

Εισαγωγή στην Ψηφιακή Επεξεργασία Σημάτων με

Python και Εφαρμογές σε Ακουστικά Σήματα

1η Εργαστηριακή Άσκηση

Ον/μο: Γεώργιος Τζαμουράνης ΑΜ: 03121141

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

1.1 Κατασκευάζουμε τους τόνους για τα 10 πλήκτρα με τις συναρτήσεις που φαίνονται παρακάτω και με τη βοήθεια της συνάρτησης `display(Audio())` και με συχνότητα δειγματοληψίας **8192Hz**.

$d0 = \text{np.sin}(0.7217 * n) + \text{np.sin}(1.0247 * n)$

$d1 = \text{np.sin}(0.5346 * n) + \text{np.sin}(0.9273 * n)$

$d2 = \text{np.sin}(0.5346 * n) + \text{np.sin}(1.0247 * n)$

$d3 = \text{np.sin}(0.5346 * n) + \text{np.sin}(1.1328 * n)$

$d4 = \text{np.sin}(0.5906 * n) + \text{np.sin}(0.9273 * n)$

$d5 = \text{np.sin}(0.5906 * n) + \text{np.sin}(1.0247 * n)$

$d6 = \text{np.sin}(0.5906 * n) + \text{np.sin}(1.1328 * n)$

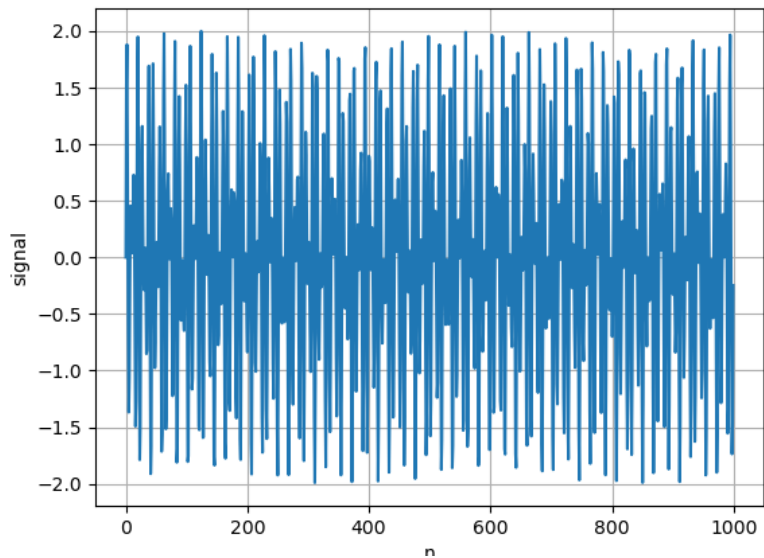
$d7 = \text{np.sin}(0.6535 * n) + \text{np.sin}(0.9273 * n)$

$d8 = \text{np.sin}(0.6535 * n) + \text{np.sin}(1.0247 * n)$

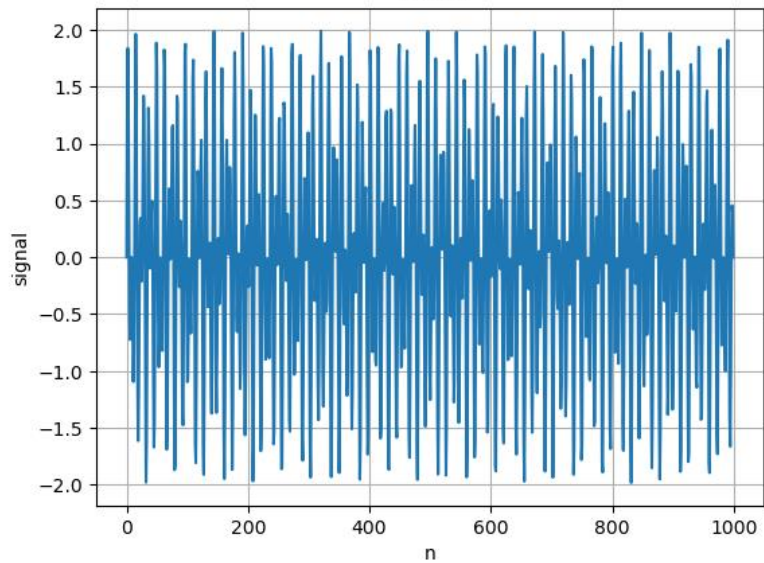
$d9 = \text{np.sin}(0.6535 * n) + \text{np.sin}(1.1328 * n)$

Οι γραφική παράσταση για κάθε τόνο φαίνεται παρακάτω:

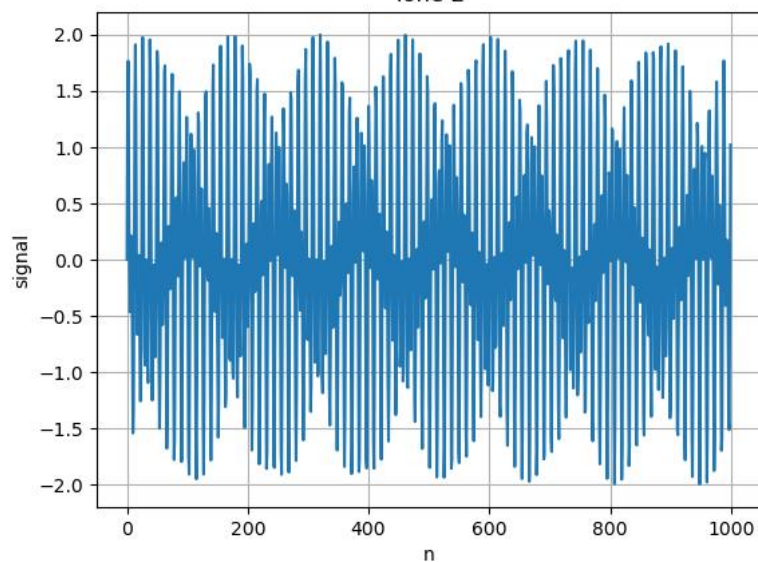
Tone 0



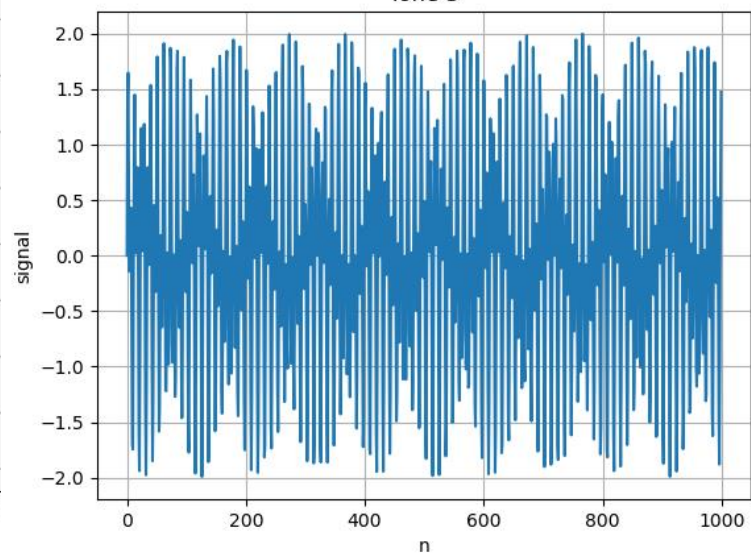
Tone 1



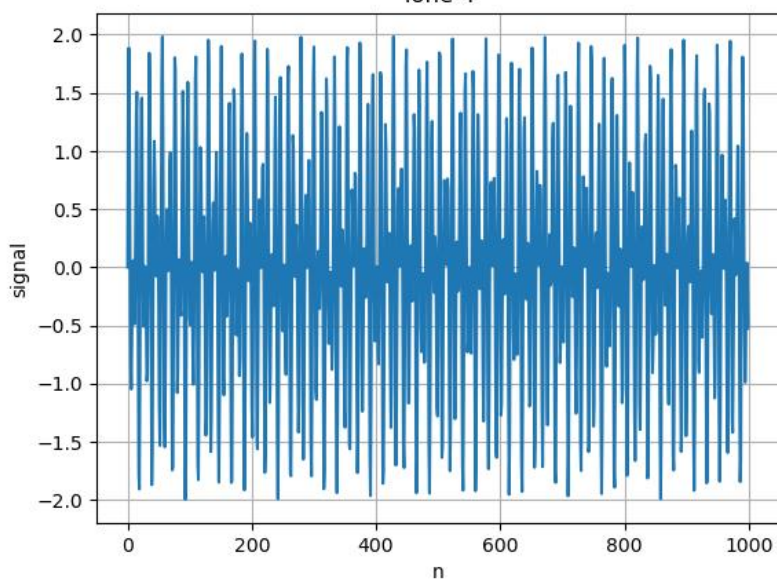
Tone 2



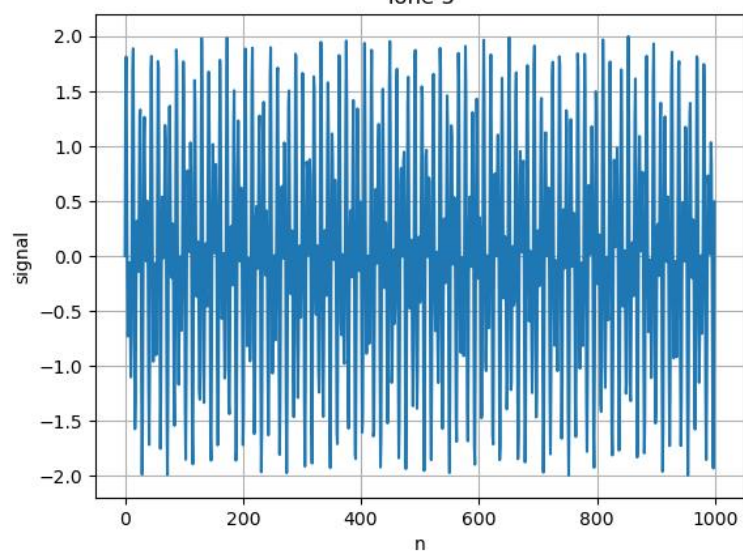
Tone 3

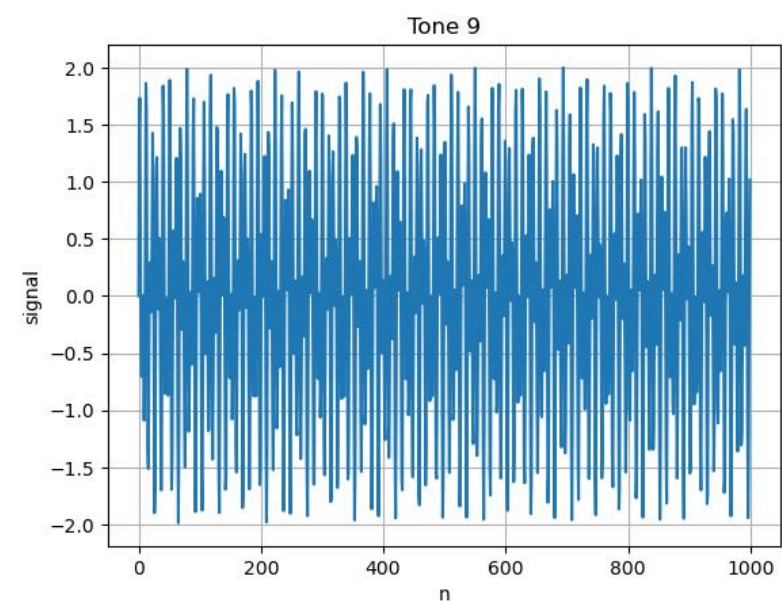
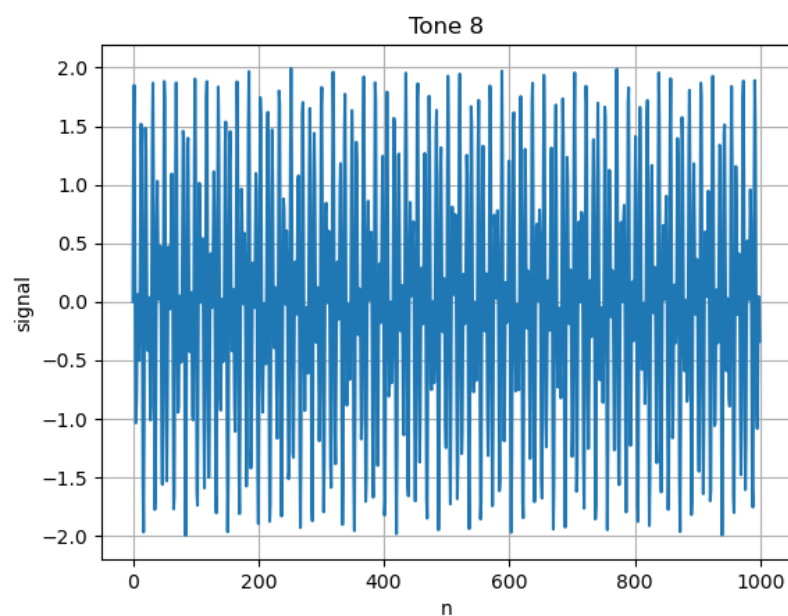
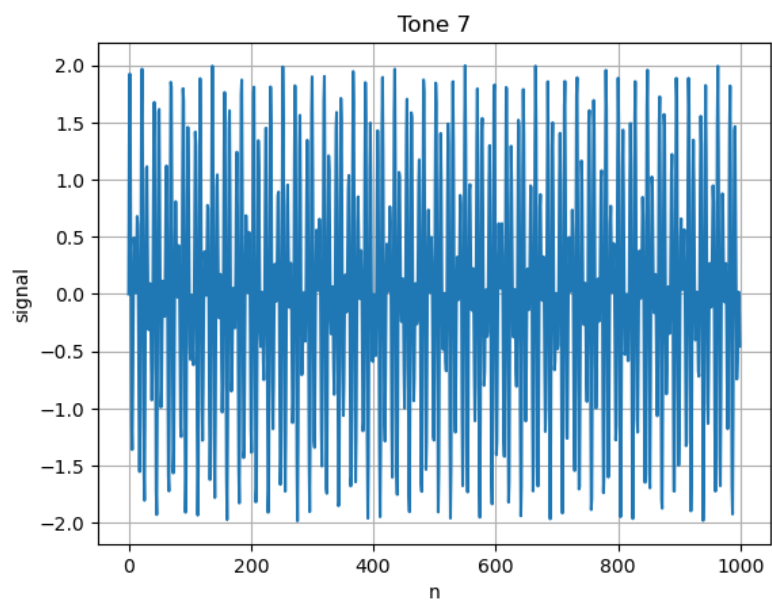
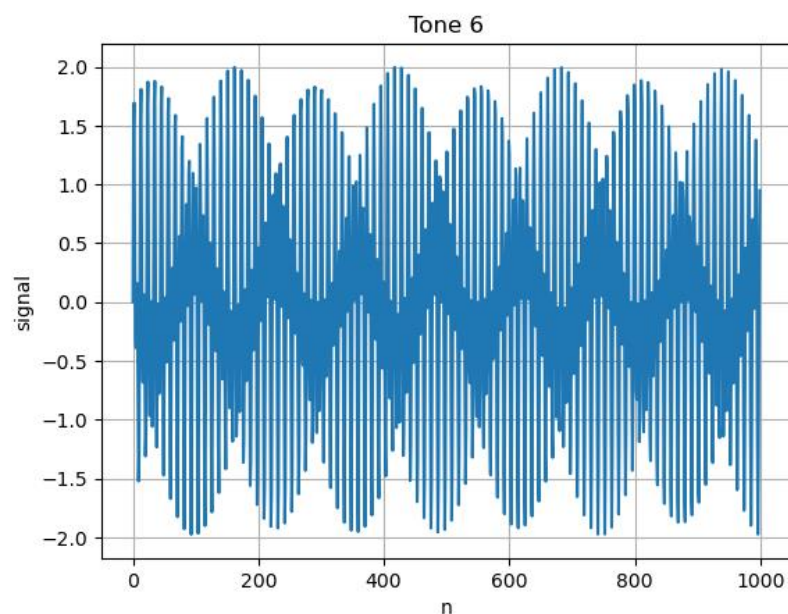


