Αναφορά πρώτης εργαστηριακής άσκησης Ψηφιακής επεξεργασίας σήματος

Ακαδημαϊκό έτος: 2019-2020

Δέλλης Θανάσης ΑΜ 03117103

Ζορμπάς Τσάλεζας Νικόλας ΑΜ 03117203

Μέρος 1ο

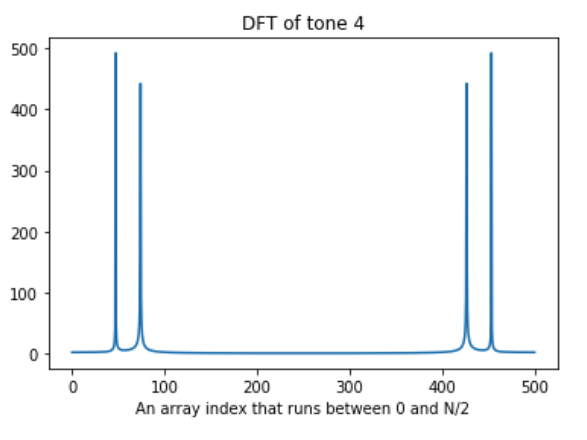
Σύστημα Εντοπισμού Τηλεφωνικών Τόνων

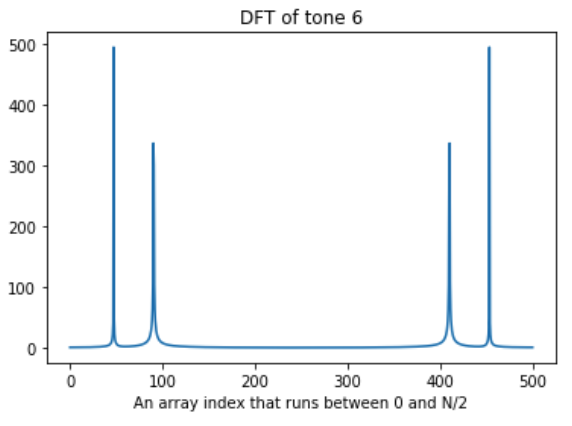
1.1

Αρχικά, φτιάχνουμε δύο πίνακες οπου ο ένας έχει ως στοιχεία τα Ω των γραμμών, ενώ ο δεύτερος τα Ω των στηλών. Με την βοήθεια ενός for loop και καλώντας την play() της sounddevice ακόυμε τους 10 διαφορετικούς τόνους.

1.2

Στο ερώτημα αυτο υπολογίζουμε τον DFT των σημάτων d4[n] και d6[n] και στην συνέχεια κάνουμε τις γραφικές παραστάσεις των μέτρων τους:



