

Εξηγείστε περιεκτικά και επαρκώς την εργασία σας. Επιτρέπεται προαιρετικά η συνεργασία εντός ομάδων των 2 ατόμων. Κάθε ομάδα 2 ατόμων υποβάλλει μια κοινή αναφορά που αντιπροσωπεύει μόνο την προσωπική εργασία των μελών της. Αν χρησιμοποιήσετε κάποια άλλη πηγή εκτός του βιβλίου και του εκπαιδευτικού υλικού του μαθήματος, πρέπει να το αναφέρετε. Η παράδοση της αναφοράς και του κώδικα της εργασίας θα γίνει ηλεκτρονικά στο mycourses.ntua.gr. Επισημαίνεται ότι απαγορεύεται η ανάρτηση των λύσεων των εργαστηριακών ασκήσεων στο github, ή άλλες ιστοσελίδες.

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

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

Στόχος αυτής της άσκησης είναι να παρουσιάσει πως λειτουργεί το τηλεφωνικό τονικό σύστημα χρησιμοποιώντας σήματα διαφορετικών συχνοτήτων για να εντοπίζει ποιο πλήκτρο έχει πατηθεί. Ο εντοπισμός αυτών των συχνοτήτων μπορεί να γίνει με την χρήση του Διακριτού Μετ/σμού Fourier (DFT) $X[k]$ του τηλεφωνικού σήματος $x[n]$. Με το πάτημα ενός πλήκτρου στο τηλέφωνο ακούγεται ένας ήχος που είναι το άθροισμα 2 ημιτόνων, το υψίσυχο ημίτονο δείχνει την στήλη που ανήκει το πλήκτρο στο touch-pad της τηλεφωνικής συσκευής και το χαμηλόσυχο ημίτονο δείχνει την αντίστοιχη γραμμή του touch-pad σύμφωνα με τον Πίνακα που ακολουθεί. Παραδείγματος χάρη, το πλήκτρο 5 αντιστοιχεί στο σήμα $d_5[n] = \sin(0.5906n) + \sin(1.0247n)$

	Ω_{column}		
Ω_{row}	0.9273	1.0247	1.1328
0.5346	1	2	3
0.5906	4	5	6
0.6535	7	8	9
0.7217		0	

Πίνακας 1: Διακριτές Συχνότητες για Τηλεφωνικούς Τόνους για Συχνότητα Δειγματοληψίας 8192 Hz.

- 1.1. Δημιουργήστε τους 10 διαφορετικούς τόνους σύμφωνα με την παραπάνω εκφώνηση. Κάθε τόνος πρέπει να έχει μήκος 1000 δείγματα. (Υπόδειξη: Μπορείτε να χρησιμοποιήσετε την ρουτίνα **play()** από το πακέτο sounddevice για να ακούσετε τους ήχους.)
- 1.2. Υπολογίστε τον DFT των σημάτων $d_4[n]$ και $d_6[n]$ και δημιουργήστε τις γραφικές παραστάσεις των $|D_4[k]|$ και $|D_6[k]|$.
- 1.3. Δημιουργήστε και αποθηκεύστε σε αρχείο “tone_sequence.wav” με την **write_wav()** της Librosa ένα σήμα με διαδοχικούς τηλεφωνικούς τόνους ‘μεταφράζοντας’ το άθροισμα των αριθμών μητρώου των μελών της κάθε ομάδας σε τονικά σήματα. Κάθε ένα ψηφίο του αθροίσματος πρέπει να διαχωρίζεται από το προηγούμενο με 100 μηδενικά δείγματα. Για παράδειγμα αν οι αριθμοί μητρώου των δύο μελών είναι $AM_1 = 03092432$ και $AM_2 = 03093543$ τότε τα ζητούμενα ψηφία είναι: 0 6 1 8 5 9 7 5 (= 03092432 + 03093543). Αν η εργασία υλοποιείται από ένα άτομο τότε τα ζητούμενα ψηφία είναι ο αριθμός μητρώου.

