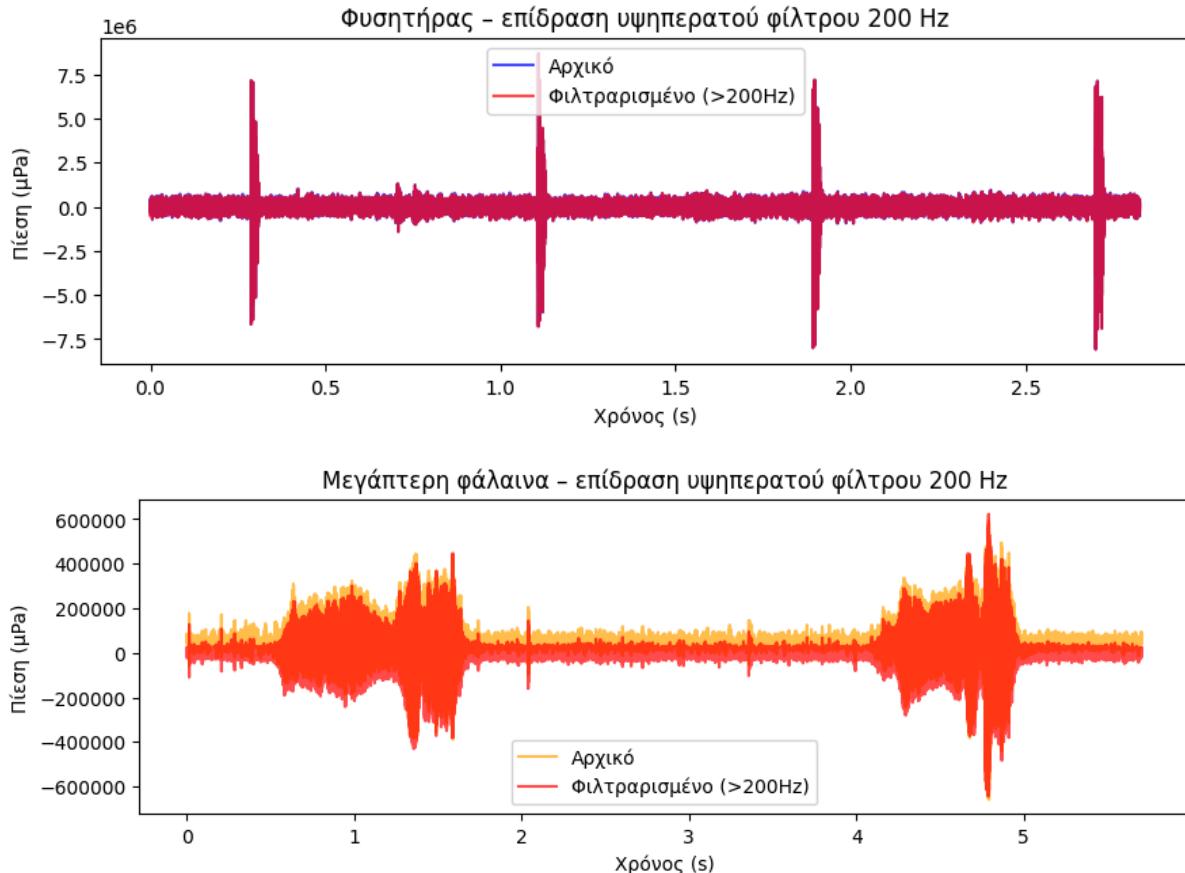
Ακολουθώντας πιστά τους μαθηματικούς ορισμούς της εκφώνησης, θεωρήσαμε συναρτήσεις που θα χρειαστούμε και στην συνέχεια και με την βοήθεια της συνάρτησης np.mean υπολογίζουμε την “ενεργό τιμή” των δύο σημάτων πίεσης, καθώς και το SPL τους, τα οποία για ευκολία παρατίθενται εδώ : p_rms (Sperm whale) = 336416.20 μPa, SPL = 110.54 dB re 1μPa
p_rms (Humpback whale) = 83914.96 μPa, SPL = 98.48 dB re 1μPa

3.3 Parseval's Theorem

Σύμφωνα πάλι με την εκφώνηση και το μαθηματικό μας υπόβαθρο, μέσω των συναρτήσεων np.abs(), np.fft.fft(), np.sum() και με την κατάλληλη κανονικοποίηση οδηγούμαστε στην απόδειξη του θεωρήματος του Parseval. για ενα απο τα δύο σήματα. (Για του λόγου το αληθές . . . 262060.96422657368 = 262060.96422657368 !)

3.4 Butterworth high pass filter

Αρχικά . κάνουμε import from scipy τις συναρτήσεις **butter()**, **filtfilt()**, και σχεδιάζουμε φίλτρο **butter** , υψηπερατό, τάξης 3, με κατώφλι **συχνότητας τα 200Hz**. Έπειτα, παίρνουμε τα αρχικά σήματα Πίεσης των φαλαινών , και τα περνάμε από το φίλτρο. Τέλος για να συγκρίνουμε τις τιμές των σημάτων πίεσης πριν και μετά το φιλτραρισμα, με τις γνωστες συναρτησεις **plt.show()**, **plt.plot()** της βιβλιοθήκης **matplotlib.pyplot** εμφανίζουμε τα 2 μεγέθη αυτά (2 για το κάθε σήμα) , σε κοινό διάγραμμα, τα οποία παρατίθενται παρακάτω.