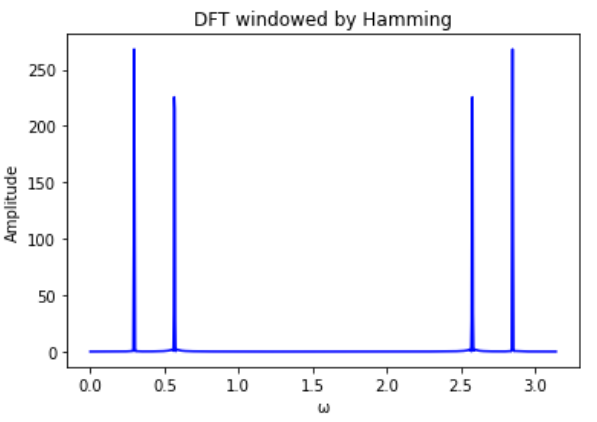


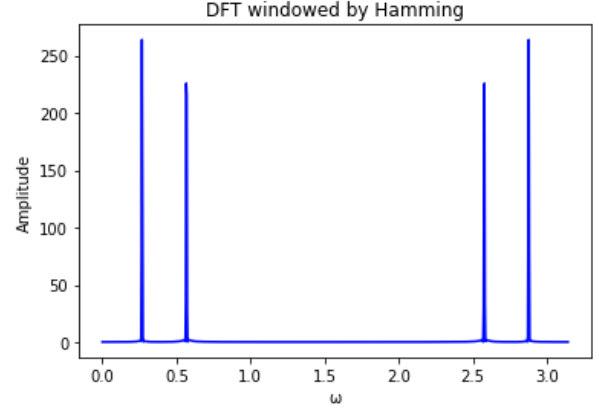
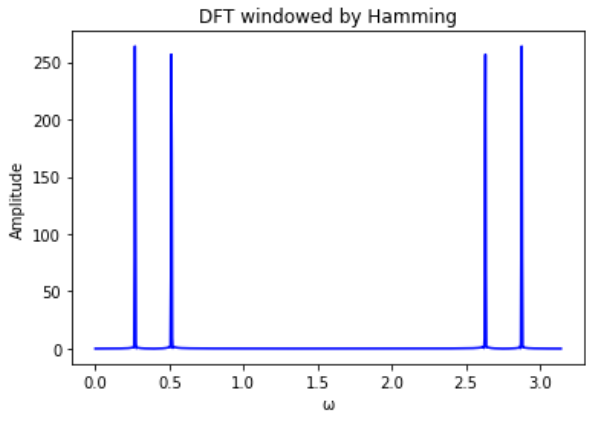
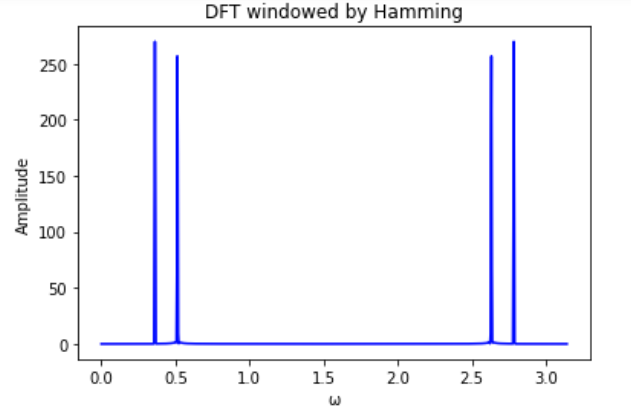
1.3

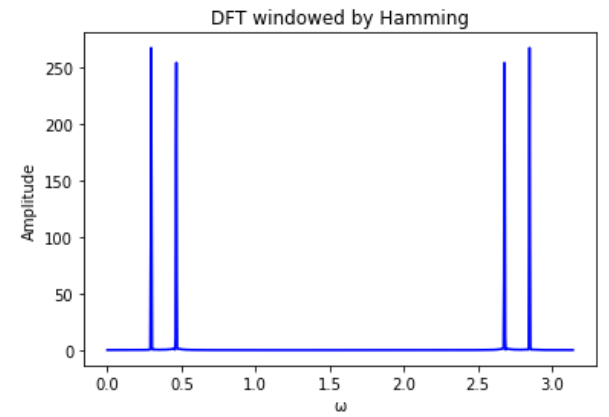
Προσδιορίζουμε την ακολουθία 06234306 προσθέτοντας τα δύο ΑΜ (03117103, 03117203). Στην συνέχεια με την χρήση της συνάρτισης concatenate() της  numpy τοποθετούμε καθε ενα τόνο (πχ. d0 = np.sin(0.7217\*t) + np.sin(1.0247\*t)) σε μία λίστα d. Κάθε τόνος διαχωρίζεται απο τον επόμενο με 100 μηδενικα. Τέλος, αποθηκεύουμε την ακολουθία τόνων (d) στο αρχείο tone\_sequence.wav.