- 1.4. Χρησιμοποιώντας την συνάρτηση **fft()** της numpy και κατάλληλα χρονικά παράθυρα, (i.) τετραγωνικά, και (ii.) Hamming, μήκους $N = 1000$, υπολογίστε τον Μετ/σμό Fourier των παραθυροποιημένων σημάτων, όπως προκύπτουν από το προηγούμενο σήμα. (Υπόδειξη: Δημιουργήστε τόσα παραθυροποιημένα σήματα όσοι και οι τόνοι που περιέχονται στο αρχικό σήμα).
 - 1.5. Υπολογίστε μια λίστα από δείκτες k και τις αντίστοιχες συχνότητες που θεωρείτε ότι βρίσκονται εγγύτερα στις touch-tone συχνότητες.
 - 1.6. Δημιουργήστε μια συνάρτηση με το όνομα **ttdecode**, που θα δέχεται σαν όρισμα εισόδου ένα τονικό σήμα (όπως περιγράφηκε στο **Ερωτ. 1.3**) και επιστρέφει ένα διάνυσμα με τα αντίστοιχα ψηφία. Για παράδειγμα αν το σήμα εισόδου **signIn** περιέχει τους τόνους για το νούμερο 210 – 3434120, η έξοδος της συνάρτησης **Vector** θα είναι $Vector = 2\ 1\ 0\ 3\ 4\ 3\ 4\ 1\ 2\ 0$. Επιβεβαιώστε την ορθή λειτουργία της ρουτίνας θέτοντας σαν είσοδο, το σήμα του **Ερωτ. 1.3**.
- Υπόδειξη: Η συνάρτηση θα πρέπει πρώτα να υπολογίζει την ενέργεια κάθε ενός από τους τόνους του σήματος εισόδου με την χρήση της ρουτίνας **fft()**. Έπειτα να εντοπίζει ποιες είναι εκείνες οι συχνότητες που έχουν την μεγαλύτερη ενέργεια και να τις αντιστοιχίζει στις αρχικές συχνότητες του Πίνακα 1. Με βάση αυτές τις αντιστοιχίσεις, εντοπίστε ποιο είναι το αντίστοιχο ψηφίο. Υπενθυμίζεται ότι η ενέργεια E_k του σήματος γύρω από τη συχνότητα με δείκτη k ισούται με $E_k = |X[k]|^2$.
- 1.7. Με χρήση της εντολής **load()** της numpy φορτώστε τα αρχεία easySig.npy, hardSig.npy από το συμπληρωματικό υλικό της άσκησης “dsp20_lab1_Data.zip” που βρίσκεται στο mycourses. Στο work space θα πρέπει να υπάρχουν 2 ξεχωριστά σήματα, το *easySig* και *hardSig*. Προσδιορίστε τα ψηφία στα οποία αντιστοιχούν οι τόνοι των 2 σημάτων με την χρήση της ρουτίνας **ttdecode()**.

Σημειώνεται ότι σε κάθε περίπτωση οι απαντήσεις σας πρέπει να συνοδεύονται με τις σχετικές γραφικές παραστάσεις και σχόλια ώστε να είναι όσο το δυνατό τεκμηριωμένες.

Μέρος 2ο - Φασματική Ανάλυση Ημιτονοειδών και Ανίχνευση Απότομων Μεταβάσεων με τον Μετ/σμό Fourier Βραχέος Χρόνου (STFT) και τον Μετ/σμό Wavelets (διακριτοποιημένο CWT)

Ο STFT είναι ο πιο διαδεδομένος μετ/σμός για την μελέτη του συχνοτικού περιεχομένου σημάτων σε μικρά χρονικά διαστήματα με υπολογισμό του Μετ/σμού Fourier Διακριτού Χρόνου (DTFT) σε καθένα από αυτά. Αναλυτικά:

$$\text{STFT}(\tau, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n - \tau]e^{-j\omega n}, \text{ όπου } \tau \in Z.$$