Tone 4

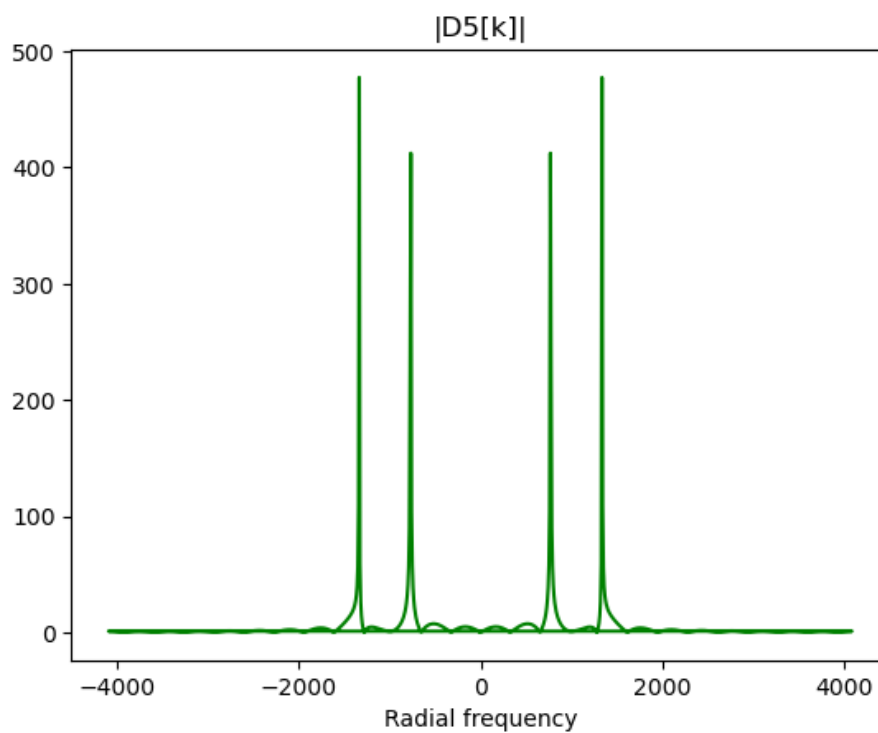
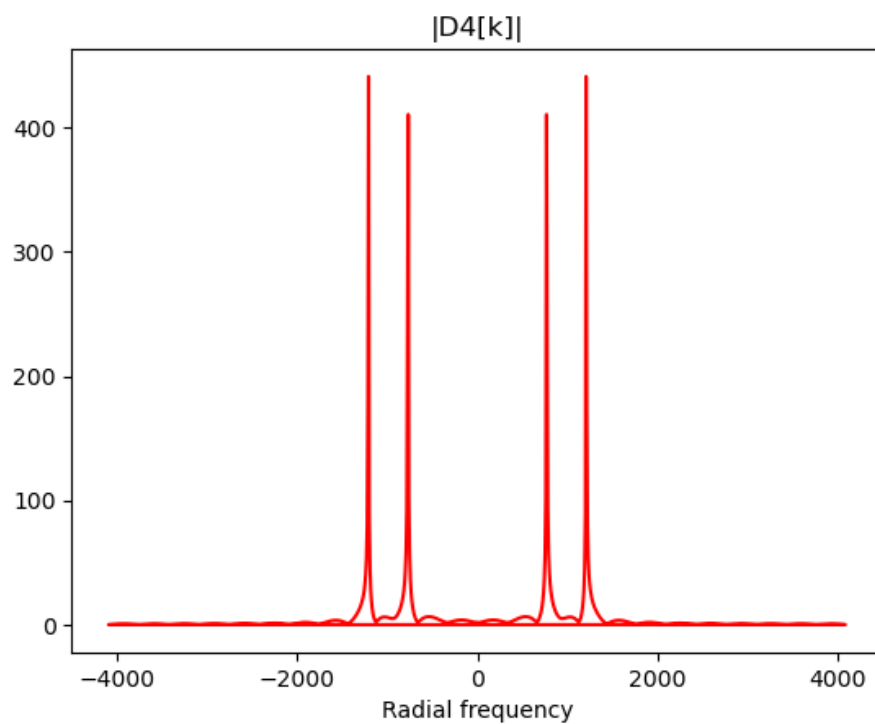


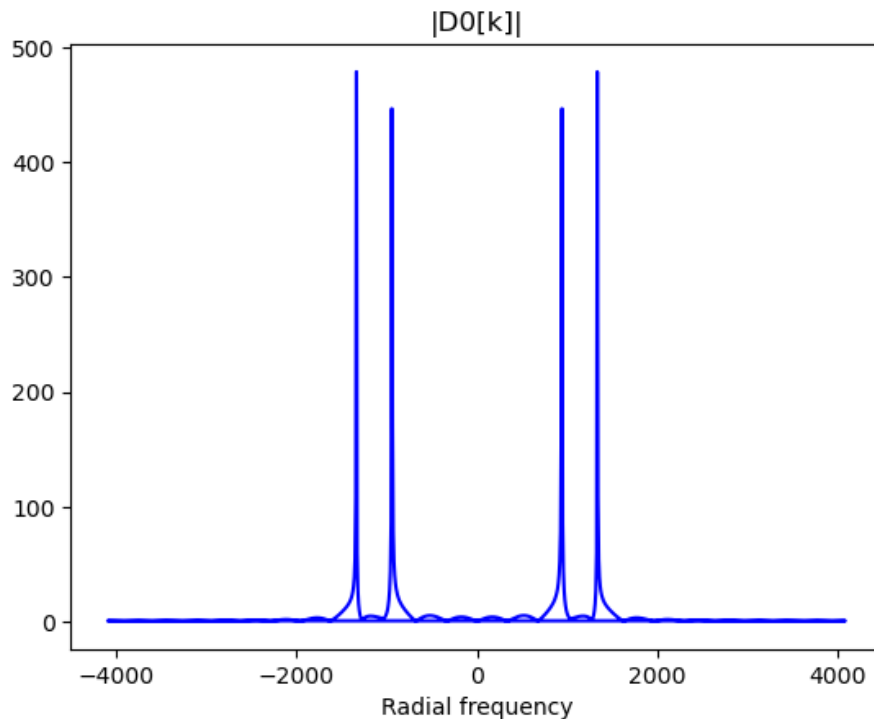
Tone 5





1.2 Σε αυτό το ερώτημα υπολογίσαμε τον DFT με τη βοήθεια της συνάρτησης **fft.fft()** και με τη **fft.fftfreq()** ρυθμίσαμε τις συχνότητες, για τους **τόνους 4, 5 και 0** και τυπώσαμε τις γραφικές παραστάσεις των μέτρων τους. Στον μετασχηματισμό χρησιμοποιήσαμε **N = 1024 σημεία**. Στα διαγράμματα παρατηρούμε **4 peaks**, δηλαδή 2 και 2 συμμετρικά τα οποία προκαλούνται λόγω των δύο ημιτόνων (2 διαφορετικών συχνοτήτων) σε κάθε τόνο.

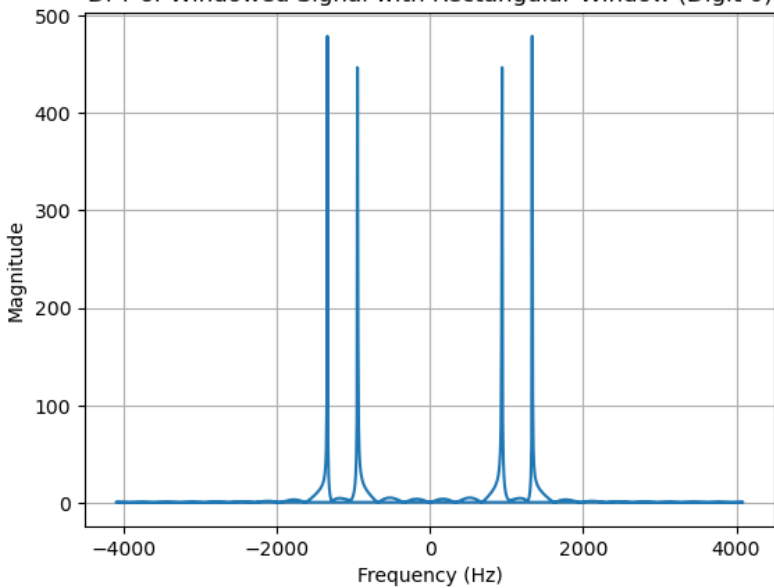




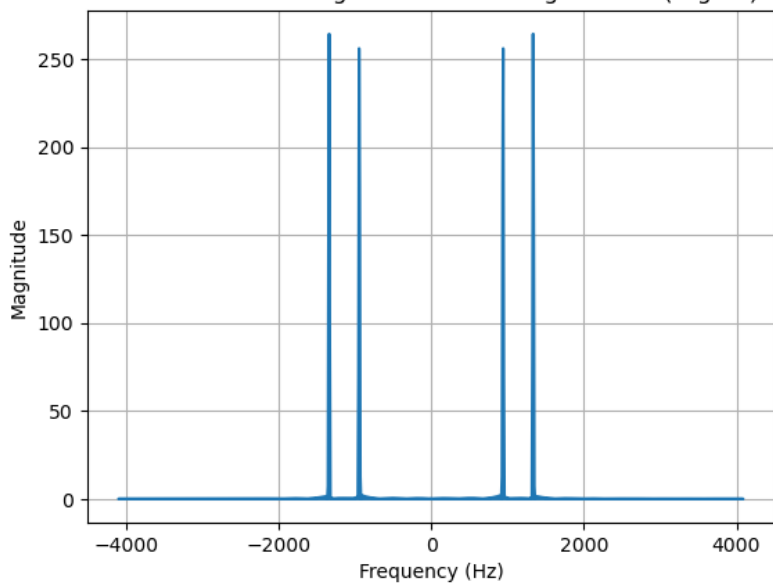
1.3 Για το ερώτημα αυτό χρησιμοποιήσαμε τον **AM: 03121141**, από το οποίο δημιουργήσαμε ένα σήμα διαδοχικών τόνων που το αποθηκεύσαμε σε αρχείο με όνομα **“tone_sequence.wav”** με τη χρήση της εντολής **write()**. Μεταξύ των τόνων βάλαμε 100 μηδενικά και με χρήση την συνάρτησης **concatenate()** συνδέσαμε τα επιμέρους μέρη του σήματος. Τέλος ακούσαμε το αποτέλεσμα με την εντολή **audio()**.

1.4 Στη συνέχεια πήραμε κάθε ψηφίο από τον AM και εφαρμόσαμε **τετραγωνικό** και **hamming παράθυρο μήκους 1000**. Στην υλοποίηση χρησιμοποιήσαμε τα κατάλληλα λεξικά ώστε να αποθηκεύουμε τα αποτελέσματα των υπολογισμών. Μετά τη παραθύρωση υπολογίσαμε τον μετ/μο Fourier των παραθυρομένων σημάτων με την συνάρτηση **fft.fft()** και πλοτάρουμε το μέτρο του αξιοποιώντας την εντολή **fft.fftfreq()** ώστε να έχουμε σωστά τις συχνότητες στον χ-άξονα. Τα διαγράμματα φαίνονται παρακάτω (δεν παραθέτουμε τα επαναλαμβανόμενα ψηφία στην αναφορά για ευνόητους λόγους) .

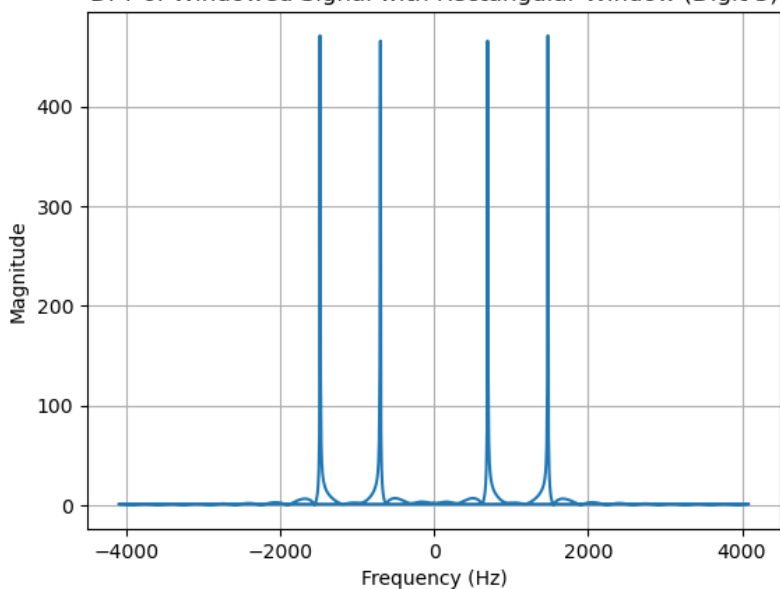
DFT of Windowed Signal with Rectangular Window (Digit 0)



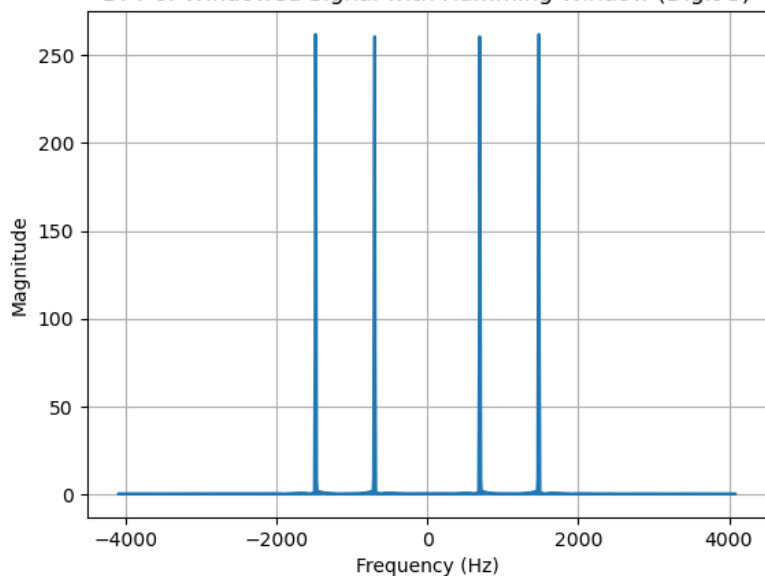
DFT of Windowed Signal with Hamming Window (Digit 0)



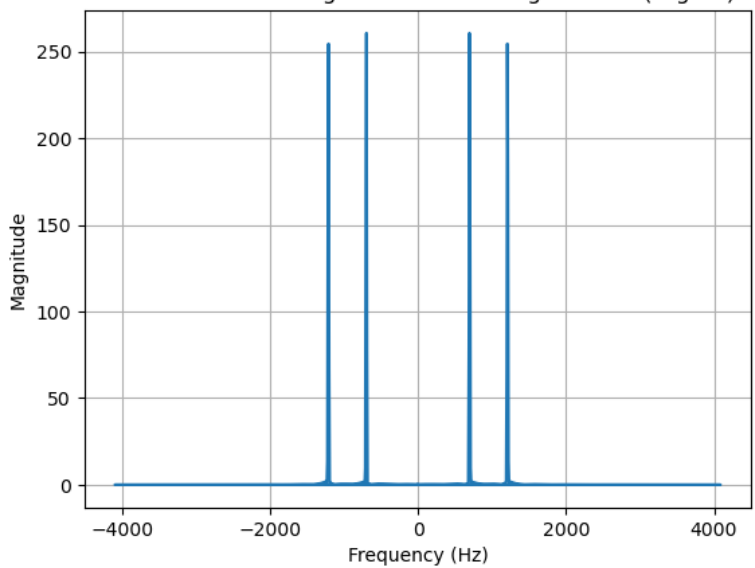
DFT of Windowed Signal with Rectangular Window (Digit 3)



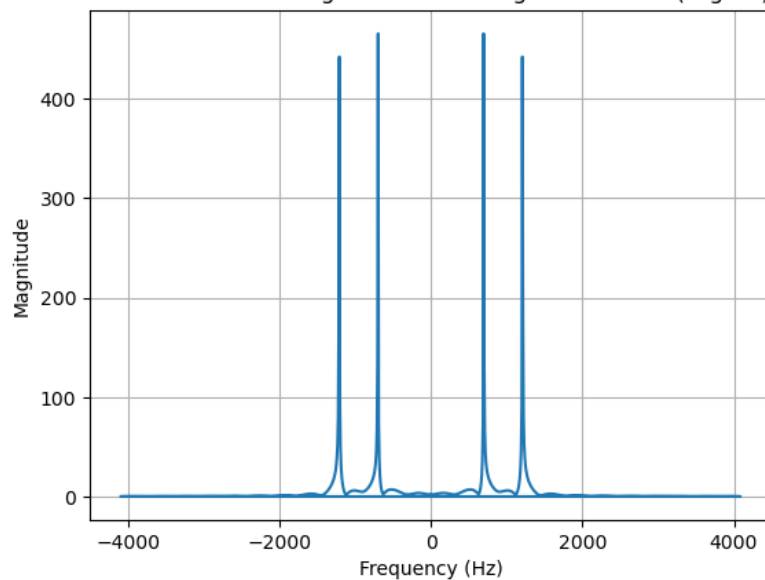
DFT of Windowed Signal with Hamming Window (Digit 3)