1.4

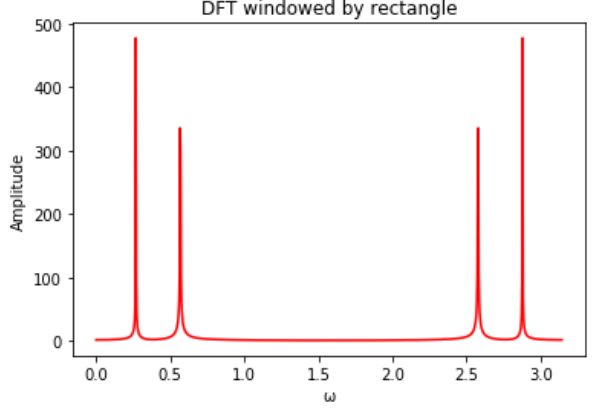
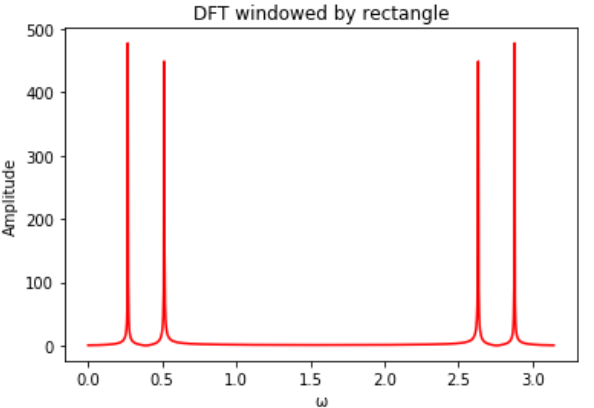
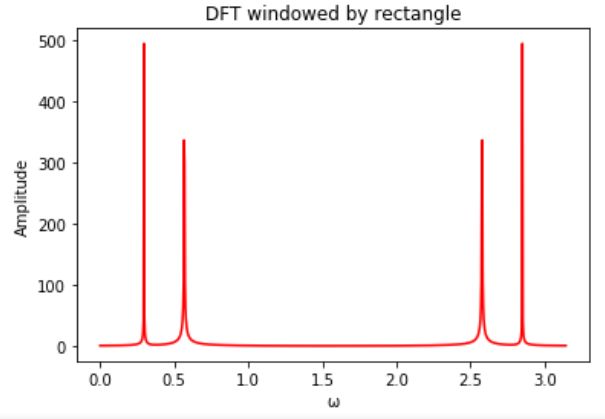
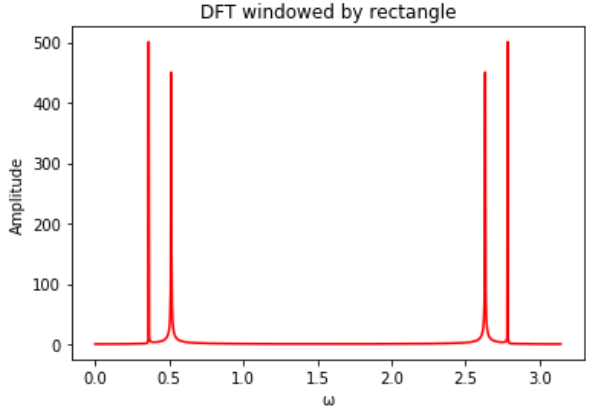
Στο ερώτημα αυτό, αρχικά, ορίζουμε το τετραγωνικό παράθυρο καθώς και το παράθυρο Hamming με N = 1000. Στην συνέχεια υπολογίζουμε τον αριθμό των τόνων που υπάρχουν μέσα στην λίστα μας d (NUM = (d.size / zeros.size)%10 + 1) όπου γνωρίζουμε οτι είναι 8. Με την χρήση for loop πολλαπλασιάζουμε τα 1000 δείγματα κάθε τόνου με το κάθε παράθυρο, εφαρμόζουμε DFT στο γινόμενο τους και κάνουμε plot() το μέτρο αυτού.

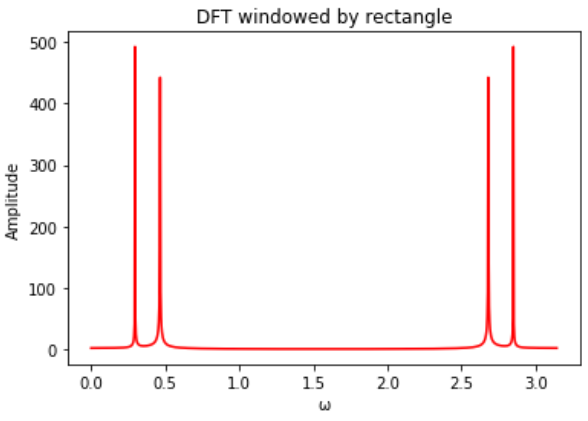
Γραφικές παραστάσεις για το σημα 06234306 με παράθυρο Hamming: (για τους τόνους 0,6,2,3,4 αντίστοιχα)





Γραφικές παραστάσεις για το rectangle παράθυρο (για τους τόνους 0,6,2,3,4 αντίστοιχα):