Στην υλοποίηση του STFT στη Librosa, χρησιμοποιείται ο DFT (και όχι ο DTFT), οπότε η συχνότητα ω δειγματοληπτείται στα $\omega_k = 2\pi k/M$, όπου M το μήκος του DFT μεγαλύτερο ή ίσο από το μήκος L του παραθύρου. Το μήκος L του παραθύρου $w[n]$ που θα χρησιμοποιηθεί καθορίζει την σχέση μεταξύ της διακριτικής ικανότητας στη συχνότητα και της ανάλυσης στο χρόνο. Μικρό παράθυρο πετυχαίνει καλή ανάλυση στο χρόνο με όμως χειρότερη διακριτική ικανότητα στην συχνότητα, ενώ αντίστροφα, μεγάλο παράθυρο στο χρόνο πετυχαίνει καλή

διακριτική ικανότητα στο πεδίο συχνοτήτων χάνοντας σε ανάλυση στον χρόνο. Ένα από τα χαρακτηριστικά του STFT είναι ότι το μήκος L του παραθύρου w είναι σταθερό και επιλέγεται εξαρχής. Συνήθεις τιμές για εφαρμογές narrow-band επεξεργασίας φωνής είναι $N=30-50\text{msec}$ (για wide-band $\approx 5\text{msec}$).

Ένας εναλλακτικός τρόπος αναπαράστασης των σημάτων είναι με τον Μετ/σμό των Wavelets. Στην περίπτωση σημάτων συνεχούς χρόνου ορίζεται ως:

$$\text{CWT}(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt,$$

όπου $\psi(t)$ είναι η βασική-μητρική συνάρτηση η οποία μπορεί να μετατοπιστεί κατά τ και να μεγεθυνθεί ή σμικρυνθεί κατά s . Στην περίπτωση διακριτών σημάτων, μπορεί να εφαρμοστεί η διακριτοποιημένη μορφή του (Discretized Continuous Wavelet Transform)¹ (ρουτίνα **cwt()** στο πακέτο pywavelets). Στην περίπτωση των Wavelets δεν υπάρχει ο περιορισμός του σταθερού μήκους παραθύρου όπως συνέβαινε στον STFT.

2.1. Έστω ότι δειγματοληπτούμε με $F_s = 1000\text{Hz}$ το σήμα $x(t)$ στο διάστημα $[0, 2]$ sec:

$$x(t) = 2 \cos(2\pi 70t) + 3 \sin(2\pi 140t) + 0.15v(t)$$

όπου $v(t)$ λευκός Gaussian θόρυβος μηδενικής μέσης τιμής (υλοποιείται με την συνάρτηση **normal()** της numpy). Ισχύει ότι $x[n] = x(nT_s)$.

(α) Υπολογίστε το σήμα $x[n]$ και κάνετε την γραφική του παράσταση με χρήση της ρουτίνας **plot**.

(β) Υπολογίστε τον STFT με χρήση της ρουτίνας **stft()** της Librosa. Χρησιμοποιήστε μήκος παραθύρου ίσο με 0.04sec και επικάλυψη ίση με 0.02sec . Αναπαραστήστε το πλάτος $|\text{STFT}(\tau, f)|$ με τις κατάλληλες τιμές στους άξονες του χρόνου και των συχνοτήτων, με χρήση της συνάρτησης **pcolormesh()** της matplotlib.

(γ) Υπολογίστε τον discretized CWT με χρήση της ρουτίνας **cwt()**, επιλέγοντας το wavelet “cmor3.0-1.0”. Ορίστε τις κλίμακες s ώστε να έχουν ανάλυση 20 κυματιδίων ανά οκτάβα, και η ελάχιστη συχνότητα να ισούται με 15.625 Hz . Αναπαραστήστε τα πλάτη $|\text{CWT}(\tau, f)|$ (συναρτήσει των συχνοτήτων σε Hz) και $|\text{CWT}(\tau, s)|$ (συναρτήσει των κλιμάκων του μετασχηματισμού) βαθμονομώντας κατάλληλα τους άξονες, με χρήση της συνάρτησης **pcolormesh()**.