3.5 Επιρροή κρούσης πασσάλων στην ακοή των δύο ειδών φαλαινών

Αρχικά, φορτώνουμε το σήμα “Pile_driving.wav” με την `soundfile.read()`.

Στην συνέχεια, αναλυτικά ψάχνουμε τη μέθοδο αναζήτησης του αν επηρεάζεται το κάθε είδος φάλαινας:

Για τον φυσητήρα, αρχικά υπολογίζουμε από τις συναρτήσεις **rms** και **spl** που είχαμε δημιουργήσει στα προηγουμενα ερωτηματα, το rms και spl του απλού σήματος πασσάλων εφόσον ο φυσητήρας **ακούει σε όλο το φάσμα**. Έτσι, υπολογίζουμε πως έχει **p_rms** (pile driving) = 245303.66 μPa, SPL = 107.79 dB re 1μPa.

Επομένως ο φυσητήρας επηρεάζεται από τους πασσάλους.

Για την μεγάπτερη, θα πρέπει πρώτα να περάσουμε ένα **bandpass butterworth filter** για συχνότητες 300 Hz και 3000 Hz με τη χρήση της συνάρτησης butter με N=3, $Wn=[cutoff1/(sr_sig/2), cutoff2/(sr_sig/2)]$, btype='bandpass'. Στην συνέχεια, για το φιλτραρισμένο σήμα pressure_pile_filtered, υπολογίζουμε με τις ήδη υπάρχουσες συναρτήσεις τα SPL και RMS. Τελικά λοιπόν έχουμε τα εξής :

$$p_{rms} \text{ filtered (pile driving)} = 15610.45 \mu\text{Pa}, \text{ SPL} = 83.87 \text{ dB re } 1\mu\text{Pa}$$

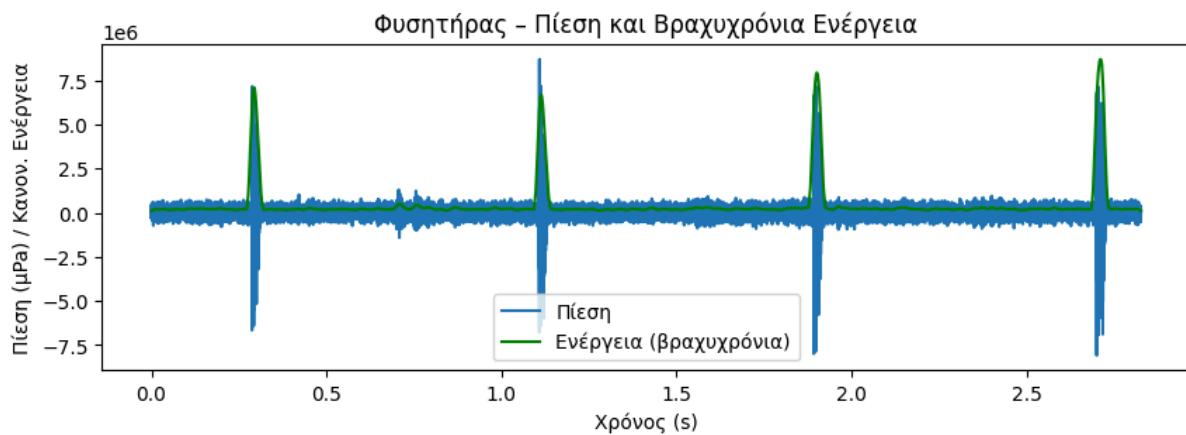
Επομένως δεν επηράζεται η μεγάπτερη αφού ενοχλείται πάνω από 100dB.

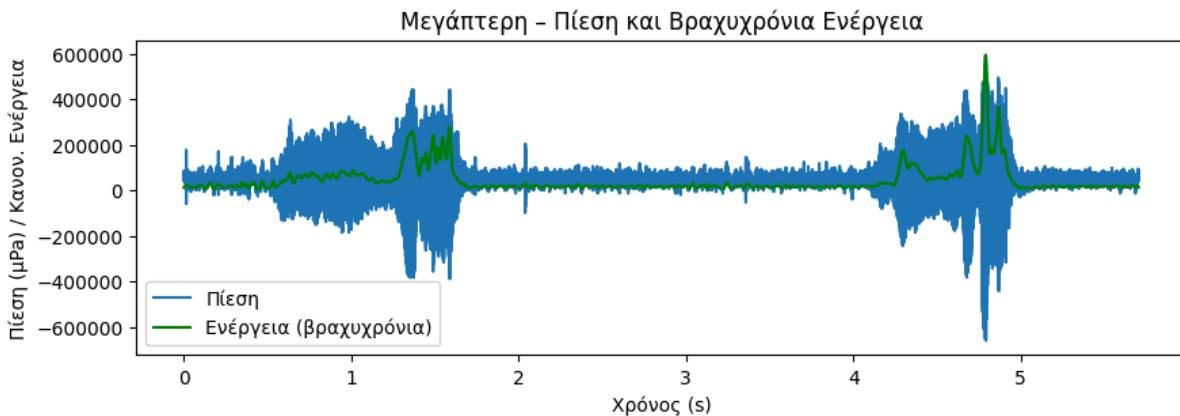
3.6 Short - Time Energy

Όπως έχουμε αναλύσει περαιτέρω στο Μέρος 2, Η **βραχυχρόνια ενέργεια** ενός σήματος είναι μια χρονικά μεταβαλλόμενη ποσότητα που δείχνει πόση ενέργεια έχει το σήμα σε ένα μικρό “παράθυρο” χρόνου, καθώς αυτό κινείται κατά μήκος του σήματος.