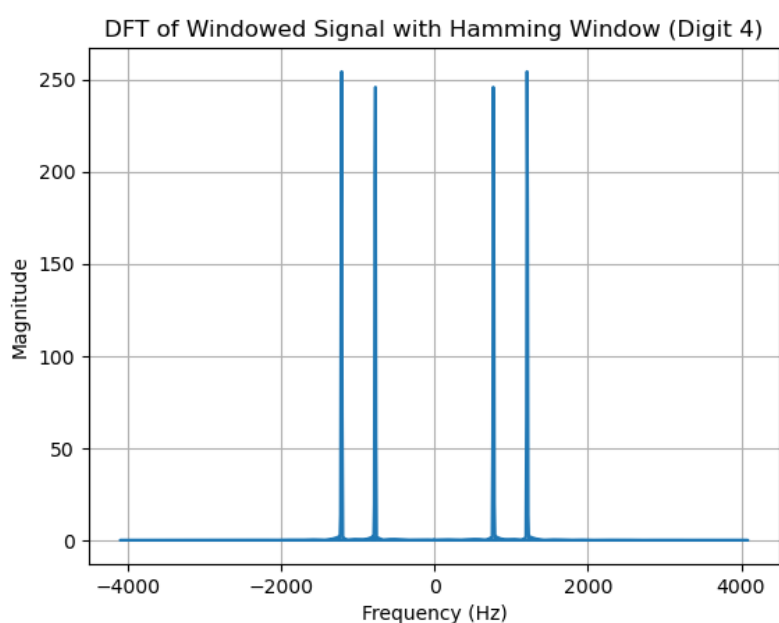
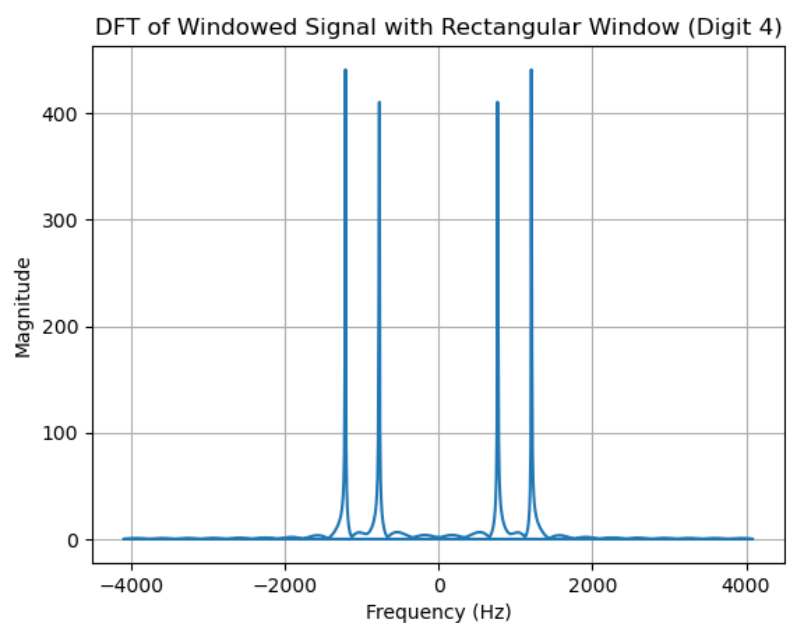
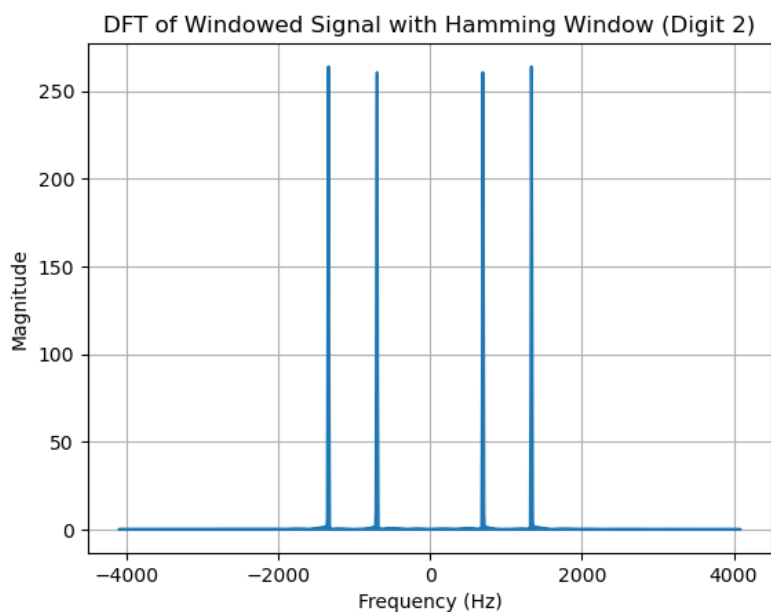
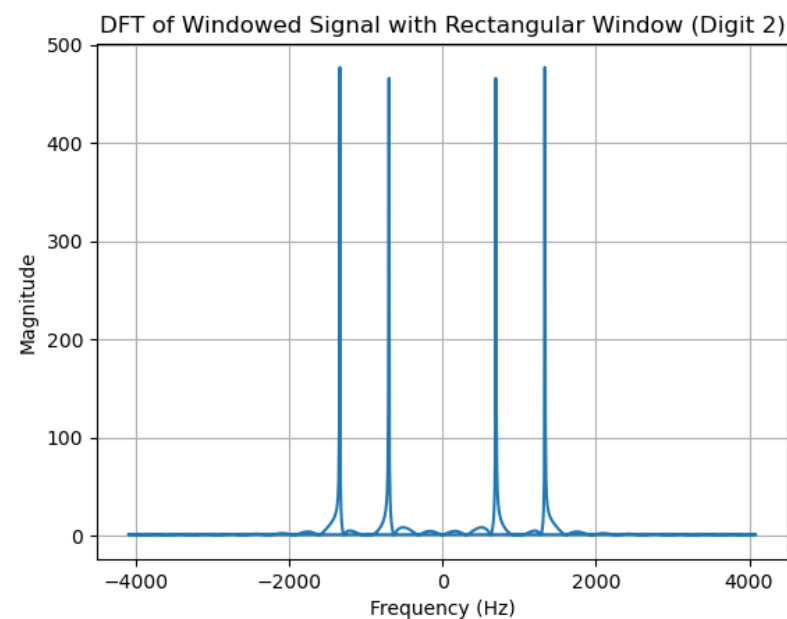


DFT of Windowed Signal with Hamming Window (Digit 1)



DFT of Windowed Signal with Rectangular Window (Digit 1)





Στα παραπάνω διαγράμματα παρατηρούμε ότι στα διαγράμματα με παράθυρο hamming έχουμε μικρότερη φασματική διαρροή, δηλαδή η ενέργεια του σήματος παραμένει πιο πολύ συγκεντρωμένη στον κύριο λοβό με λιγότερη διαρροή στις γειτονικές συχνότητες.

1.5. Για το ερώτημα αυτό υπολογίσαμε την ενέργεια από το fft των τόνων και κρατήσαμε τους δείκτες στους οποίους η ενέργεια ήταν > 10000 . Από αυτούς κρατήσαμε τους δύο πρώτους δείκτες κάθε φορά ώστε να πάρουμε τη λίστα των k. Στη συνέχεια φτιάξαμε μία λίστα με συχνότητες οι οποίες προκύπτουν από τη συνάρτηση : $2 * \pi * k_i / 1000$. Η λίστα λοιπόν που προέκυψε για τους τόνους 0-9 είναι :

List of k
[[115 163]
[85 148]
[85 163]
[85 180]
[94 148]
[94 163]
[94 180]
[104 148]
[104 164]
[104 180]]

Και η λίστα των αντίστοιχων συχνοτήτων που θεωρούμε ότι βρίσκονται εγγύτερα στις touch-tone συχνότητες είναι :

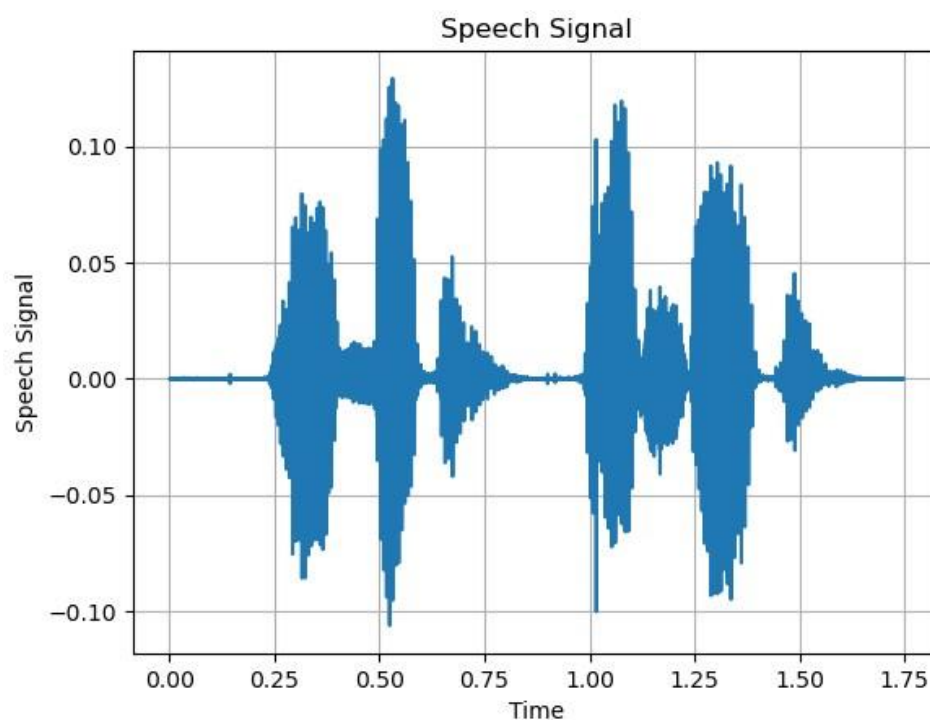
List of w
[[0.72256631 1.02415921]
[0.53407075 0.92991143]
[0.53407075 1.02415921]
[0.53407075 1.13097336]
[0.59061942 0.92991143]
[0.59061942 1.02415921]
[0.59061942 1.13097336]
[0.65345127 0.92991143]
[0.65345127 1.03044239]
[0.65345127 1.13097336]]

1.6. Σε αυτό το ερώτημα έχει γίνει μία προσπάθεια υλοποίησης, η οποία ωστόσο δεν λειτούργησε και για αυτό τον λόγο την παραλείπουμε. Για να μπορέσουμε να υλοποιήσουμε όμως το ερ 1.7 καλούμε την συνάρτηση της 1.6 με όνομα **ttdecode()** που θα **επέστρεφε τα ψηφία** του σήματος το οποίο βάζουμε ως όρισμα.

1.7. Ολοκληρώνοντας την πρώτη άσκηση χρειάστηκε να φορτώσουμε τα τρία σήματα 'easy_sig.npy', 'medium_sig.npy', 'hard_sig.npy' με την εντολή **np.load()** της numpy. Στη συνέχεια τα βάλαμε σαν ορίσματα της **ttdecode()** και τα ψηφία για κάθε σήμα που αποκωδικοποίησε η συνάρτηση του ερωτήματος 1.6 τα τυπώνουμε σε μορφή λίστας..

Μέρος 2ο - Χαρακτηριστικά Βραχέος Χρόνου Σημάτων Φωνής και Μουσικής

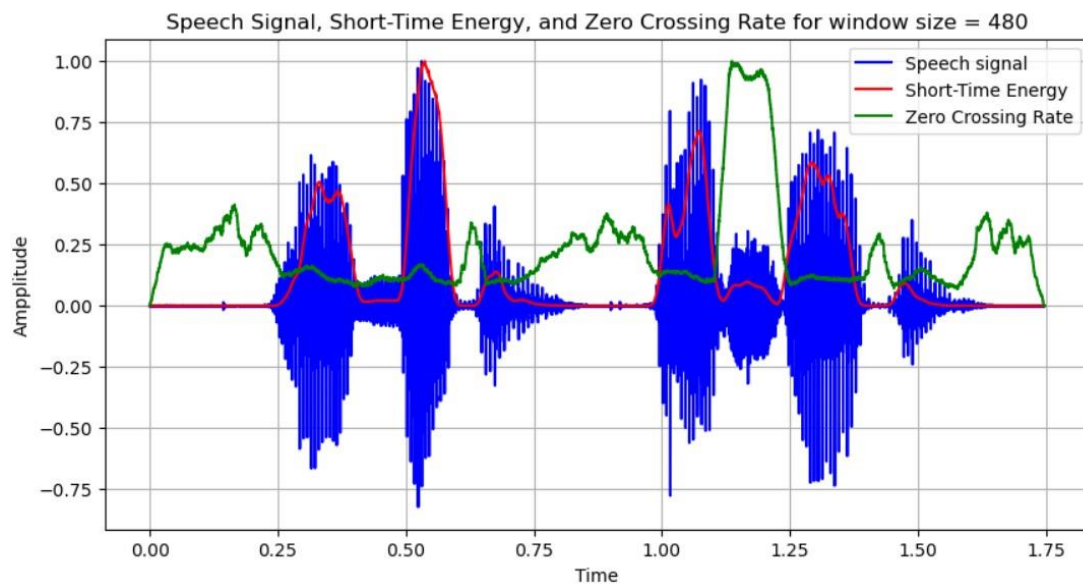
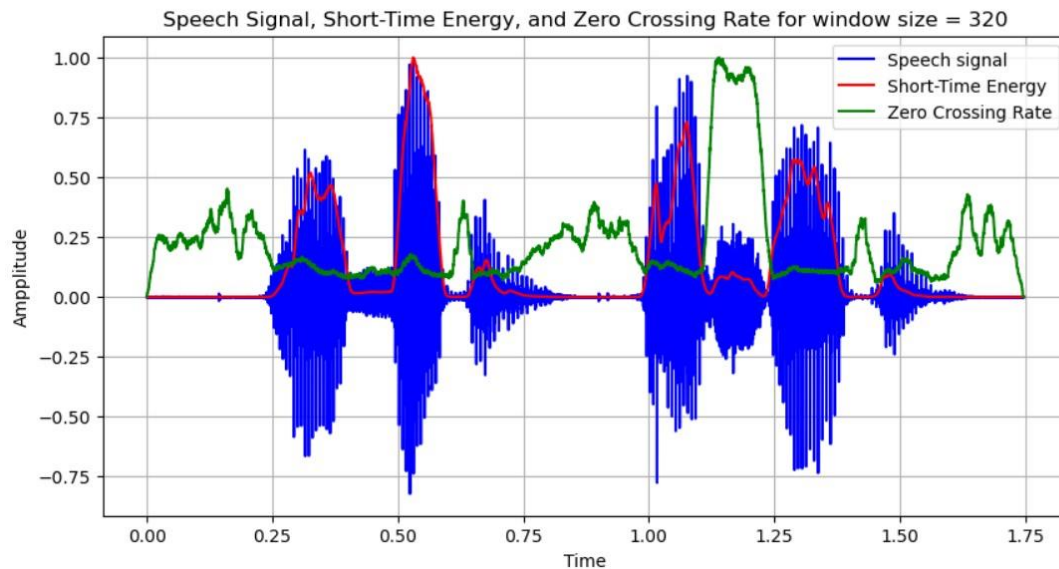
2.1. Διαβάζουμε το σήμα ομιλίας από το δοσμένο αρχείο “speech_utterance.wav” με την εντολή **librosa.load()** της βιβλιοθήκης **librosa** και παίρνουμε τη συχνότητα δειγματοληψίας **Fs=16kHz**. Έπειτα πλοτάρουμε το δοσμένο σήμα στο πεδίο του χρόνου χρησιμοποιώντας **np.linspace()** με άνω όριο το μήκος του σήματος διά τη συχνότητα δειγματοληψίας και μέγεθος ίσο με το μήκος του σήματος. (Στο τέλος αναπαράγουμε το σήμα φωνής με την εντολή **display(Audio())**). Η απεικόνιση του σήματος στο πεδίο του χρόνου (διάρκειας 1,75) είναι:

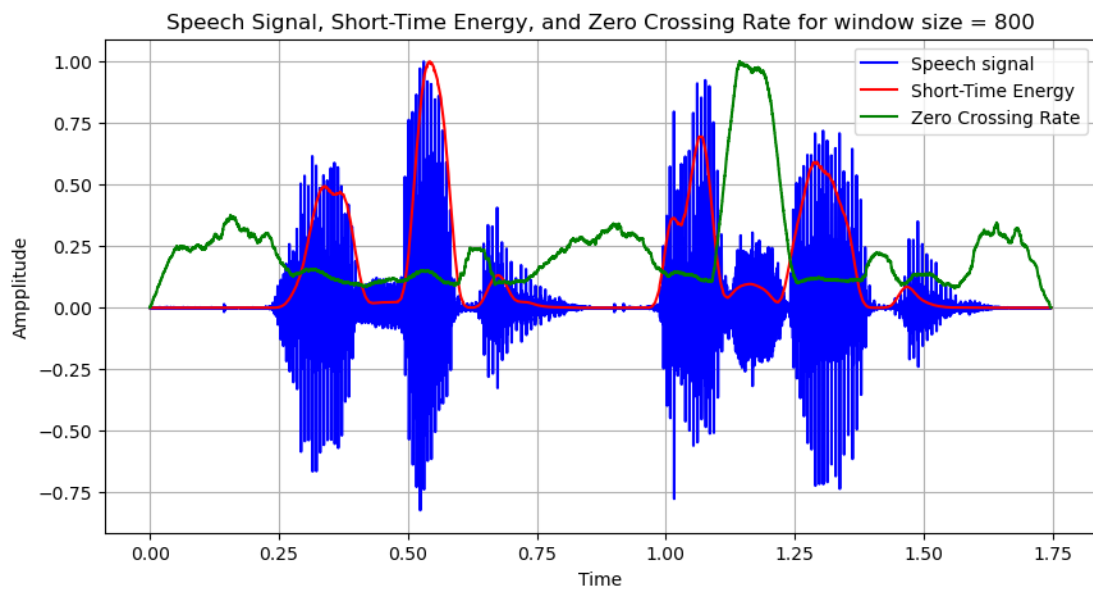
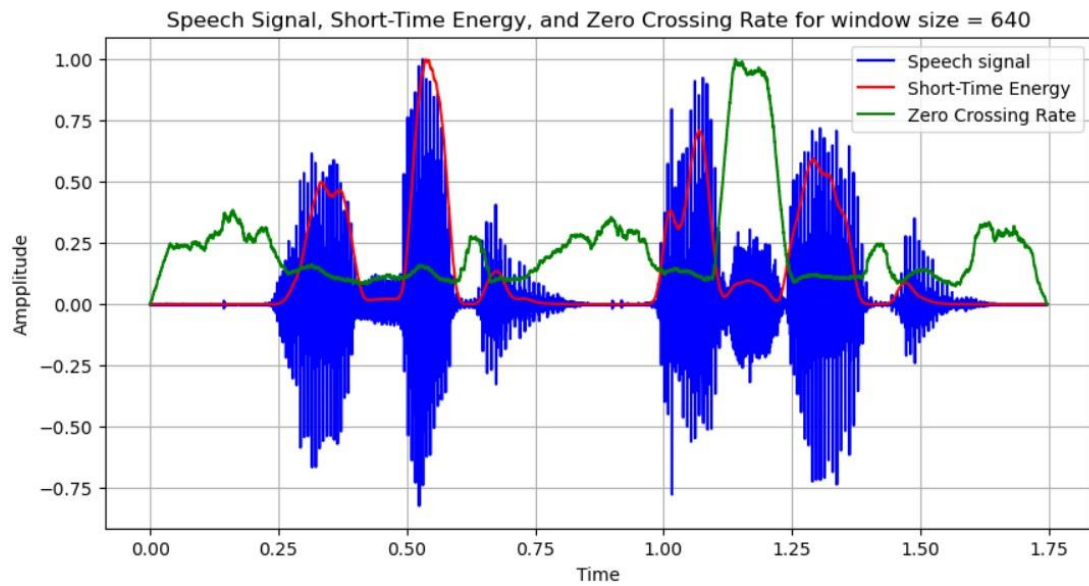
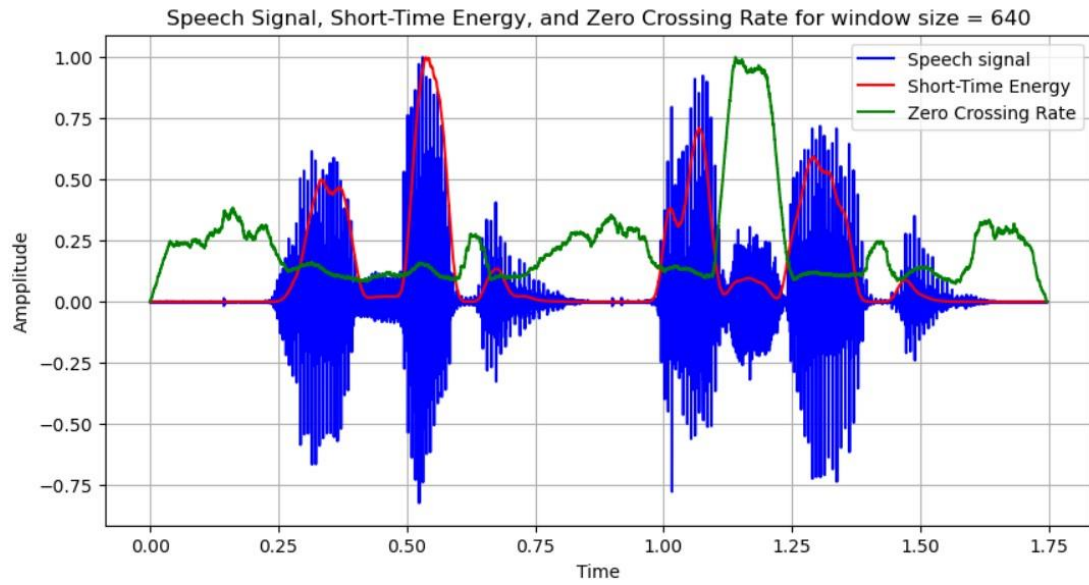


2.2. Υπολογίζουμε την ενέργεια βραχέως χρόνου και το ρυθμό εναλλαγής προσήμου από τους δοσμένους τύπους (1),(2), για τα οποία χρησιμοποιούμε παράθυρο μεταβαλλόμενου **μεγέθους N** από **320 (20ms)** έως **800 (50ms)** με **βήμα 160 (0.01ms)**. Χρησιμοποιούμε Hamming για την ενέργεια με την συνάρτηση **hamming()** και τετραγωνικό με την εντολή **ones()** για το ρυθμό εναλλαγής προσήμου. Για την κατασκευή των μεγεθών χρησιμοποιήσαμε συνέλιξη με τη συνάρτηση **convolve()** της βιβλιοθήκης **scipy.signal.library**. Στον υπολογισμό των συνελίξεων χρειάστηκε να υπολογίσουμε τα μετατοπισμένα σήματα, το οποίο κάναμε με τη χρήση της **roll()** της **numpy**. Τα διαγράμματα τα **κανονικοποιήσαμε** στο 0-1 ώστε να είναι πιο εύκολη η σύγκριση τους.

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**

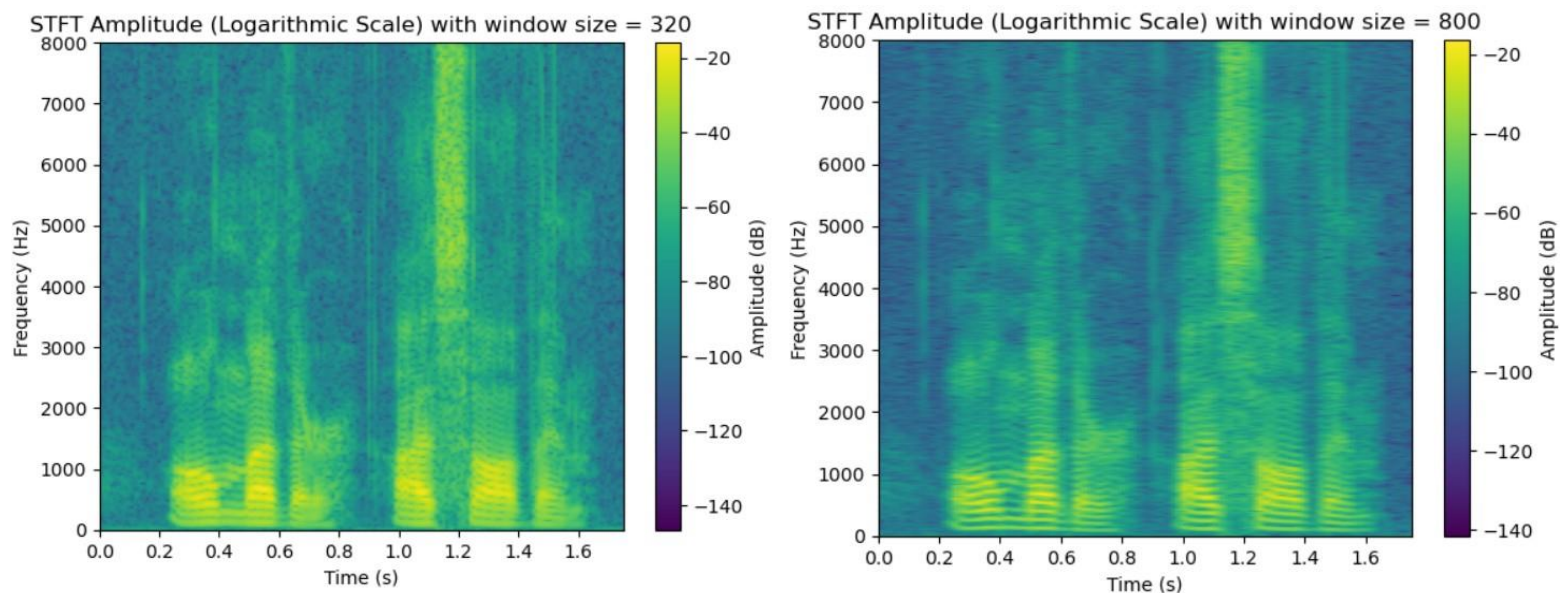
θόρυβος οπότε η απεικόνιση της ενέργειας και του ρυθμού εναλλαγής προσήμου είναι πιο εύχρηστη. Από τα διαγράμματα μπορούμε να διαχωρίσουμε φωνή από σιωπή, όταν η **ενέργεια είναι υψηλότερη** και όταν είναι **κοντά ή ίση με το μηδέν** αντίστοιχα. Οι **έμφωνοι ήχοι** παρατηρούνται στις περιοχές όπου έχουμε **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**. Οι μετρήσεις παρουσιάζονται σε κοινά διαγράμματα παρακάτω:





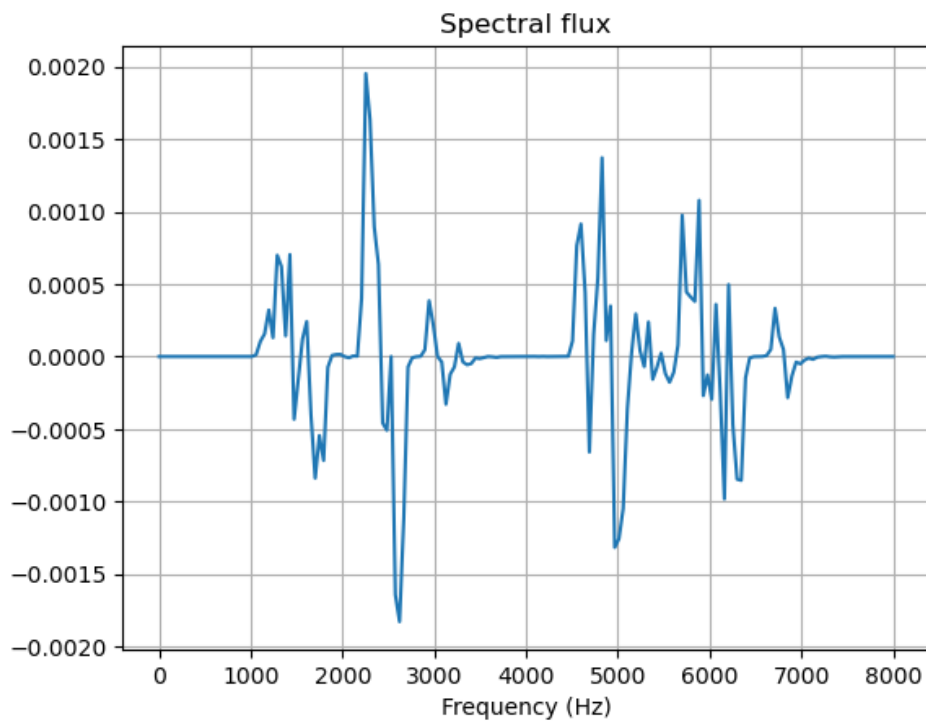
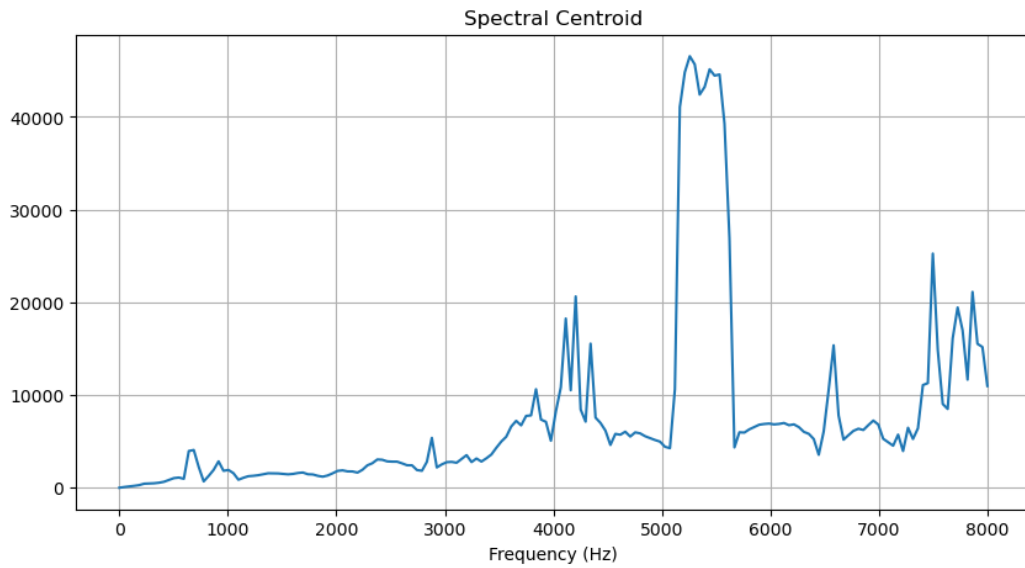
2.3. Υπολογίσαμε το μετασχηματισμό Fourier βραχέως χρόνου (STFT) με τη βοήθεια της συνάρτησης `stft()` της `scipy.signal` και σχεδιάσαμε το πλάτος του (`abs()`) σε λογαριθμική κλίμακα (`log10()`) από τη `numpy`. Κατασκευάσαμε φασματογράφημα με την συνάρτηση `pcolormesh()` και βάλαμε χρωματική κλίμακα με τη συνάρτηση `colorbar()` της `matplotlib.pyplot`.

Πειραματιζόμενοι με το **εύρος** του παραθύρου για τιμές **20-50ms** 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. Αντίθετα, η συχνοτική ανάλυση γίνεται υψηλότερη όσο μεγαλώνει το παράθυρο καθώς το φασματογράφημα έχει πιο στενές συχνοτικές διαστάσεις, γεγονός που επιτρέπει **καλύτερο διαχωρισμό και αναγνώριση των συχνοτικών στοιχείων**. Αυτά φαίνονται στα παρακάτω παραδείγματα της min και max ενδεικτικής τιμής παραθύρου:

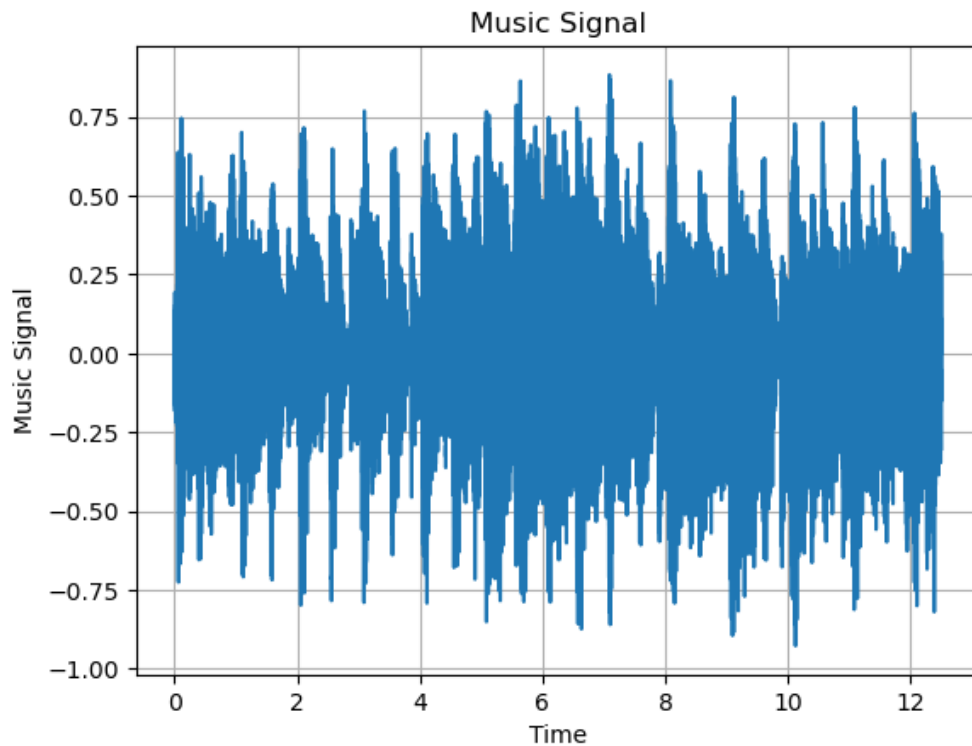


Στη συνέχεια χρησιμοποιούμε το STFT για να πάρουμε τις τιμές του φασματικού κέντρου (Spectral Centroid) και της φασματικής ροής, σύμφωνα με τους δοσμένους τύπους, τα οποία φαίνονται παρακάτω, για 20ms παράθυρο.