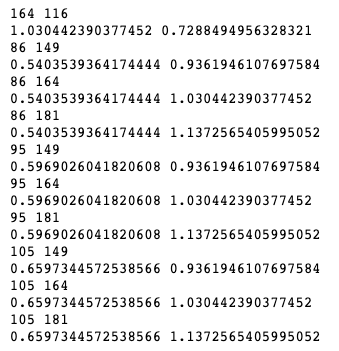




1.5

Για να προσδιορίσουμε τα vectors K, κάνουμε DFT για κάθε sin(Ωt) και στην συνέχεια βρίσουμε την θέση στην οποία παρουσιάζει μέγιστο (np.argmax()). Ως αποτέλεσμα έχουμε 7 θέσεις μεγίστων, οπότε και 7 Κ. Με την χρήση for loop προσθέτουμε σε μορφή τουπλέ (χ,y) μέσα σε μια λίστα numbers[], κάθε συνδυασμό που προσδιορίζει ένα νούμερο. Για δικιά μας διευκόλυνση, προσθέτουμε τα Κ του τόνου 0 πρώτα μέσα στο στην λίστα, και μετά χρησιμοποιούμε το φορ για τα υπόλοιπα 1-9.

Για να προσδιορίσουμε τις αντίστοιχες συχνότητες χρησιμοποιήσαμε τον τύπο ω = 2πΚ/Ν, Ν = 1000. Τέλος, προσθέτουμε τις συχνότητες αυτές, όπως περιγράψαμε και για τα Κ, μέσα σε μια λίστα freq[].

Καθώς η np.argmax() αρχίζει από το 0, προσθέτουμε 1 σε κάθε Κ.

Όπου κάθε δύο γραμμές αντιπροσωπέυουν ένα τόνο, αρχίζοντας απο 0,…,9 με τη πρώτη να ειναι τα Κ και τη δεύτερη οι συχνότητες αντίστοιχα.

1.6

Αρχικά, δημιουργούμε την ttdecode η οποία φτιάχνει ένα array που περιέχει μόνο τους τόνους και αγνοεί τα μηδενικά δείγματα. Στην συνέχεια γνωρίζοντας το μέγεθος του νέου πίνακα μπορούμε να βρούμε τον αριθμό των τόνων που υπάρχουν στο αρχικό σήμα. Με ένα φορ λουπ επαναλαμβάνουμε την διαδικασία του 1.5, όμως για κάθε επανάληψη βρίσουμε τις θέσεις των δύο μεγίστων (αντί του ενός μεγίστου που κάναμε στο 1.5) και τα αποθηκεύουμε σε μια λίστα numdet[]. Τέλος, συγκρίνουμε την λίστα αυτήν, με την λίστα που φτιάξαμε στο 1.5, και κάνουμε πριντ την θέση που βρίσκουμε ομοιότητα ανάμεσα στις δυο.

Ετσι έχουμε για το tone\_seq :

1.7

Απλά κάνουμε load τα αρχεια και με την ttdecode έχουμε:

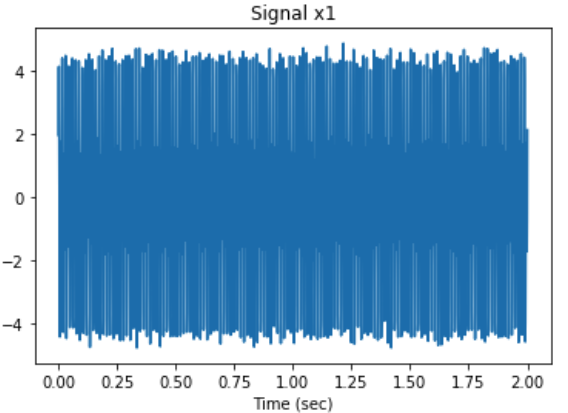
Μέρος 2ο

Φασματική Ανάλυση Ημιτονοειδών και Ανίχνευση Απότομων Μεταβάσεων με τον Μετ/σμό Fourier Βραχέος Χρόνου (STFT) και τον Μετ/σμό Wavelets (διακριτοποιημένο CWT)