(δ) Τί παρατηρείτε;

2.2. Έστω ότι δειγματοληπτούμε με $F_s = 1000\text{Hz}$ το σήμα $x(t)$ στο διάστημα $[0, 2]$ sec:

$$x(t) = 1.7 \cos(2\pi 90t) + 0.15v(t) + 1.7(\delta(t - 0.625) + \delta(t - 0.800))$$

όπου $v(t)$ λευκός θόρυβος μηδενικής μέσης τιμής όμοια με πριν. Ισχύει ότι $x[n] = x(nT_s)$.

Στις χρονικές στιγμές 625msec και 800msec το σήμα είχε απότομες μεταβολές, τις οποίες θέλουμε να εντοπίσουμε.

(α) Υπολογίστε το σήμα $x[n]$ και κάνετε την γραφική του παράσταση με χρήση της ρουτίνας **plot()**.

(β) Υπολογίστε τον STFT με χρήση της ρουτίνας **stft()**. Χρησιμοποιείστε μήκος παραθύρου ίσο με 0.04sec και επικάλυψη ίση με 0.02sec . Αναπαραστήστε το πλάτος $|\text{STFT}(\tau, f)|$, με τις κατάλληλες τιμές στους άξονες του χρόνου και των συχνοτήτων, με χρήση της συνάρτησης **contour()** της matplotlib.

(γ) Υπολογίστε τον discretized CWT με χρήση της ρουτίνας **cwt()**, επιλέγοντας το wavelet “cmor3.0-1.0”. Ορίστε τις κλίμακες s ώστε να έχουν ανάλυση 20 κυματιδίων ανά οκτάβα, και η ελάχιστη συχνότητα να ισούται με 15.625 Hz . Αναπαραστήστε τα πλάτη $|\text{CWT}(\tau, f)|$

¹Πρακτικά, εφαρμόζουμε τον CWT σε σήματα διακριτού χρόνου, δειγματοληπτώντας επιπλέον τη μεταβλητή κλίμακας, s .

(συναρτήσει των συχνοτήτων σε Hz) και $|CWT(\tau, s)|$ (συναρτήσει των κλιμάκων του μετασχηματισμού) βαθμονομώντας κατάλληλα τους άξονες, με χρήση της συνάρτησης **contour()**.
(δ) Τί παρατηρείτε;

Μέρος 3ο - Χαρακτηριστικά Βραχέος Χρόνου Σημάτων Φωνής και Μουσικής (Ενέργεια και Ρυθμός Εναλλαγής Προσώμου)

Οι μετρήσεις βραχέος χρόνου είναι μετρήσεις που γίνονται σε ένα μετακινούμενο παράθυρο του σήματος. Πιο συγκεκριμένα, η ενέργεια βραχέος χρόνου ορίζεται ως:

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)]^2 w(n-m) \quad (1)$$

όπου w ένα παράθυρο της επιλογής μας, το οποίο συνήθως είναι το Hamming παράθυρο (συνάρτηση **hamming()** στη numpy). Αντίστοιχα, ο ρυθμός εναλλαγής προσώμου (Zero Crossing Rate), για την περίπτωση που χρησιμοποιήσουμε τετραγωνικό παράθυρο N δειγμάτων, με πλάτος $1/2N$, ορίζεται ως:

$$Z_n = \sum_{m=-\infty}^{\infty} |\text{sgn}[x(m)] - \text{sgn}[x(m-1)]| w(n-m) \quad (2)$$

- 3.1. Θεωρήστε το σήμα φωνής της πρότασης ‘Αλλά και για να επέμβει χρειάζεται συγκεκριμένη καταγγελία’ που περιέχεται στο αρχείο “speech_utterance.wav” (συχνότητα δειγματοληψίας: 16 kHz) του συμπληρωματικού υλικού “dsp20_lab1_Data.zip” της άσκησης στο mycourses. Ο στόχος είναι να μετρήσετε την ενέργεια βραχέος χρόνου και το ρυθμό εναλλαγής προσώμου. Χρησιμοποιήστε παράθυρα μήκους 20-30 ms. Τί παρατηρείτε μεγαλώνοντας το μήκος του παραθύρου; Μπορείτε να χρησιμοποιήσετε αυτές τις μετρήσεις για να διαχωρίσετε φωνή από σιωπή ή έμφωνους (π.χ. /aa/, /ih/) από άφωνους ήχους (π.χ. /f/, /p/);
- 3.2. Επαναλάβετε την ίδια διαδικασία για το σήμα μουσικής “music.wav” που επίσης βρίσκεται στο συμπληρωματικό υλικό.