Με χρήση ενός παραθύρου Hamming και τη συνέλιξη του τετραγώνου του, υπολογίζουμε με τη συνάρτηση short_time_energy για την ενέργεια βραχέος χρόνου.

Παίρνουμε παράθυρο 30ms, υπολογίζουμε το μήκος των frames για κάθε είδος, και τέλος την short_time_energy τους. Στην συνέχεια, plottάρουμε σε κοινό διάγραμμα πίεσης-χρόνου και βραχυχρόνιας ενέργειας - χρόνου για κάθε είδος:



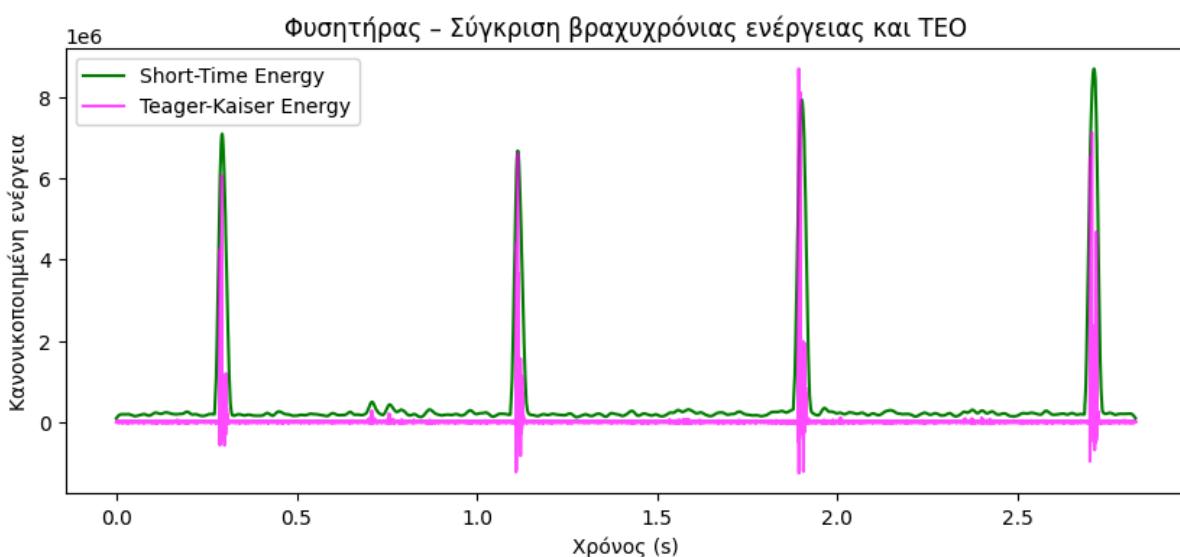


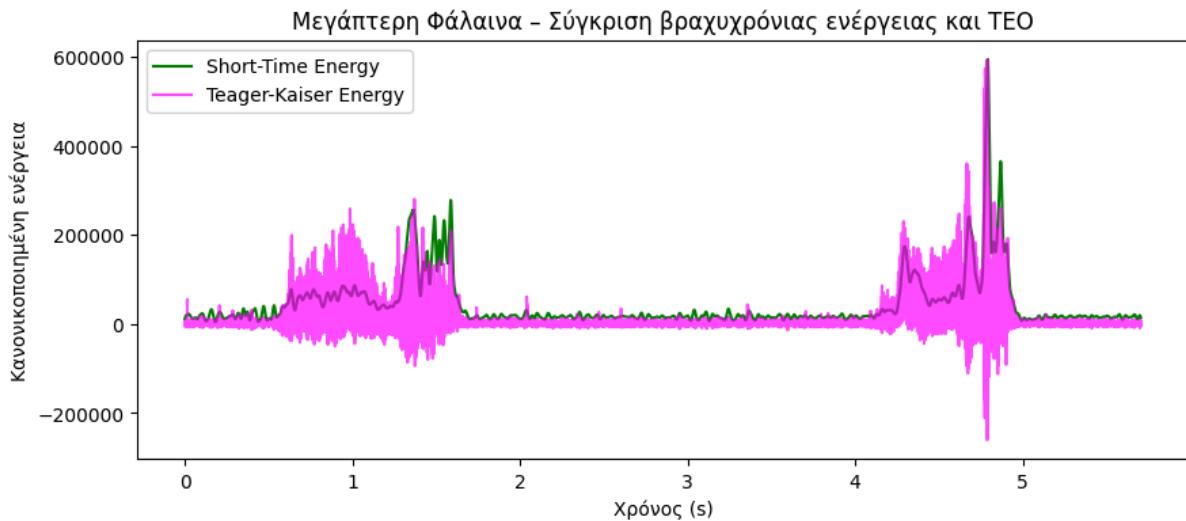
Για τον φυσητήρα, παρατηρούμε πως η βραχυχρόνια ενέργεια παρουσιάζει απότομες κορυφές στα σημεία όπου υπάρχουν clicks, και θα είναι σχεδόν μηδενική (πολύ χαμηλή) στα ενδιάμεσα διαστήματα σιωπής. Αυτό διευκολύνει τον εντοπισμό των clicks. Για τη μεγάπτερη φάλαινα, το τραγούδι της έχει μια πιο συνεχόμενη παρουσία ήχου. Η βραχυχρόνια ενέργεια ακολουθεί τις δυναμικές του τραγουδιού – όπου το τραγούδι είναι πιο έντονο (π.χ. σε κορύφωση ενός φωνήματος) θα έχει υψηλότερη ενέργεια, ενώ μπορεί να πέφτει σε στιγμές “παύσης” ή μετάβασης.