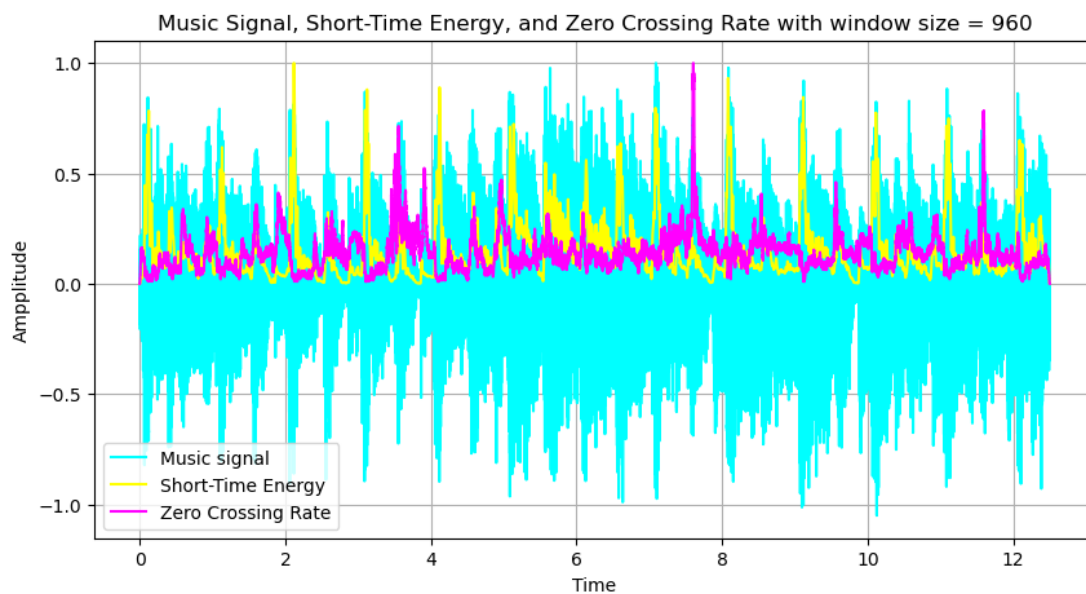
Το **φασματικό κέντρο** που βρίσκεται το **κέντρο μάζας** του φάσματος. Δείχνει το μέτρο της φασματικής θέσης και στα σημεία που παρατηρούμε **υψηλές συχνότητες** αντιστοιχούν **έντονοι ήχοι** του σήματος μας. Η **φασματική ροή** παρατηρούμε, όπως είναι αναμενόμενο, ότι παρουσιάζει **υψηλές τιμές για ξαφνικές αλλαγές** του σήματος μας. Γι' αυτό μεταξύ των φράσεων παρουσιάζεται κενό ενώ όταν εκφωνούνται τα γράμματα έχουμε έντονες διακυμάνσεις.

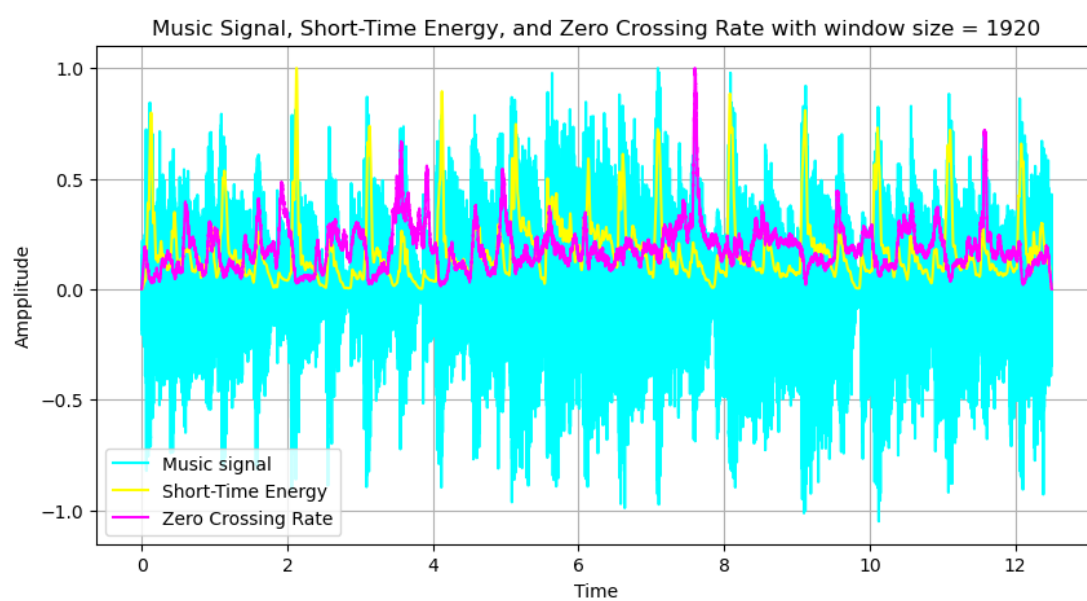
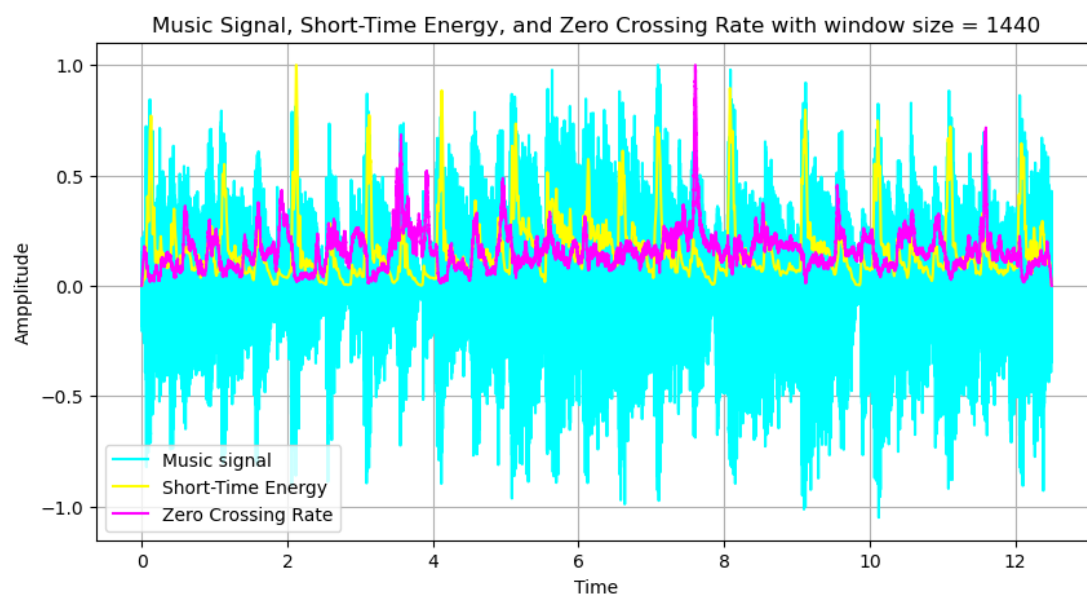


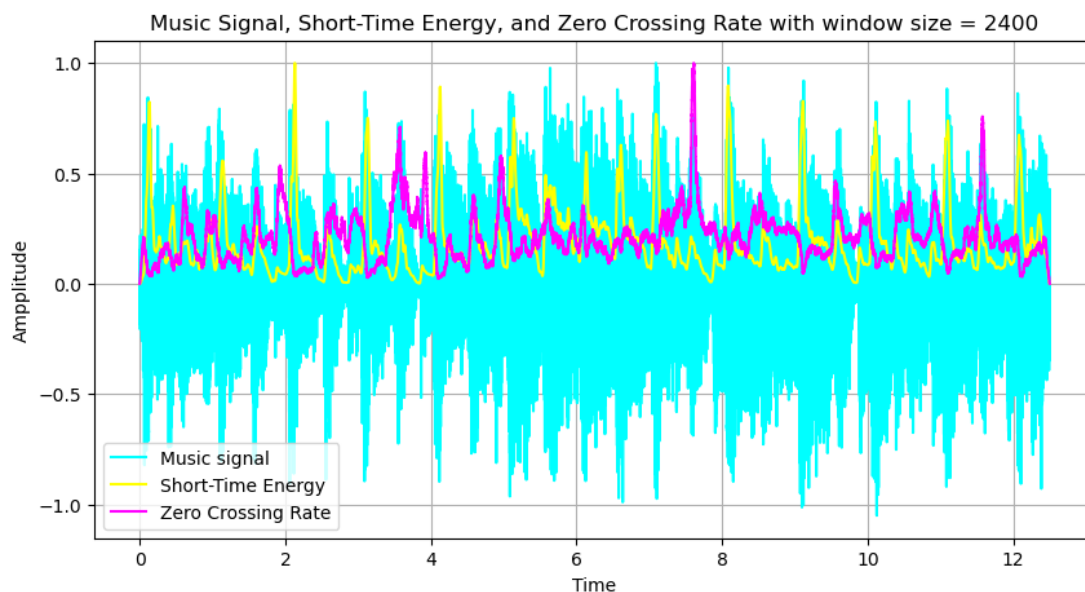
2.4. Επαναλάβαμε τη διαδικασία για το δοσμένο σήμα μουσικής “**music.wav**”, το οποίο αρχικά διαβάζουμε και εξάγουμε συχνότητα δειγματοληψίας **Fs_m = 48kHz**, το πλοτάρουμε (και το αναπαράγουμε). Η απεικόνιση του σήματος μουσικής στο πεδίο του χρόνου είναι:



Στη συνέχεια υπολογίσαμε την ενέργεια βραχέως χρόνου όμοια με το ερ. 2.2, και παρατηρήσαμε ότι όσο μεγαλώνουμε το εύρος του παραθύρου τόσο πιο λείες γίνονται οι γραφικές παραστάσεις. Παρακάτω απεικονίζονται σε κοινό διάγραμμα το μουσικό σήμα η ενέργεια βραχέως χρόνου και ο ρυθμός εναλλαγής προσήμου για τα διάφορα μεγέθη του παραθύρου, σε ίδια λογική με το 2.2.

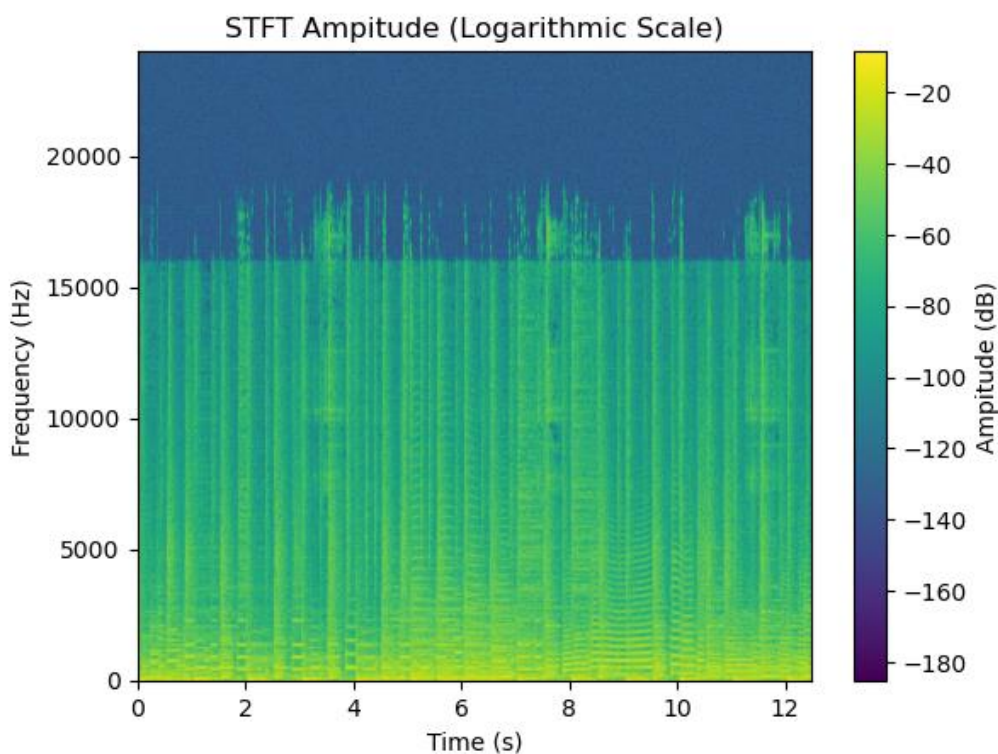




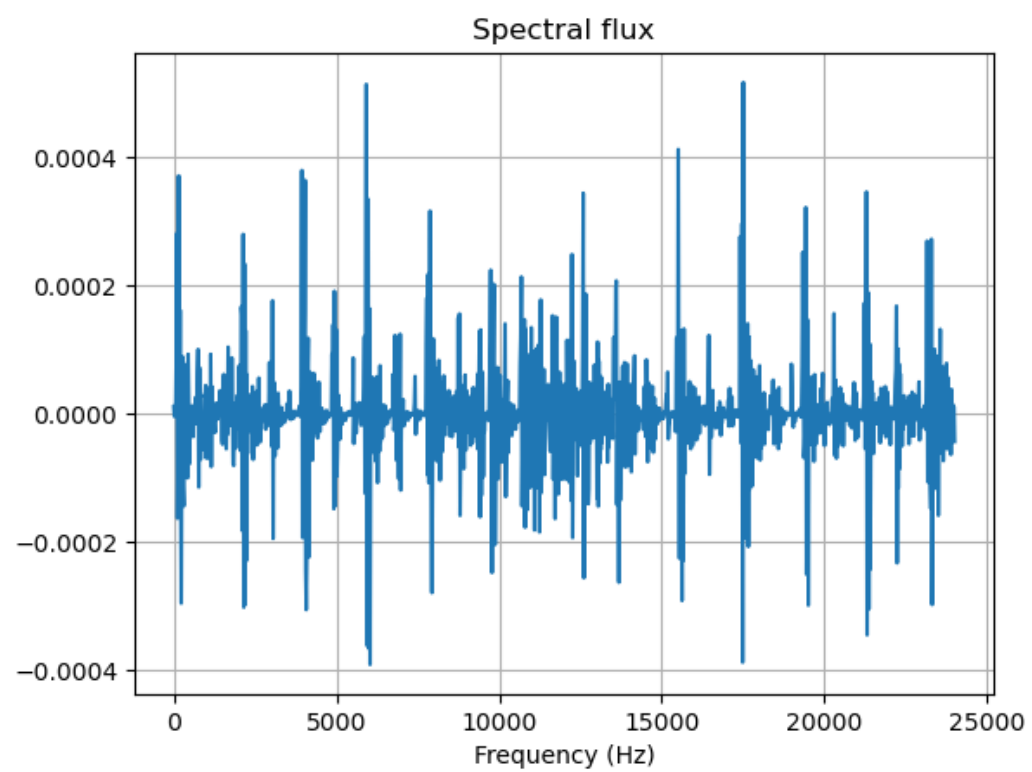
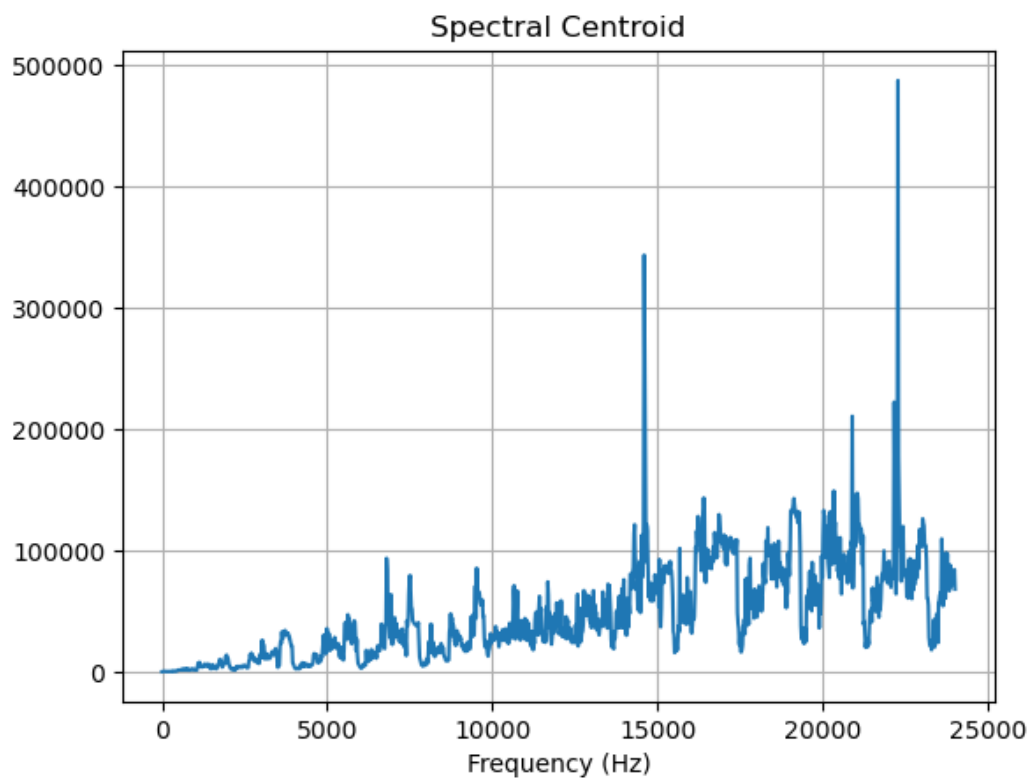


Όμοια με προηγούμενως παρατηρούμε ότι όσο μεγαλώνει το μήκος του παραθύρου η γραφική παράσταση εξομαλύνεται (τα σχόλια σε όλα τα ερωτήματα της 2.4 είναι όμοια με τα προηγούμενα ερωτήματα).