Μέρος 4ο - Πολυκλιμακωτή Ανάλυση Σημάτων μέσω του Διακριτού Μετασχηματισμού Wavelets για τη Ρυθμική Ανάλυση Τραγουδιών²

Στα μουσικά σήματα, προσεγγιστικά περιοδική συμπεριφορά εμφανίζεται σε μία πληθώρα χρονικών κλιμάκων. Από αυτές, οι δύο κυριότερες αφορούν την περιοδικότητα σε επίπεδο νότας (όπου η συχνότητα μετράται σε Hertz), και τη ρυθμική ομοιότητα, που αφορά στη διαδοχή παλμών στο μουσικό σήμα, η οποία μετράται σε παλμούς ανά λεπτό (Beats per Minute - BPM). Στο ερώτημα αυτό, θα υλοποιήσουμε σε Python έναν αλγόριθμο για την εξαγωγή του ρυθμού από μουσικά σήματα με χρήση του Διακριτού Μετασχηματισμού Wavelets (Discrete Wavelet Transform - DWT), παρόμοιο με αυτόν που περιγράφεται στο [1]³. Πρακτικά, ο DWT αφορά στην ανάλυση ενός σήματος, με φασματικό περιεχόμενο στο διάστημα $[0, f_{max}]$ σε δύο συνιστώσες - μία η οποία θα περιέχει το χαμηλόσυχο περιεχόμενο του σήματος (στο διάστημα $[0, f_{max}/2]$) και μία η οποία θα περιέχει το υψίσυχο περιεχόμενο του σήματος (στο διάστημα $[f_{max}/2, f_{max}]$). Επανάληψη αυτής της διαδικασίας, διαδοχικά, στη χαμηλόσυχη συνιστώσα του σήματος που προκύπτει κάθε φορά οδηγεί σε μία ανάλυση του φασματικού περιεχομένου του σήματος σε πολλαπλές κλίμακες, αντίστοιχα με τον discretized CWT, όπου οι κλίμακες παίρνουν τις τιμές 2^k , $k = 1 \dots N$, για N επίπεδα ανάλυσης.

²Το ερώτημα αυτό είναι προαιρετικό, και επίλυσή του συνεισφέρει βαθμολογικό bonus μέχρι και 25% επί του βαθμού της άσκησης.

³[1]: Tzanetakis, George, and Perry Cook. “Musical Genre Classification of Audio Signals.” IEEE Transactions on Speech and Audio Processing 10.5 (2002): 293-302.

- 4.1 Φορτώστε στον υπολογιστή σας το αρχείο “salsa_excerpt1.mp3” από το συμπληρωματικό υλικό της άσκησης, με συχνότητα δειγματοληψίας 22.05 kHz. Απομονώστε ένα απόσπασμα μήκους $L = 2^{16}$ δειγμάτων (περίπου 3 δευτερολέπτων) από αυτό. Σχεδιάστε το απόσπασμα αυτό, και προσπαθήσετε να εκτιμήσετε την παλμική περιοδικότητά του.
- 4.2 Εφαρμόστε για το απόσπασμα αυτό τον Διακριτό Μετασχηματισμού Wavelets, ώστε να εξάγετε την πολυκλιμακωτή του ανάλυση σε συνιστώσες. Χρησιμοποιήστε κυματίδια Daubechies-4 (ρουτίνα **dwt()** στο πακέτο pywavelets, με όρισμα “db4” ως τύπο κυματιδίου), και συνολικά 7 επίπεδα ανάλυσης. Με το πέρας της διαδικασίας, θα πρέπει να έχετε 7 συνιστώσες λεπτομέρειας (detail), $y_{di}[n], i = 1...7$, και μία χαμηλόσυχη συνιστώσα approximation, $y_{a7}[n]$.
- Υπόδειξη:** Για τη διατήρηση των σημάτων, μετά την εφαρμογή του DWT, στην ίδια κλίμακα, διαιρέστε τις συνιστώσες y_a, y_d κατά τον παράγοντα $\sqrt{2}$.
- 4.3 Για την κάθε συνιστώσα, εξάγετε την περιβάλλουσά της. Η διαδικασία αυτή αποτελείται από τα εξής βήματα:
- α) Υπολογισμός της απόλυτης τιμής του σήματος, $z_i[n] = |y_i[n]|$.
 - β) Φιλτράρισμα με χρήση βαθυπερατού φίλτρου, ώστε