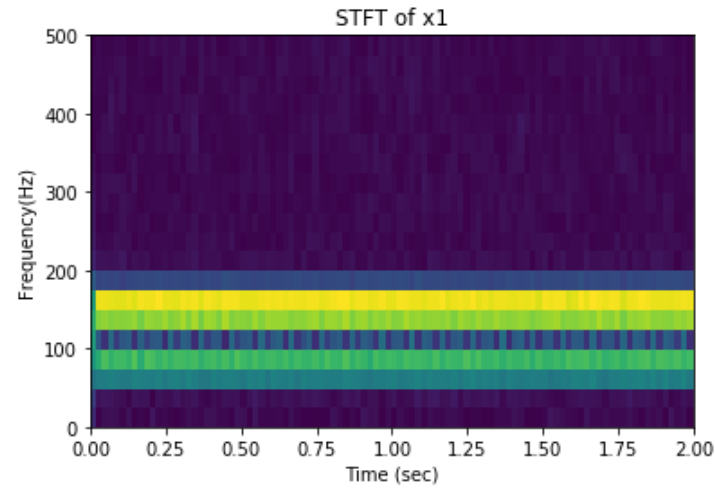
2.1

x(t) = 2 cos(2π70t) + 3 sin(2π140t) + 0.15v(t)

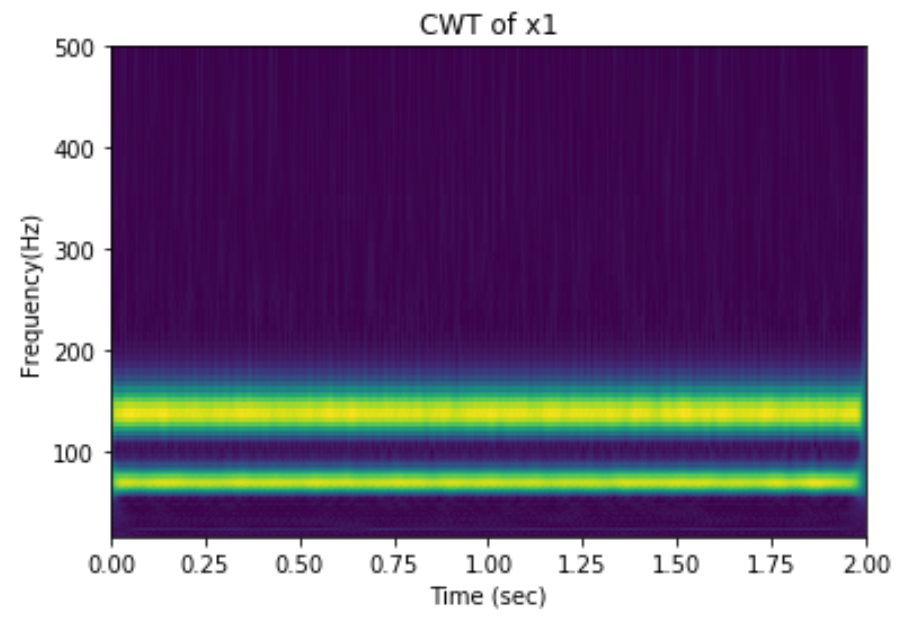
(α)

Φτιάχνουμε τη γραφική παράσταση της συνάρτησης χ. Το γράφημά φαίνεται παρακάτω:

(β)

Υπολογίζοουμε τον STFT του χ με μήκος παραθύρου 0.04s και επικάλυψη 0.02s. Το γράφημά φαίνεται παρακάτω:

(γ)