Έπειτα, υπολογίσαμε τον μετ/μο Fourier βραχέως χρόνου και σχεδιάσαμε το πλάτος του σε λογαριθμική κλίμακα πλοτάροντας το παρακάτω φασματογράφημα:



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

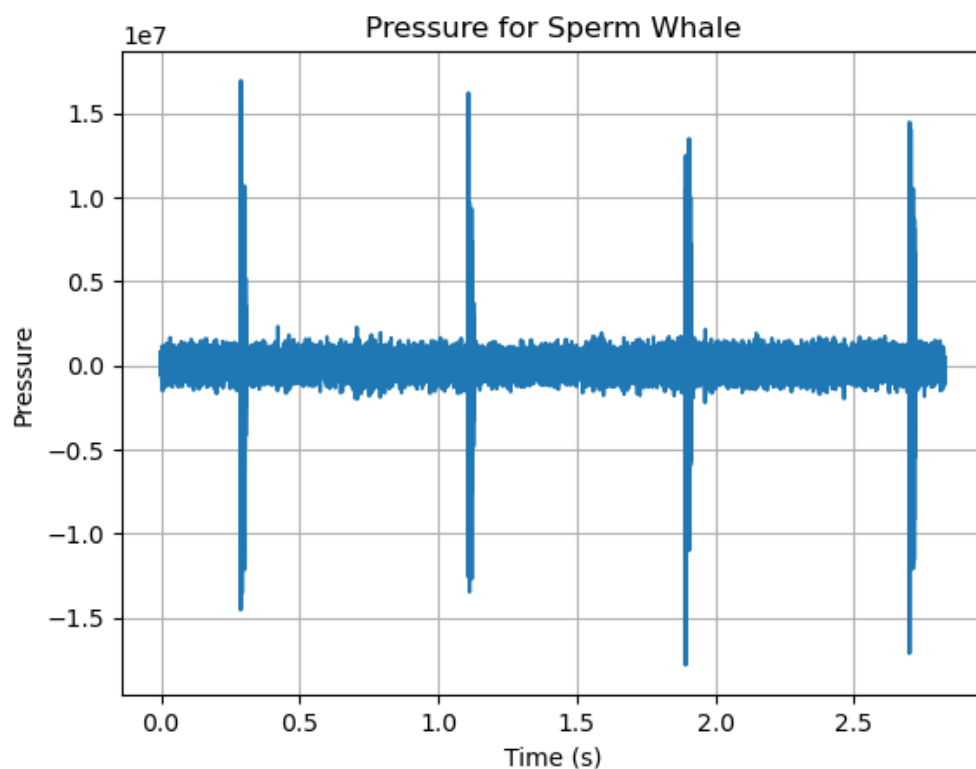


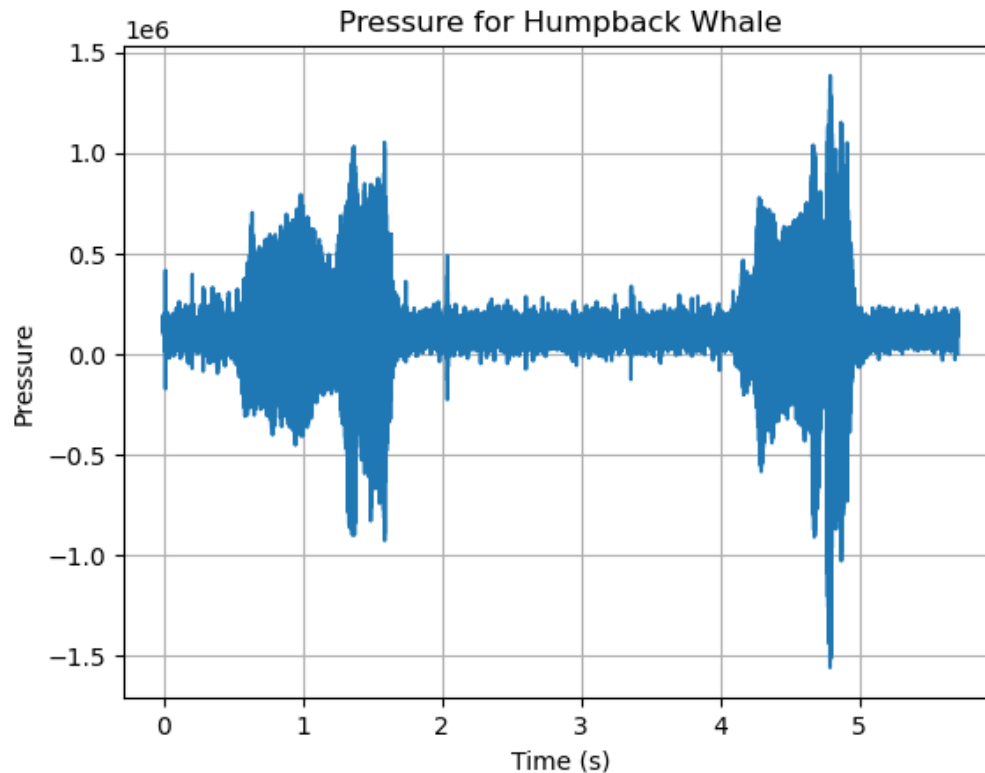
Μέρος 3ο - Ψηφιακή Επεξεργασία και Φασματική Ανάλυση

Βιοσημάτων

Κητωδών

3.1. Φορτώσαμε το σήμα ήχου για κάθε φάλαινα με την συνάρτηση `librosa.load()` και πήραμε ρυθμό δειγματοληψίας **$F_s_{sp} = 22050 \text{ Hz}$** . Έπειτα μετατρέψαμε τα σήματα τάσης σε σήματα πίεσης μέσω του τύπου (5) (με δεδομένο το **$GF = 0.16$** και τη **sensitivity = -155 dB relative to 1V/ μPa** και πλοτάρουμε τη πίεση στο χρόνο για κάθε φάλαινα όπως φαίνεται παρακάτω. Παρατηρούμε στα διαγράμματα ότι, πράγματι, οι φάλαινες φυσητήρες εκπέμπουν παλμικούς ασυνεχείς ήχους, το οποίο όμοια φαίνεται στη πίεση, και οι megάπτερες φάλαινες εκπέμπουν συνεχή σήματα.





3.2. Σε αυτό το ερώτημα υπολογίσαμε από τους δοσμένους τύπους τη μέση τετραγωνική ρίζα της πίεσης (**prms**) για κάθε φάλαινα και μετά υπολογίσαμε το επίπεδο ηχητικής πίεσης (**SPL**) από το δοσμένο τύπο. Τα αποτελέσματα που πήραμε ήταν τα εξής:

Για τη φάλαινα φυσητήρα: $prms = 749840.3376660865$ και $SPL_{rms} = 117.49937599177156$

Για τη megάπτερη φάλαινα: $prms = 749840.3376660865$ και $SPL_{rms} = 105.83511686211784$

3.3. Σε αυτό το σημείο κληθήκαμε να επιβεβαιώσουμε για ένα από τα βιοσήματα (**επιλέξαμε της Sperm whale**) το θεώρημα Parseval. Έτσι πήραμε και υπολογίσαμε **κάθε μέλος** της εξίσωσης **ξεχωριστά** και καταλήξαμε **στο ίδιο αποτέλεσμα**. Τα νούμερα αποκλίνουν ελάχιστα, πράγμα λογικό λόγω πιθανών προσεγγίσεων του θεωρήματος του Parseval και τυχαίων σφαλμάτων στις μετρήσεις. Οπότε εκτυπώσαμε τα αποτελέσματα:

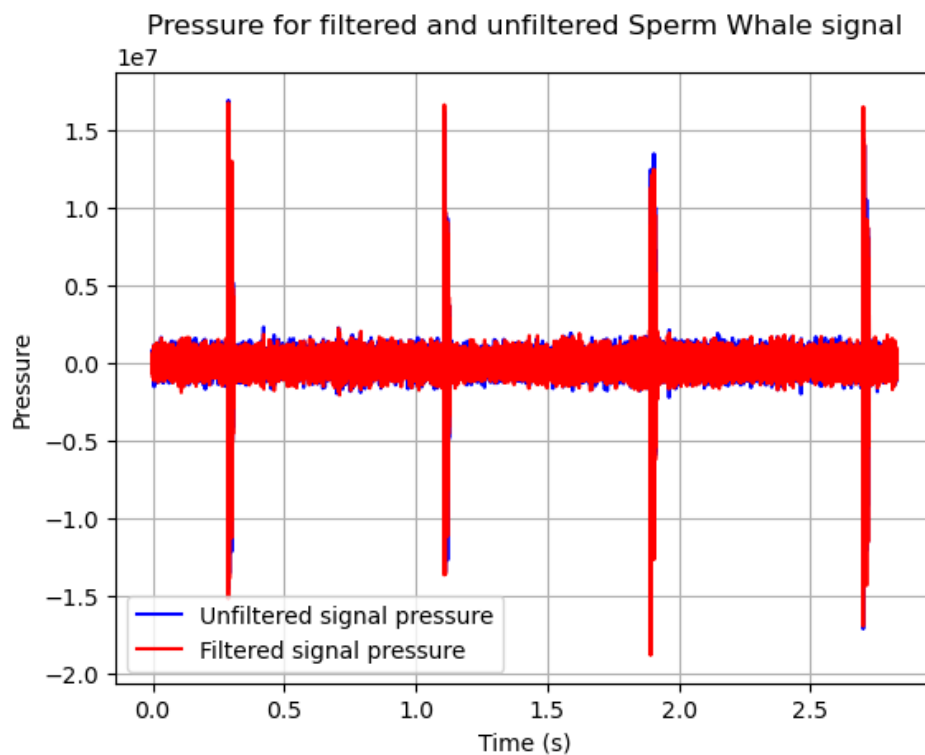
left is 35.0434477693993

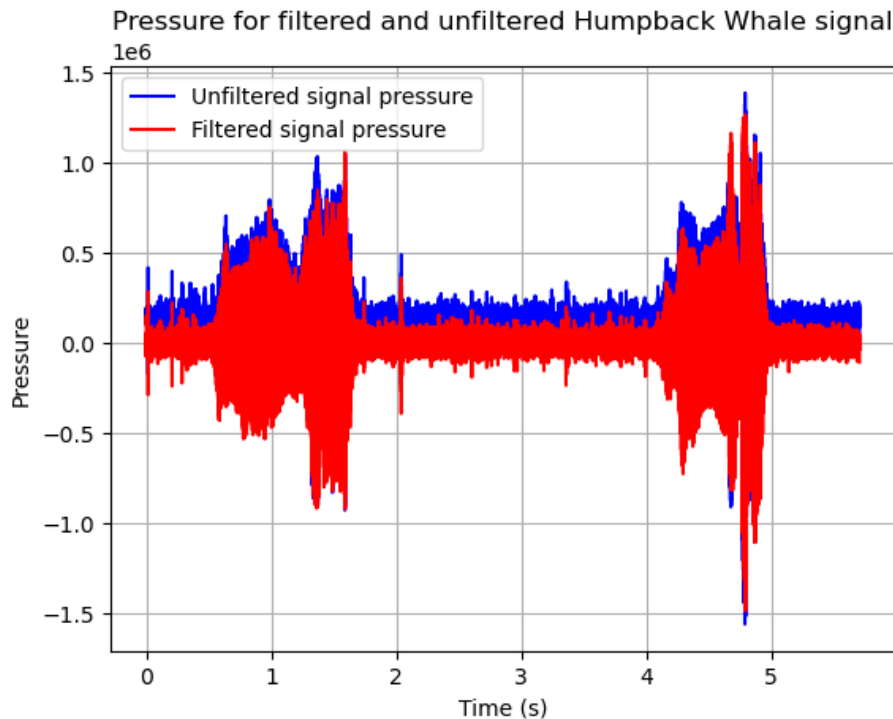
right is 35.0434477631478

Parseval theorem is confirmed

3.4. Για να απομονώσουμε το σήμα από χαμηλές ανθρωπογενείς συχνότητες εφαρμόσαμε στα σήματα **ζωνοπερατό(ουσιαστικά υψιπερατό) Butterworth φίλτρο** τάξης **N=3**, με συχνότητα αποκοπής 200 Hz. Αξιοποιήσαμε τις συναρτήσεις **signal.butter()** και **signal.lfilter()** της βιβλιοθήκης **scipy**. Τυπώσαμε τα γραφήματα πίεσης χρόνου του φιλτραρισμένου και του αρχικού σήματος πίεσης σε κοινά διαγράμματα για κάθε φάλαινα, τα οποία φαίνονται παρακάτω.

Στα γραφήματα αυτά παρατηρούμε ότι τα σήματα των γραφικών παραστάσεων μετακινήθηκαν ελαφρώς πιο κάτω, δηλαδή **αφαιρέθηκε πλάτος από τη πίεση**. Αυτό ήταν αναμενόμενο καθώς με την **αποκοπή των συχνοτήτων κάτω από τα 200 Hz αφαιρέσαμε θόρυβο** και λάβαμε τώρα ένα πιο «καθαρό» αποτέλεσμα εστιασμένο στα ζητούμενα σήματα, που είναι τα σήματα των κητωδών.