$$x_i[n] = (1 - a)x_i[n - 1] + az_i[n] \quad (3)$$

Η παράμετρος $a = a_0 2^k$, όπου k το επίπεδο ανάλυσης που βρισκόμαστε. Η επιλογή αυτή μας εξασφαλίζει ομοιομορφία στη χρονική κλίμακα του βαθυπερατού φίλτρου. Για την παράμετρο a_0 , πειραματιστείτε με τις τιμές $\{0.001, 0.002, 0.005\}$.

γ) Κεντράρισμα του σήματος $x_i[n]$, αφαιρώντας από αυτό τη μέση τιμή του.

Δώστε με τη μορφή διαγραμμάτων τα σήματα $y_{d3}[n], y_{d6}[n]$, μαζί με τις περιβάλλουσές τους $x_{d3}[n], x_{d6}[n]$.

- 4.4 Αθροίστε τις περιβάλλουσες των συνιστωσών DWT που προέκυψαν από το ερώτημα 4.3. Για να αποκτήσουν οι περιβάλλουσες το ίδιο μήκος, προτείνεται να εφαρμόσετε γραμμική παρεμβολή, μέσω της εντολής **interp()** στη numpy. Στη συνέχεια, υπολογίστε και σχεδιάστε την αυτοσυσχέτιση του αθροίσματος αυτού (εντολή **correlate()** στη numpy). Τι παρατηρείτε ως προς τη μορφή του;
- 4.5 Από τα peaks του διάγραμμα αυτοσυσχέτισης, εξάγετε το ρυθμό (σε Beats per Minute - BPM) του αποσπάσματος που επιλέξατε. Για να βρείτε τις τιμές των μεγίστων (σε samples) της συνάρτησης αυτοσυσχέτισης, χρησιμοποιήστε τη ρουτίνα **argrelextrema()** της scipy. Σε περίπτωση που το διάγραμμα αυτοσυσχέτισης είναι αρκετά τραχύ, μπορείτε να το εξομαλύνετε με χρήση Γκαουσιανού φίλτρου (ρουτίνα scipy: **scipy.ndimage.gaussian_filter1d()**). Επιβεβαιώστε το αποτέλεσμα σας και ακουστικά.
- 4.6 Επαναλάβετε την παραπάνω διαδικασία για τα κομμάτια “salsa_excerpt2.mp3” και “rumba_excerpt.mp3” από το συμπληρωματικό υλικό της άσκησης. Σχολιάστε τα αποτελέσματά σας ως προς την ικανότητα διαχωρισμότητας τραγουδιών που ανήκουν σε διαφορετικά χορευτικά είδη.

ΠΑΡΑΔΟΤΕΑ Ηλεκτρονική παράδοση του κώδικα Python, της ακολουθίας touch-tone υπο μορφή .wav που φτιάξατε στο ερώτημα 1.3 του Μέρους 1, καθώς και συνοπτικής αναφοράς που θα απαντάει στα δοθέντα ερωτήματα και θα περιλαμβάνει τις ζητούμενες γραφικές αναπαραστάσεις.