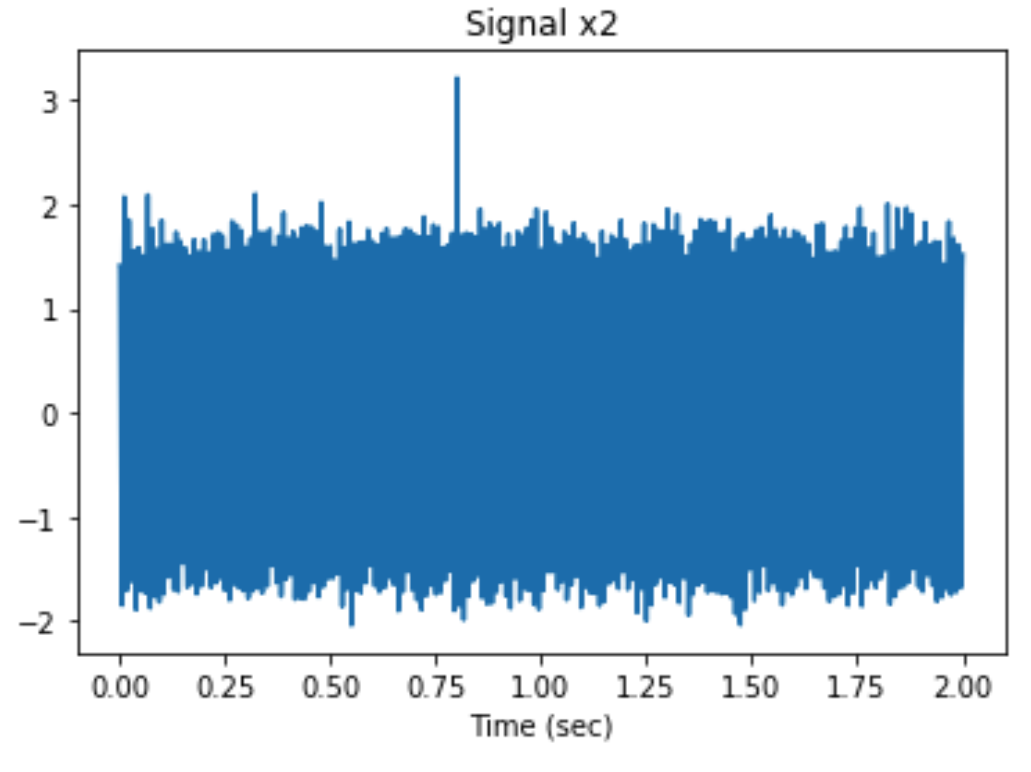
Στο ερώτημα αυτό υπολογίζοουμε τον discretized CWT του χ. Βρίσκουμε ότι η κλίμακα s = 5 αφού γνωρίζουμε την ελάχιστη συχνότητα, την Fs/2 καθώς και τον αριθμό των κυματιδίων ανά οκτάβα(20). Το απεικονίζουμε παρακάτω:

(δ)

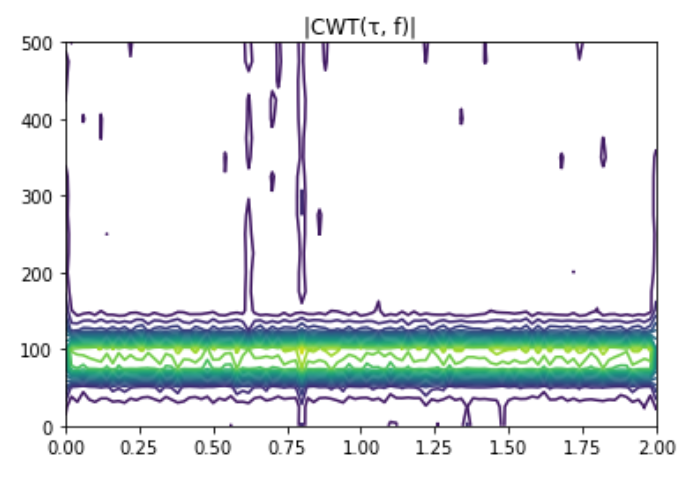
2.2

x(t) = 1.7 cos(2π90t) + 0.15v(t) + 1.7(δ(t − 0.625) + δ(t − 0.800))

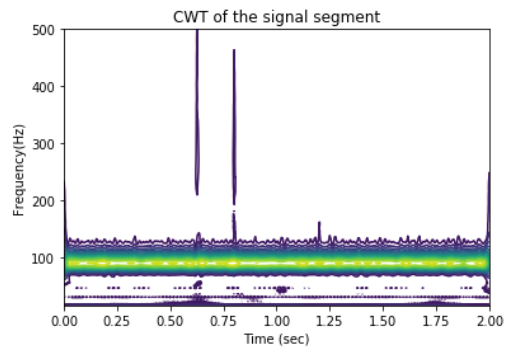
(α)

Φτιάχνουμε τη γραφική παράσταση της συνάρτησης χ. Το γράφημά φαίνεται παρακάτω:

(β)

Υπολογίζοουμε τον STFT του χ με μήκος παραθύρου 0.04s και επικάλυψη 0.02s. Το γράφημά φαίνεται παρακάτω:

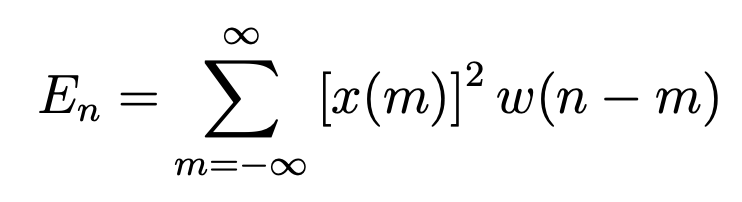
(γ)