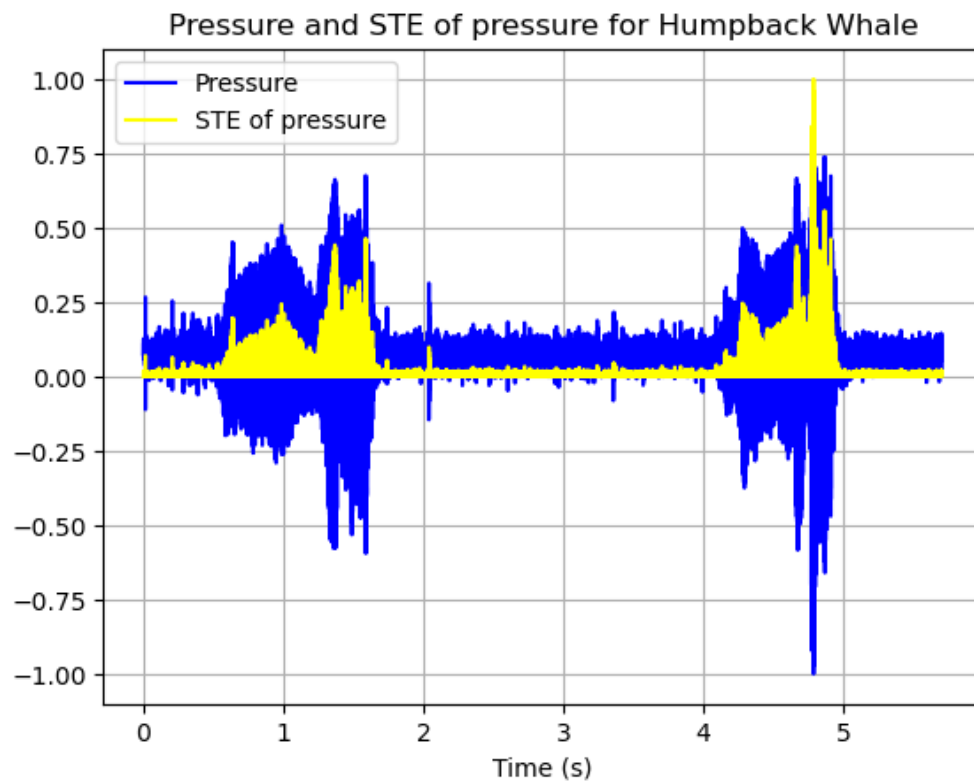
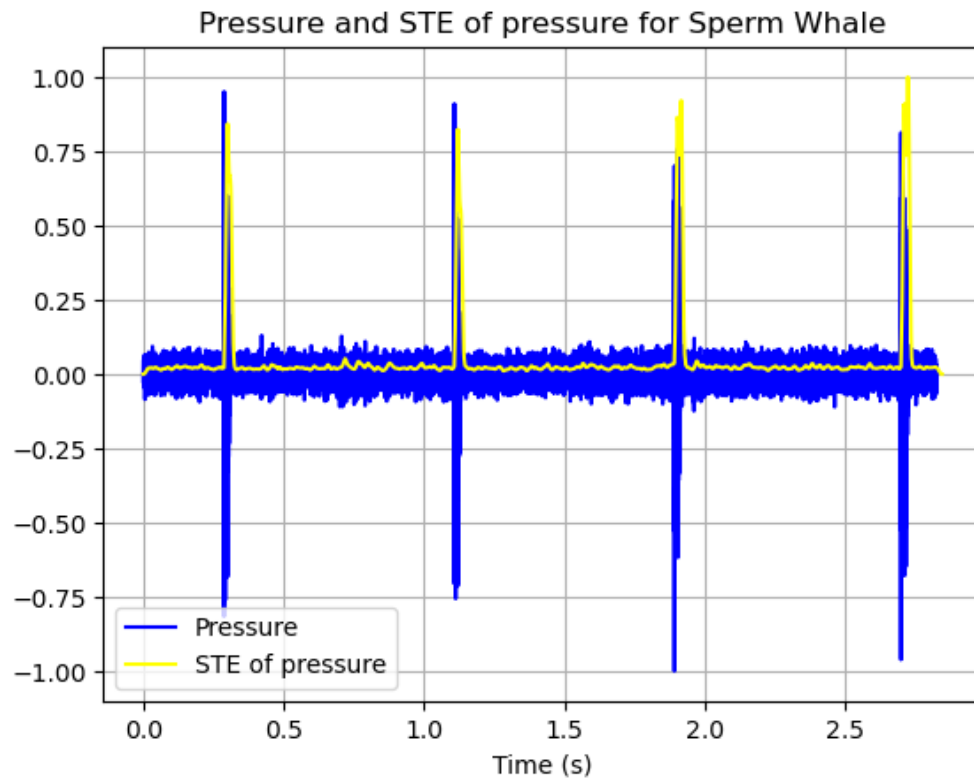


3.5. Φορτώνουμε το αρχείο με τον ανθρωπογενή ήχο '**Pile driving.wav**' το οποίο βρίσκουμε ότι έχει συχνότητα δειγματοληψίας $F_s_hum = 22050$ Hz. Αφού υπολογίσαμε τη πίεση του (**pressure_hum**), τη μέση τετραγωνική ρίζα της πίεσης (**p_rms_hum**) και το επίπεδο ηχητικής πίεσης (**SPL_rms_hum**), παρατηρήσαμε ότι η φυσήτρες που ακούνε σε όλο το φάσμα συχνοτήτων επηρεάζονται από το ανθρωπογενή ήχο καθώς τα επίπεδα ηχητικής πίεσης υπερβαίνουν τα 100 dB relative to 1μPa, σύμφωνα με τη παραδοχή. Συγκεκριμένα πήραμε την εξής μέτρηση: **SPLrms_hum is 165.15296950722166**

Για την μεγάπτερη φάλαινα, εφαρμόσαμε **ζωνοπερατό φίλτρο** στο σήμα ανθρωπογενούς ήχου ώστε να πάρουμε μόνο τις συχνότητες **300-3000 Hz** στις οποίες ακούνε αυτές οι φάλαινες. Από το αποτέλεσμα του **SPL_rms** για το φιλτραρισμένο σήμα συμπεράναμε ότι και αυτό το είδος φάλαινας επηρεάζεται από τους ανθρωπογενείς υποθαλάσσιους ήχους. Συγκεκριμένα πήραμε την εξής μέτρηση: **SPLrms_fitlered_hum is 141.51802441407614**

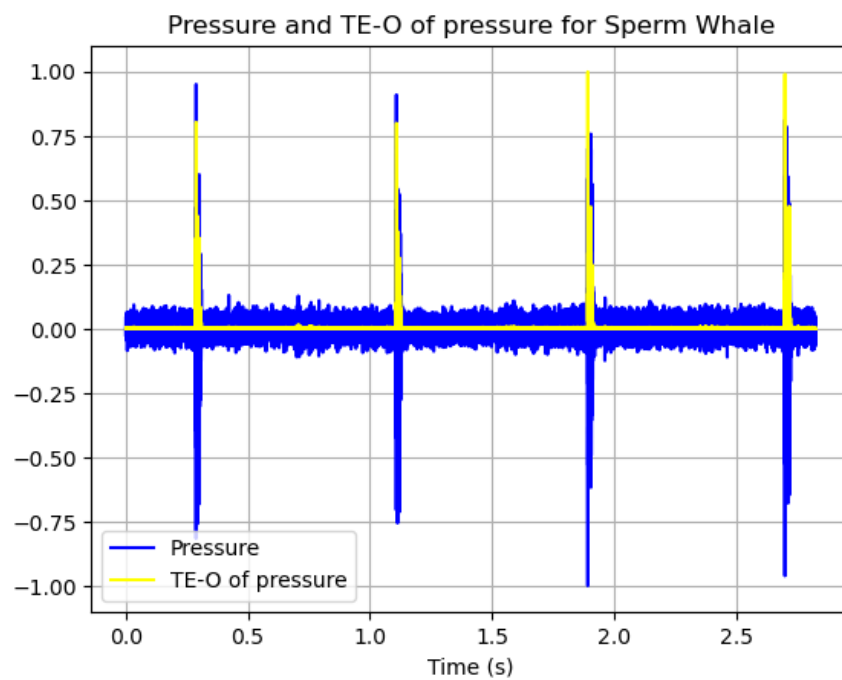
3.6. Αξιοποιώντας την ομοιότητα με συνέλιξη (συνάρτηση **convolve()**) του τύπου της ενέργειας βραχέως χρόνου υπολογίσαμε το ζητούμενο μέγεθος για το αρχικό σήμα πίεσης-χρόνου, για κάθε φάλαινα. Τα αποτελέσματα τα σχεδιάσαμε σε κοινά διαγράμματα με το αρχικό σήμα πίεσης-χρόνου αντίστοιχα για κάθε θηλαστικό, τα οποία φαίνονται παρακάτω. Για την ευκολότερη ποιοτική σύγκριση των δύο διαγραμμάτων σε κάθε περίπτωση εφαρμόσαμε κανονικοποίηση των μεγεθών στο 0-1. Στα διαγράμματα παρατηρούμε ότι η ενέργεια κυμαίνεται με τον ίδιο τρόπο (αν και με μικρότερο πλάτος) με τη πίεση. Δηλαδή έχουν όμοια μορφή οι γραφικές τους παραστάσεις. Πράγμα λογικό καθώς το διάγραμμα πίεσης χρόνου μας αποτελεί

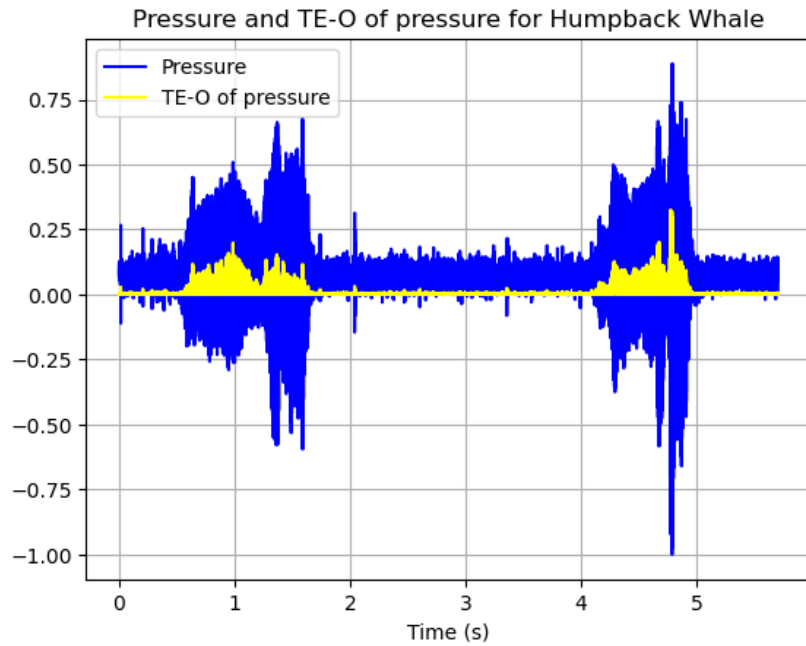
οπτική αναπαράσταση της φυσικής διακύμανσης του ήχου και η ενέργεια βραχέως χρόνου δείχνει την ενέργεια του ήχου σε διαφορετικά χρονικά παράθυρα.



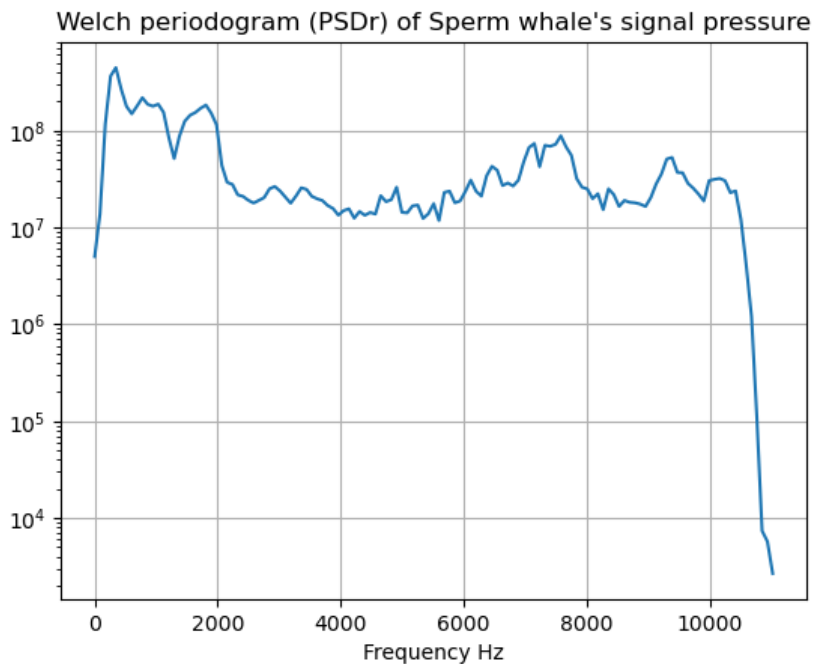
3.7. Εφαρμόσαμε τον ενεργειακό τελεστή **Teager-Kaiser Operatator** στο διακριτού χρόνου σήμα πίεσης-χρόνου σύμφωνα με τη δοσμένη εξίσωση (10) για κάθε φάλαινα, όπου για την ολίσθηση χρησιμοποιήσαμε την **roll()** και πήραμε τα ακόλουθα κοινά διαγράμματα πίεσης χρόνου και TE-O – χρόνου για κάθε ένα από τα δύο βιοσήματα αντίστοιχα.

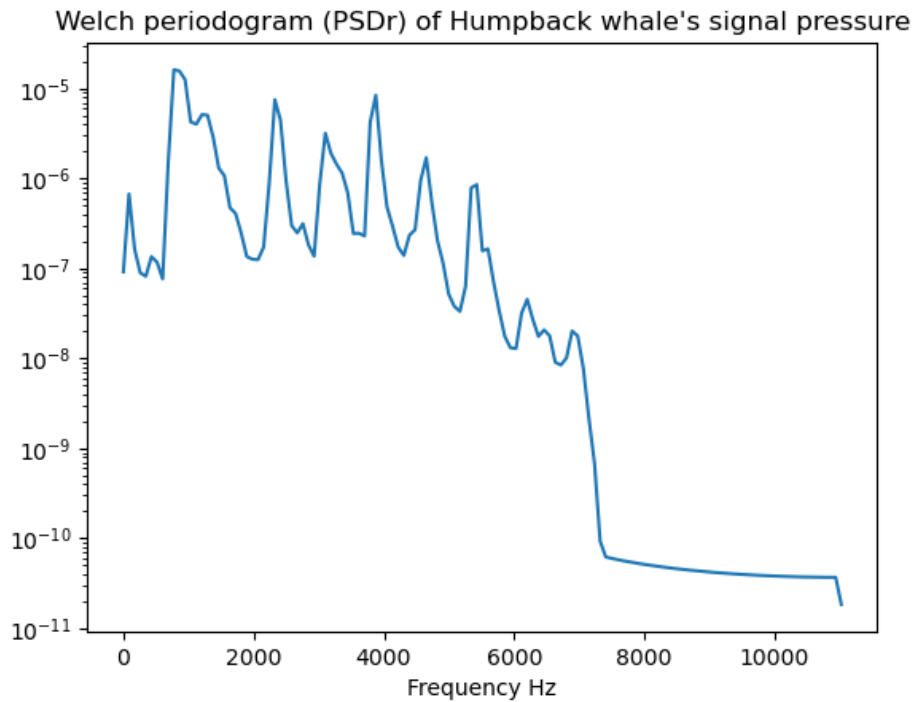
Από τα παρακάτω διαγράμματα παρατηρούμε ότι η ενέργεια από τον ενεργειακό τελεστή έχει μικρότερο πλάτος σαν γενική εικόνα από την ενέργεια βραχέως χρόνου. Πιο συγκεκριμένα, παρατηρούμε και διαφορά στη μορφή καθώς σε κάποια σημεία δεν ακολουθεί πιστά την γραφική παράσταση της πίεσης όσον αφορά τις αυξομειώσεις.





3.8. Υλοποιώντας έναν εκτιμητή φασματικής πυκνότητας εφαρμόσαμε μέσω της συνάρτησης `scipy.signal.welch()` το **περιοδόγραμμα Welch** στο σήμα της πίεσης για κάθε φάλαινα. Τα αποτελέσματα φαίνονται στα παρακάτω γραφήματα. Σε αυτά παρατηρούμε ότι για τη φάλαινα φυσητήρα η κατανομή της ενέργειας είναι σχετικά ομοιόμορφη από 0 έως 10000 Hz. Στη περίπτωση της μεγάπτερης φάλαινας παρατηρούμε πιο οξείες αυξομειώσεις στην ενέργεια η οποία μειώνεται κατακόρυφα κοντά στα 7500Hz μέχρι να μηδενίσει μετά τα 10000Hz.