3.7 Ενέργεια Teager - Kaiser

Ο ενεργειακός τελεστής Teager-Kaiser (TEO) είναι ένας μη-γραμμικός τελεστής που υπολογίζει την “ενέργεια” ενός ταλαντωτή απευθείας από το σήμα, τονίζοντας την τοπική ενέργεια στο χρόνο.

Εφαρμόσαμε τον ενεργειακό τελεστή Teager-Kaiser Operator στο διακριτού χρόνου σήμα πίεσης-χρόνου σύμφωνα με τη δοσμένη εξίσωση (10) για κάθε φάλαινα, όπου για την ολίσθηση χρησιμοποιήσαμε κανονικοποίηση σε κοινούς άξονες και πήραμε τα ακόλουθα κοινά διαγράμματα short time energy - χρόνου και TE-O – χρόνου για κάθε ένα από τα δύο βιοσήματα αντίστοιχα.





Παρατηρούμε πως ο ΤΕΟ για τον φυσητήρα δίνει επίσης αιχμές στα clicks, με ακόμα πιο απότομη κορύφωση και λιγότερη “ουρά” συγκριτικά με την απλή βραχυχρόνια ενέργεια. Αντό σημαίνει ότι τονίζει περισσότερο τα στιγμιαία γεγονότα (τα clicks) και να αγνοεί καλύτερα το υπόβαθρο. Για τη μεγάπτερη φάλαινα, που το σήμα της είναι πιο ημιομαλό, ο ΤΕΟ αναδεικνύει συγκεκριμένες μεταβολές στο σήμα (π.χ. αρχές ή τέλη φωνημάτων) πιο έντονα από την απλή ενέργεια.

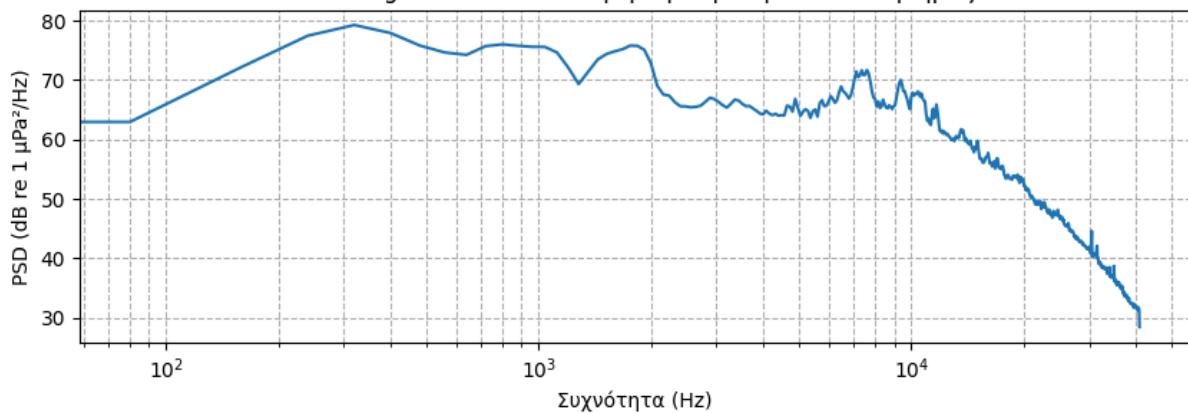
3.8 Σχεδίαση περιοδόγραμματος του Welch σε λογαριθμική κλίμακα

Η φασματική πυκνότητα ισχύος (PSD) δείχνει πώς κατανέμεται η ισχύς ενός σήματος ως προς τη συχνότητα. Είναι χρήσιμη για να αναγνωρίσουμε τα φάσματα των σημάτων μας. Θα την υπολογίσουμε με τη μέθοδο Welch, η οποία ουσιαστικά υπολογίζει το μέσο όρο του περιοδογράμματος σε τμήματα του σήματος για να δώσει ένα πιο ομαλό φάσμα.