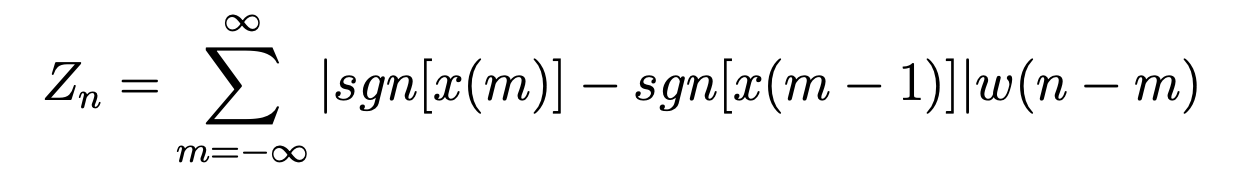
Στο ερώτημα αυτό υπολογίζοουμε τον discretized CWT του χ. Βρίσκουμε ότι η κλίμακα s = 5 αφού γνωρίζουμε την ελάχιστη συχνότητα, την Fs/2 καθώς και τον αριθμό των κυματιδίων ανά οκτάβα(20). Το απεικονίζουμε παρακάτω:

(δ)

Μέρος 3ο

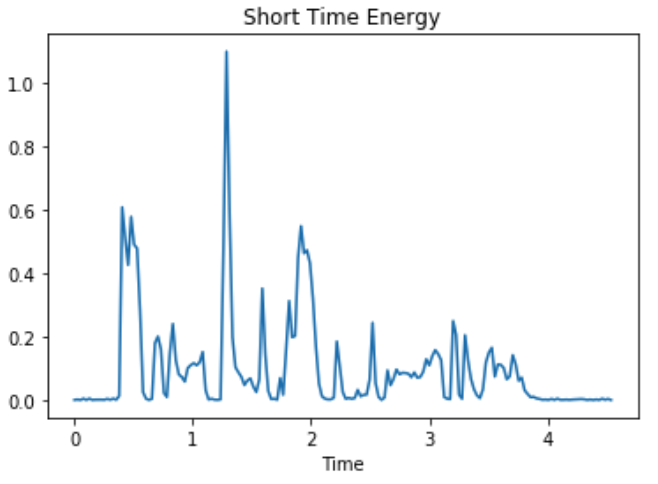
Χαρακτηριστικά Βραχέος Χρόνου Σημάτων Φωνής και Μουσικής (Ενέργεια και Ρυθμός Εναλλαγής Προσήμου)

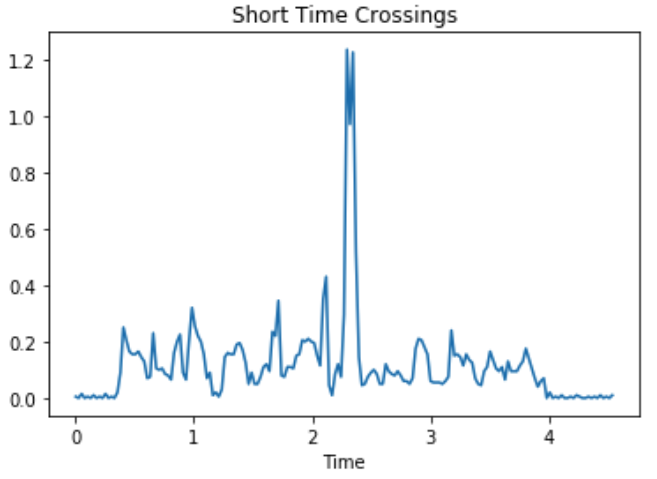
Ενέργειας βραχέος χρόνου:

Ρυθμού εναλλαγής προσήμου:

Αρχικά, φτιάξαμε δύο lists την STEs και την ZCCs όπου αποθηκεύουν αντιστιχα τις τιμές ενέργειας βραχέος χρόνου και ρυθμού εναλλαγής προσήμου. Αυτές τις τιμές τις βρίσκουμε χρησιμοποιόντας παράθυρο μήκους 25 ms και εφαρμόζοντας DFT.