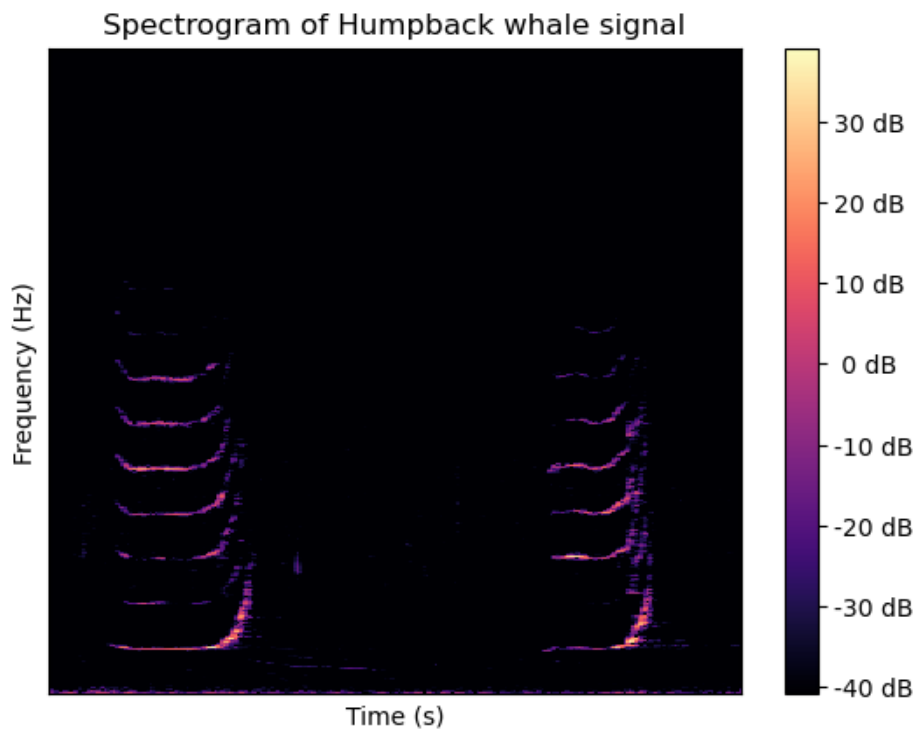
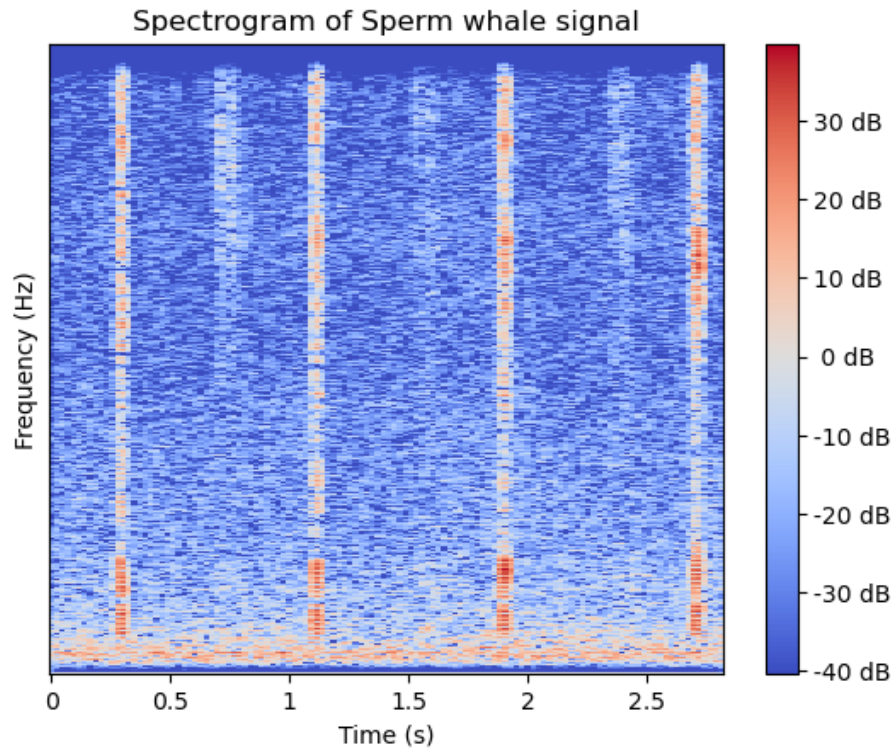
3.9. Σε αυτό το ερώτημα για κάθε φάλαινα υπολογίσαμε το μετασχηματισμό STFT των σημάτων τους με το **librosa.stft()**, αλλάξαμε σε κλίμακα dB το τετράγωνο του μετ/μου με τη συνάρτηση **librosa.amplitude_to_db()** και τυπώσαμε τα φασματογραφήματα των βιοσημάτων χρησιμοποιώντας την **librosa.display.specshow()**, τα οποία φαίνονται παρακάτω.

Στα φασματογραφήματα παρατηρούμε ότι η ενέργεια του σήματος τη Sperm whale είναι μεγαλύτερη καθώς είναι πιο έντονα τα χρώματα το φασματογράφημα της.

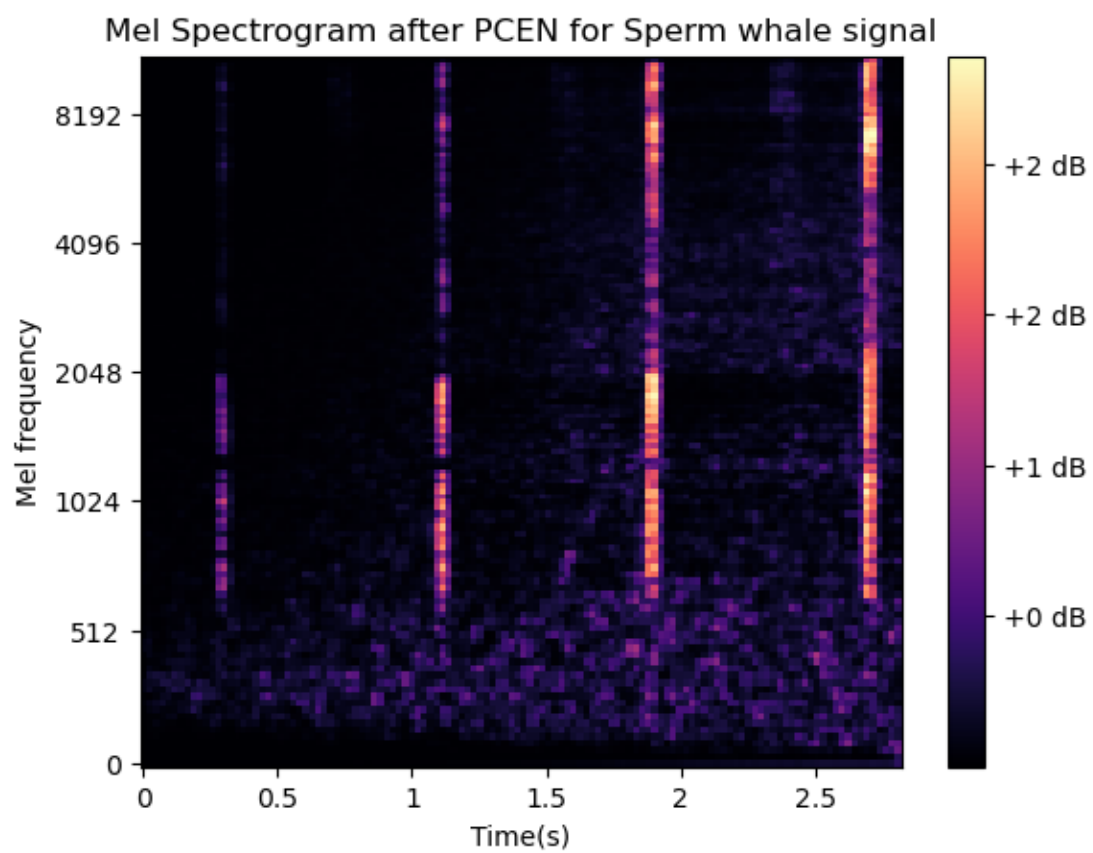
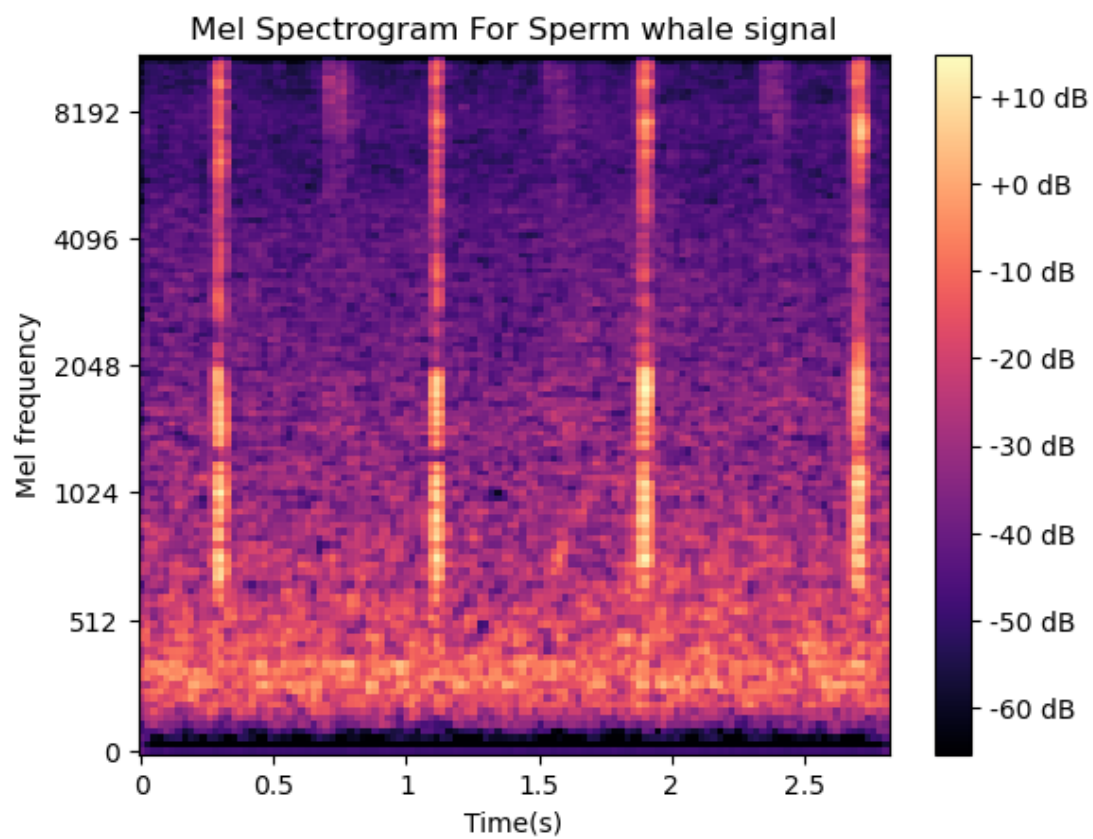
Επίσης, αυτή την ενέργεια την εμφανίζει στα ίδια χρονικά διαστήματα που εμφανιζόταν και με τα διαγράμματα ενέργειας βραχέως χρόνου.

Τέλος, παρατηρούμε ότι όσο μεγαλώνει η συχνότητα η ενέργεια μικραίνει πράγμα που είδαμε και στο προηγούμενο ερώτημα.

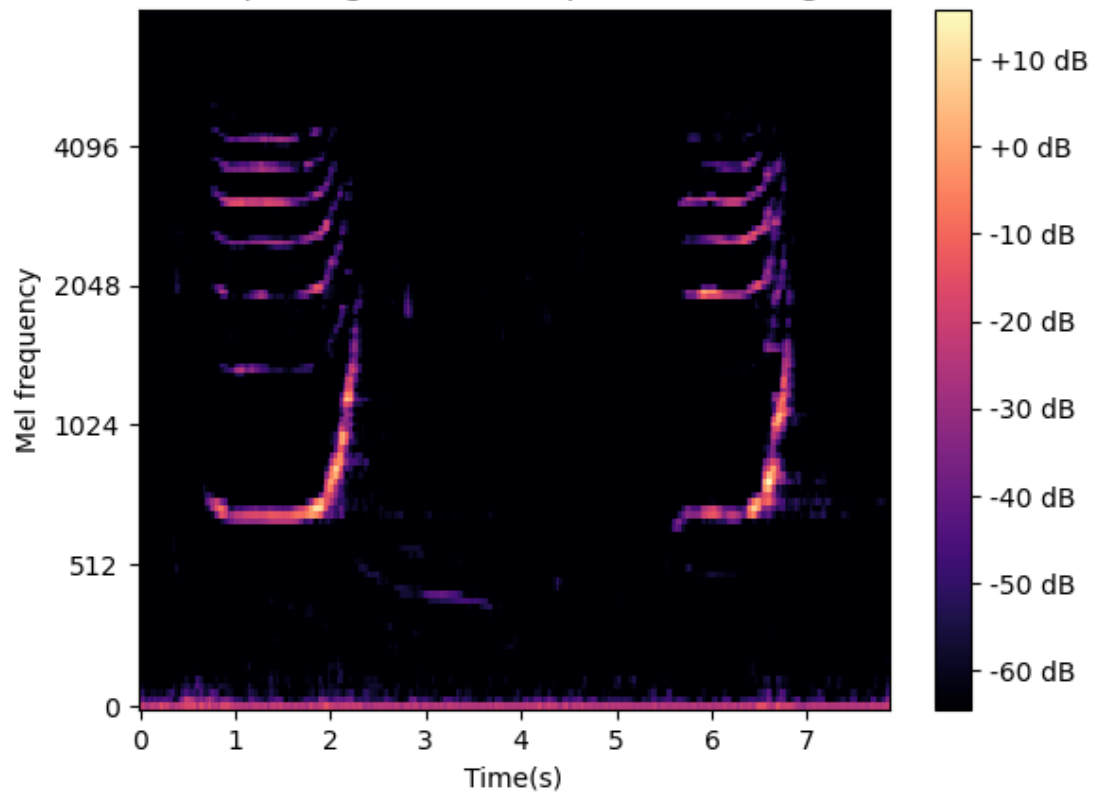
Επομένως, τα δύο αποτελέσματα αντίστοιχα συμφωνούν,



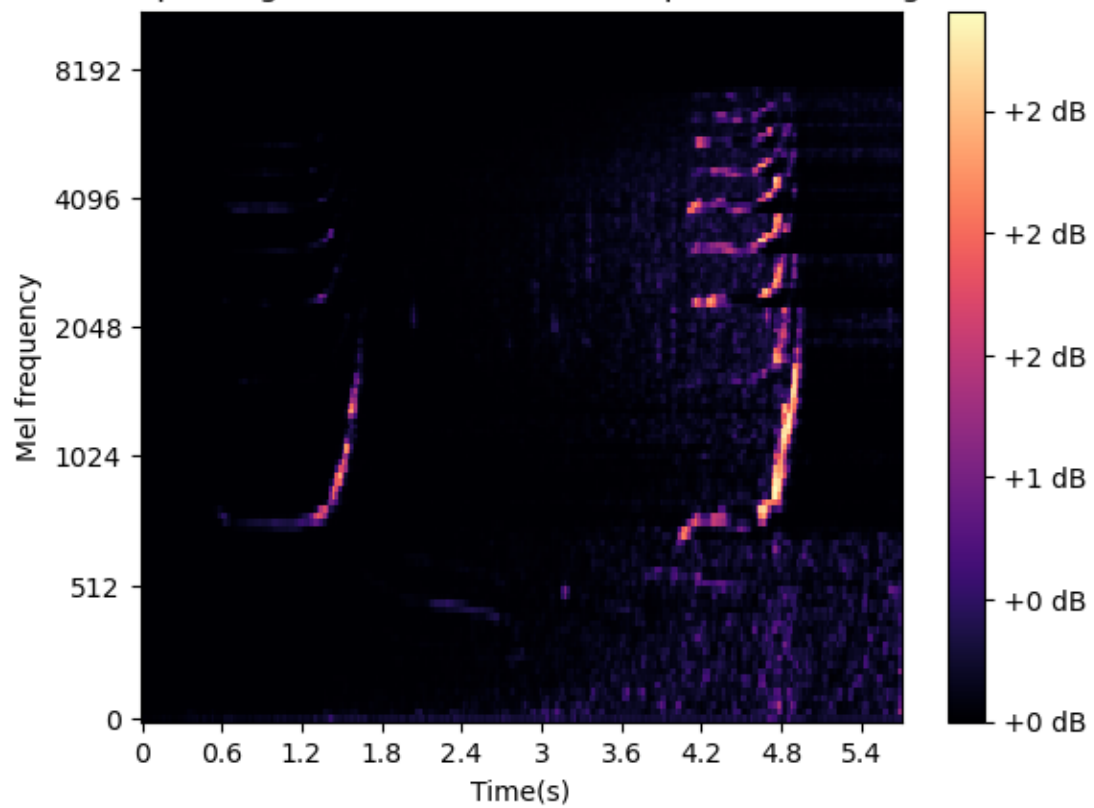
3.10. Δημιουργούμε για κάθε φάλαινα το mel-φασματογράφημα με τη βοήθεια της συνάρτησης `librosa.feature.melspectrogram()` και το σχεδιάζουμε αφού μετατρέψουμε σε dB με την `librosa.amplitude to db()`. Έπειτα εφαρμόσαμε τη συνάρτηση `librosa.pcen()` και κατασκευάσαμε το αντίστοιχο διάγραμμα.



Mel Spectrogram For humpback whale signal



Mel Spectrogram after PCEN for Humpback whale signal



Το φασματογράφημα χωρίς τη μέθοδο pcen εμφανίζει το πραγματικό περιεχόμενο του σήματος με τον όποιο θόρυβο κλπ 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.

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 η ενέργεια κατά μήκος των συχνοτήτων.