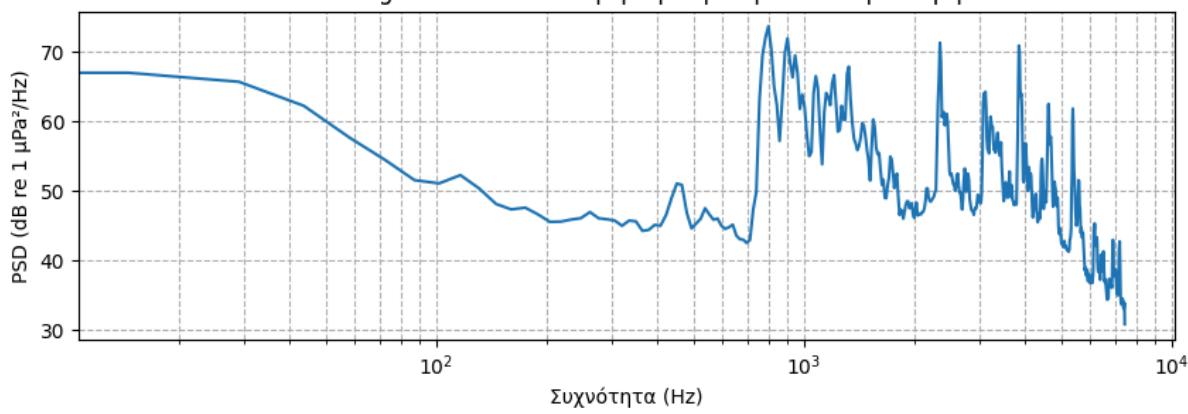
Για την υλοποίηση, χρησιμοποιήσαμε τη συνάρτηση `signal.scipy.welch()`. Εφαρμόσαμε στη συνέχεια τη μέθοδο αυτή στα σήματα πίεσης (σε μPa) των δύο βιοσημάτων (φυσητήρα και μεγαπτερης) και την κάναμε σχετική με $1 \mu\text{Pa}^2/\text{Hz}$.

Σχεδιάσαμε σε λογαριθμική κλίμακα τον άξονα της συχνότητας, μέσω του `plt.semilogx()` και είχαμε τα έξης διαγράμματα:

Periodogram Welch σε Λογαριθμική Κλίμακα: Φυσητήρας

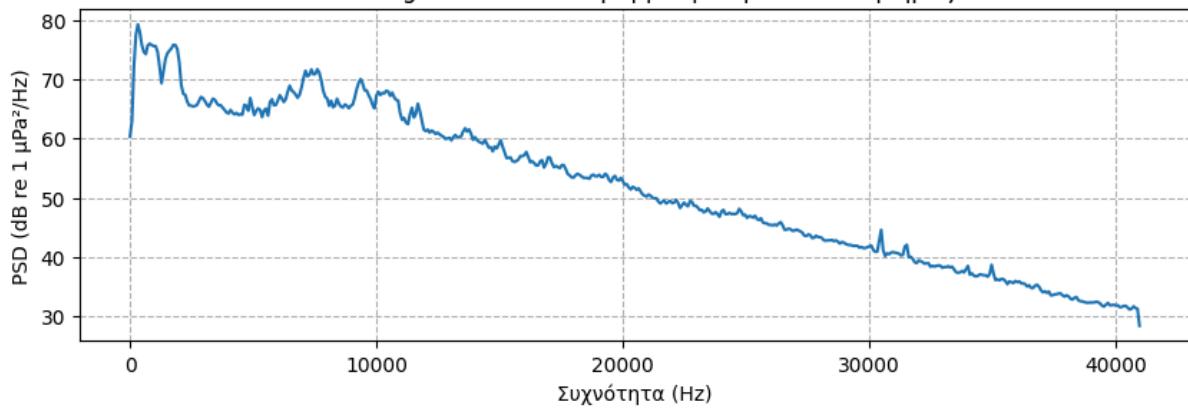


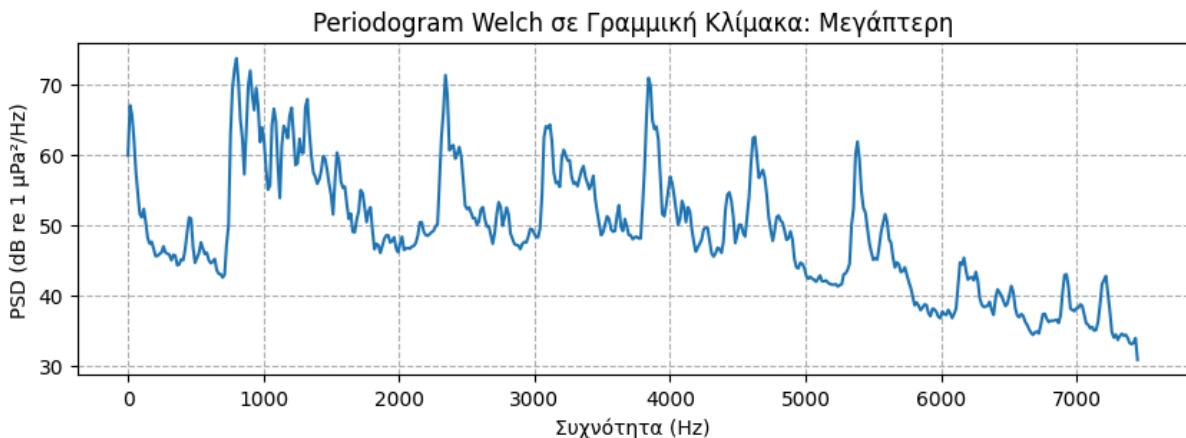
Periodogram Welch σε Λογαριθμική Κλίμακα: Μεγάπτερη



Παρατηρούμε πως γενικότερα δεν μπορούμε να εξάγουμε απόλυτα ακριβή συμπεράσματα από τις γραφικές, οπότε αναλύουμε τον άξονα χ σε γραμμική κλίμακα με απλό plot, και έχουμε τα εξής αποτελέσματα:

Periodogram Welch σε Γραμμική Κλίμακα: Φυσητήρας



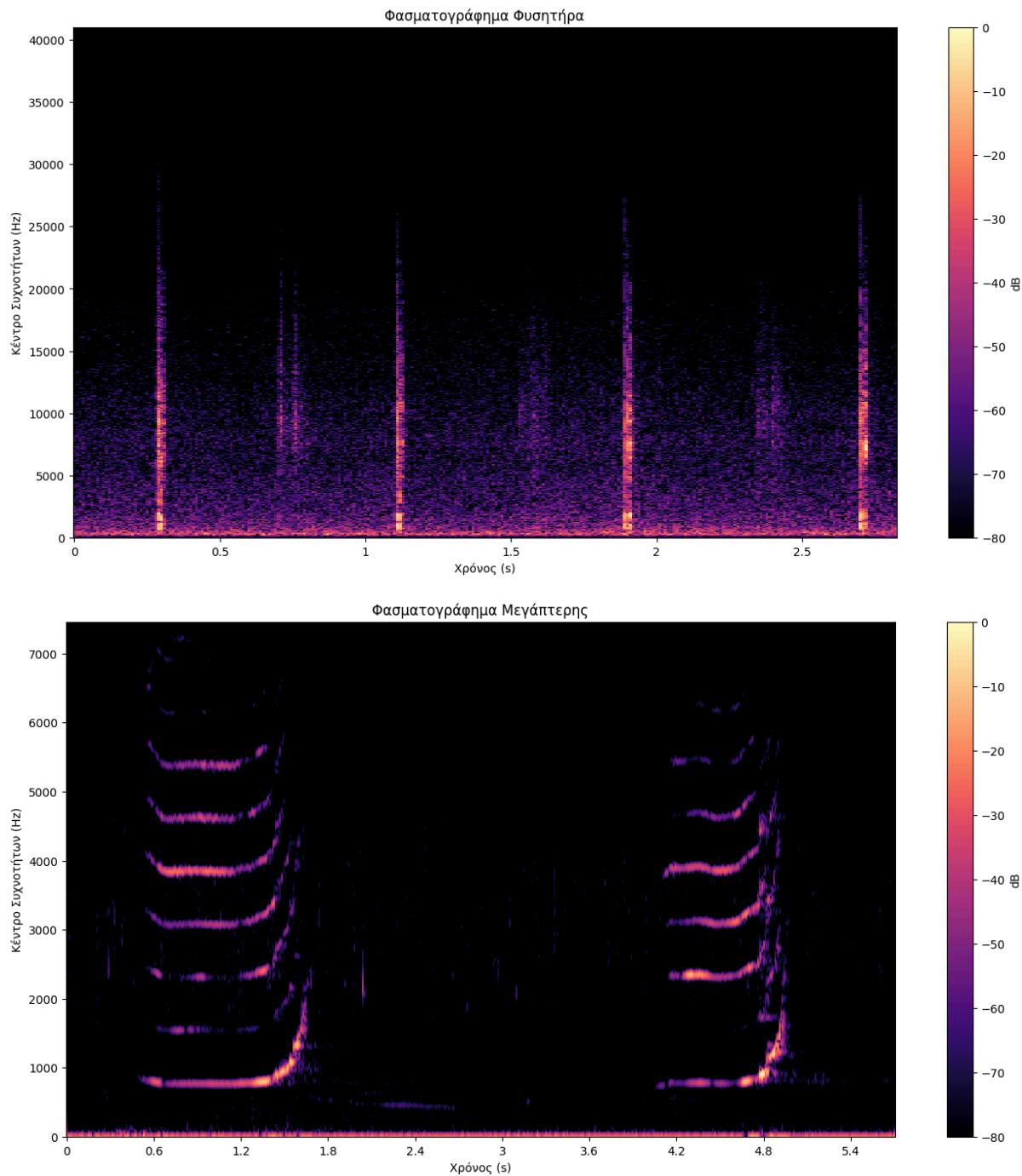


Παρατηρήσεις:

- Γενικά, η **φασματική πυκνότητα ισχύος (PSD)** δείχνει πώς κατανέμεται η ισχύς ενός σήματος ως προς τη **συχνότητα**. Είναι χρήσιμη για να αναγνωρίσουμε τα φάσματα των σημάτων μας. Την υπολογίζαμε με τη μέθοδο Welch, η οποία ουσιαστικά υπολογίζει το μέσο όρο του periodogram σε τμήματα του σήματος για να δώσει ένα πιο ομαλό φάσμα.
- Δοκιμάσαμε να φτιάξουμε διαγράμματα με γραμμική και λογαριθμική κλίμακα συχνότητας, και καταλήξαμε πως η γραμμική μας προσφέρει περισσότερες παρατηρήσεις για να αναλύσουμε.
- **Φυσητήρας:** Βλέπουμε υψηλή ενέργεια μέχρι το 1kHz, καθώς τα clicks έχουν ευρύ φάσμα που επεκτείνεται από εκατοντάδες Hz έως και πολλαπλά kHz (ακόμα και δεκάδες kHz), ενώ στη συνέχεια σταδιακά μειώνεται μέχρι να σβήσει.
- **Μεγάπτερη φάλαινα:** Το ηχητικό της σήμα έχει ισχυρά στοιχεία στις συχνότητες 1000Hz με 8kHz, με αρμονικές. Το PSD δείχνει διάφορες κορυφές αρμονικών, στα σημεία πχ 900Hz, 2400Hz, 3000Hz κλπ, και γενικά όσο αυξάνεται η συχνότητα φθίνει η ενέργεια.

3.9 Σχεδιασμός Φασματογραφήματος (Spectrogram) των δύο βιοσημάτων

Σε αυτό το ερώτημα για κάθε φάλαινα δημιουργήσαμε τη συνάρτηση `compute_spectrogram()` που υπολογίζει το μετασχηματισμό STFT των σημάτων τους με το `librosa.stft()`, αλλάζει σε κλίμακα dB το τετράγωνο του μετ/μου με τη συνάρτηση `librosa.amplitude_to_db()` και υπολογίζει τα `hop lengths` = frame length * hop fraction. Τέλος, με τις γνωστές συναρτήσεις τυπώσαμε τα φασματογραφήματα των βιοσημάτων, χρησιμοποιώντας την `librosa.display.specshow()`, τα οποία φαίνονται παρακάτω.



Παρατηρούμε πως:

- Το **φασματογράφημα** ενός σήματος δείχνει πώς το φάσμα μεταβάλλεται με τον χρόνο. Είναι ουσιαστικά ένας χάρτης χρόνου-συχνότητας : στον ένα άξονα ο χρόνος, στον άλλο η συχνότητα, και με χρωματική κλίμακα η ένταση (σε dB)
- **Φυσητήρας:** Βλέπουμε τα clicks ως κατακόρυφες “λωρίδες” ενέργειας που καλύπτουν ευρύ φάσμα (από χαμηλά έως αρκετά υψηλά Hz). Μεταξύ τους υπάρχει ησυχία, καθώς και από το ηχητικό δεν ακούμε άλλο ίχο παρά μερικό background effect. Ακόμη, αυτές οι λωρίδες ενέργειας κρατάνε πολύ λίγο χρόνο.
- **Μεγάπτερη φάλαινα:** Βλέπουμε έναν πιο συνεχή και αρμονικό σχηματισμό. Βλέπουμε για κάθε γεγονός που ακούγεται η φάλαινα πολλές παράλληλες φωτεινές

γραμμές στο ίδιο χρονικό διάστημα, οι οποίες είναι σταθερές και διαρκούν αρκετά. Αυτές αποτελούν τις αρμονικές του τραγουδιού της φάλαινας.

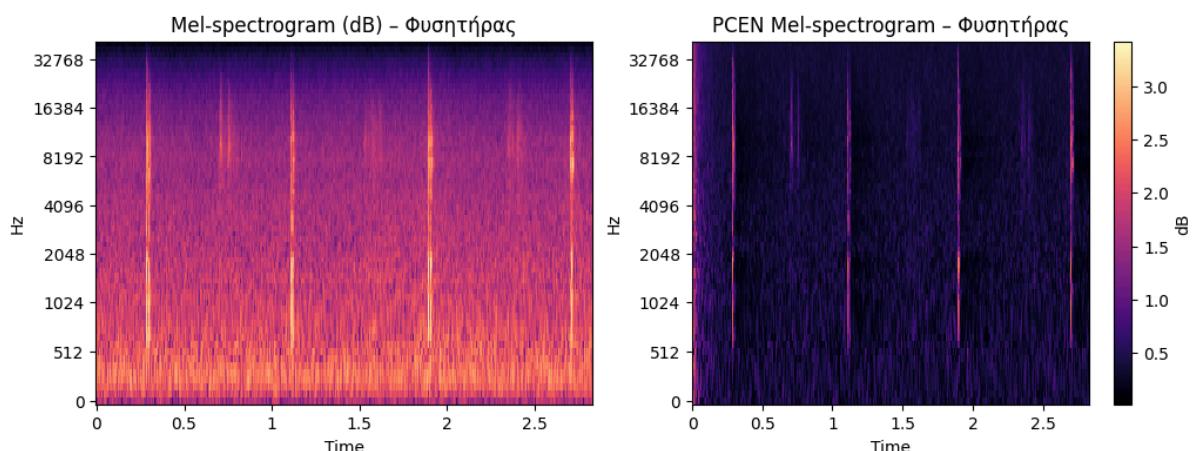
3.10 Εφαρμογή της μεθόδου PCEN

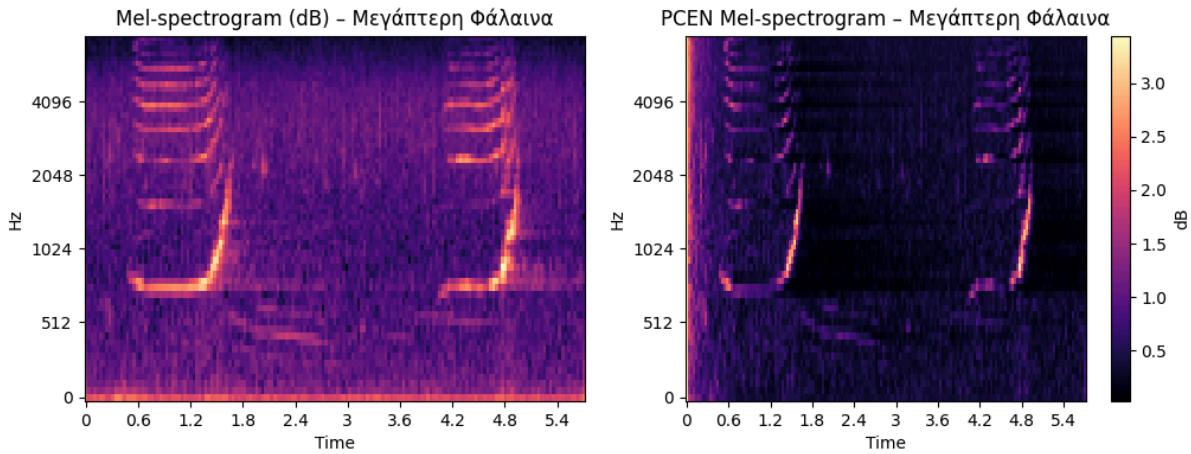
Τέλος, θα εξετάσουμε τα **mel-spectrograms** των σημάτων και θα εφαρμόσουμε **Per-Channel Energy Normalization (PCEN)**. Για την υλοποίηση του συγκεκριμένου ερωτήματος ακολουθήσαμε πιστή διαδικασία υπολογίζοντας αρχικα το mel-spectrogram με `n_fft = 1024, hop_length = 512, n_mels = 64` και `power = 1` για ενεργεια για την συναρτηση `librosa.feature.melspectrogram()`. Επειτα μετατρέπουμε σε dB με την βοήθεια της συνάρτησης `librosa.amplitude_to_db()`, και επαναλαμβάνουμε την ίδια διαδικασία για `librosa.pcen()`, και πάλι `librosa.amplitude_to_db()`. Τέλος, με τις γνωστές συναρτήσεις της βιβλιοθήκης `matplotlib.pyplot` και της συνάρτησης `librosa.display.specshow()` κάνουμε την σύγκριση για κάθε σήμα .

Mel-spectrogram: Είναι φασματογράφημα όπου ο άξονας συχνότητας έχει μετασχηματιστεί σε mel κλίμακα (μια κλίμακα που προσεγγίζει τον τρόπο που ακούει τις συχνότητες ο άνθρωπος – με υψηλότερη ανάλυση σε χαμηλές συχνότητες και χαμηλότερη σε υψηλές). Συνήθως μειώνουμε και τον αριθμό των συχνοτικών καναλιών (π.χ. 40 ή 64 mel bins).

PCEN: Είναι μια τεχνική ομαλοποίησης που βελτιώνει την αντίθεση ανάμεσα σε ήχους που χρειαζόμαστε (ενδιαφέροντος), π.χ. τα clicks ή τα φωνήματα φαλαινών και τον “background” θόρυβο.

Τα Διαγράμματα που σχεδιάσαμε είναι τα εξής:





Συμπεραίνουμε λοιπόν πως **το φασματογράφημα χωρίς τη μέθοδο PCEN εμφανίζει το πραγματικό περιεχόμενο του σήματος με τον όποιο θόρυβο αυτό περιέχει.**

Παρατηρούμε, λοιπόν ότι εμφανίζει ποικίλες διακυμάνσεις στην ένταση λόγω των διαφορετικών επιπέδων **σήματος** και **θορύβου**. Ωστόσο τα βασικά χαρακτηριστικά είναι ασαφή και δεν αναδεικνύονται.

Τέλος, παρατηρώντας τα **mel-φασματογραφήματα** πριν και μετά την εφαρμογή της μεθόδου PCEN, παρατηρούμε ότι με τη PCEN **ενισχύεται η αντίθεση των χαρακτηριστικών** του φασματογραφήματος, καθιστώντας έτσι πιο εμφανή. Αυτό συμβαίνει διότι μειώνει την επίδραση του background noise. Τέλος στο PCEN εμφανίζεται **κανονικοποιημένη και ομαδικοποιημένη** η ενέργεια κατά μήκος των συχνοτήτων.