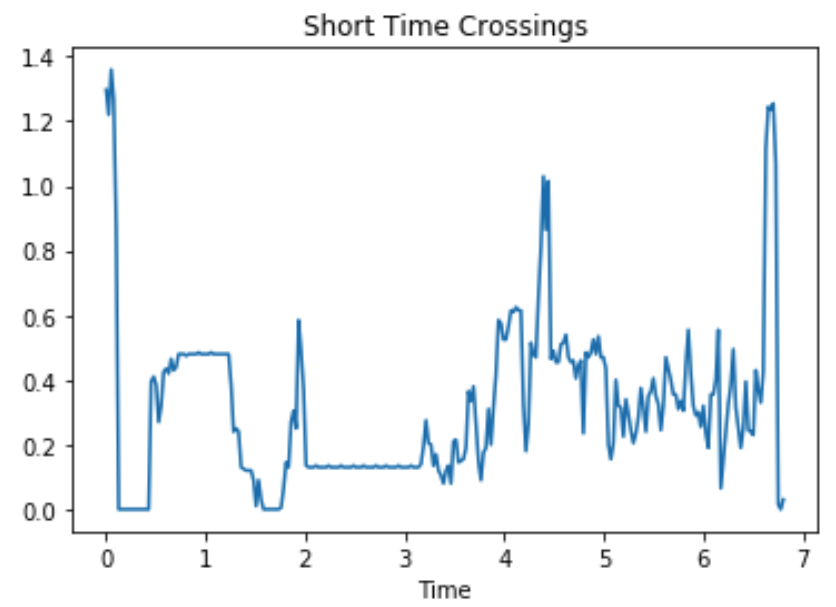
3.1

Οι γραφικές παραστάσεις για την επεξεργασία του αρχείου “speech\_utterance.wav” φαίνονται παρακάτω:



Εύκολα μπορεί κανείς να διαχωρίσει τις περιοχές φωνής από αυτές της σιγής.

Για τα σήματα φωνής, γνωρίζουμε ότι στα φωνήεντα έχουμε μεγάλη ενέργεια και χαμηλό zero crossings, ενώ στα σύμφωνα μεγάλο zero crossings και μικρή ενέργεια.

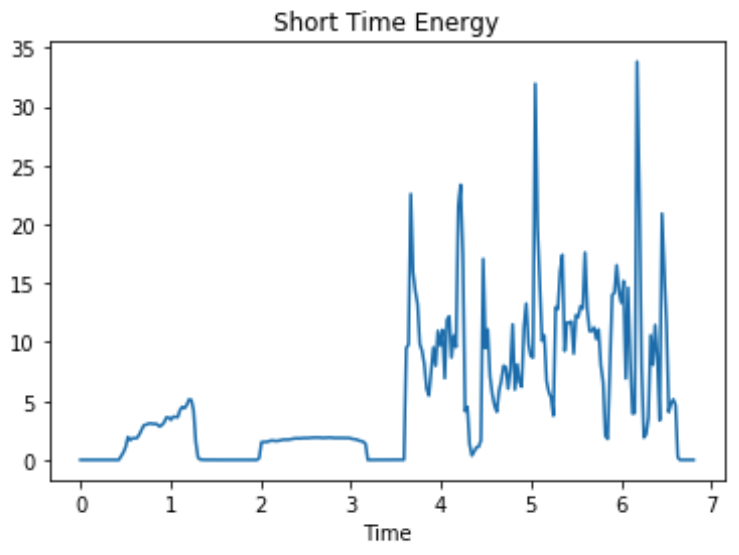
Με την χρήση δύο if statements μπορούμε να χωρίσουμε το σήμα σε φωνήεντα και σύμφωνα. Ανάλογα με το είδος, αποθηκεύουμε την θέση του σε νέα voices[] ή unvoiced[] λίστα. Επαληθεύουμε ότι λειρουργεί σωστά για το αντίστοιχο κομμάτι του χ με την βοήθεια της play().

πχ.

Η unvoiced[]  περιέχει τα στοιχεία …,91,92,93,94,… τα οποία αντιστοιχούν στo χ[91\*Ν:94\*Ν] όπου Ν = 400. Με sd.play(χ[91\*Ν:94\*Ν] ,fs) ακούμε τον ήχο του γράμματος Σ.

H  voiced[] περιέχει το στοιχείο …,63,… το οποίο αντιστοιχεί στo χ[62.5\*Ν:63.5\*Ν] όπου Ν = 400. Με sd.play(χ[62.5\*Ν:63.5\*Ν] ,fs) ακούμε τον ήχο του γράμματος Η.

3.2

Οι γραφικές παραστάσεις για την επεξεργασία του αρχείου “speech\_utterance.wav” φαίνονται παρακάτω: