



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ψηφιακή Επεξεργασία Σήματος

Ροή Σ: Σήματα, Έλεγχος, Ρομποτική

6^ο Εξάμηνο

1^η Εργαστηριακή Άσκηση

*Εισαγωγή στη Ψηφιακή Επεξεργασία Σήματος με
python και Εφαρμογές σε Ακουστικά Σήματα*

Χρήστος Δημόπουλος – 031 17 037

Δημήτρης Δήμος – 031 17 165

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

Σκοπός Άσκησης

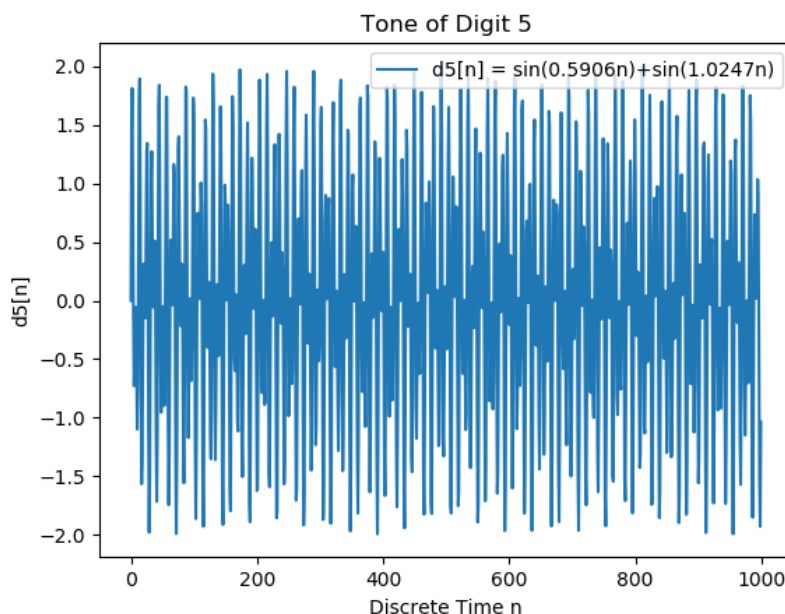
Στόχος της συγκεκριμένης άσκησης είναι να παρουσιάσει το τηλεφωνικό τονικό σύστημα, χρησιμοποιώντας σήματα διαφορετικών συχνοτήτων για να εντοπίζει ποιο πλήκτρο έχει πατηθεί. Ο εντοπισμός αυτών των συχνοτήτων πραγματοποιείται με χρήση του Διακριτού Μετασχηματισμού Fourier (DFT) $X[k]$ του τηλεφωνικού σήματος $x[n]$. Με το πάτημα ενός πλήκτρου στο τηλέφωνο παράγεται ένας ήχος που είναι το άθροισμα 2 ημιτόνων: ενός υψίσυχνου που δείχνει την στήλη που ανήκει το πλήκτρο στο touch-pad και ενός χαμηλόσυχνου που δηλώνει την αντίστοιχη γραμμή του touch-pad.

1.1 Δημιουργία των 10 διαφορετικών τόνων

Αρχικά, παρατίθεται ο πίνακας σύμφωνα με τον οποίο γίνεται η αντιστοίχιση του υψίσυχνου ημιτόνου ενός τόνου με την αντίστοιχη στήλη, καθώς και του χαμηλόσυχνου ημιτόνου με την αντίστοιχη γραμμή του touch-pad. Ειδικότερα:

	Ωcolumn		
Ωrow	0.9273	1.0247	1.1328
0.5346	1	2	3
0.5906	4	5	6
0.6535	7	8	9
0.7217		0	

Με βάση τον παραπάνω πίνακα δημιουργήσαμε τους 10 διαφορετικούς τόνους, καθένας εκ των οποίων έχει διάρκεια 1000 δειγμάτων. Ενδεικτικά, παρατίθεται η γραφική παράσταση του τόνου που αντιστοιχεί στο πλήκτρο 5, $d_5[n] = \sin(0.5906n) + \sin(1.0247n)$:



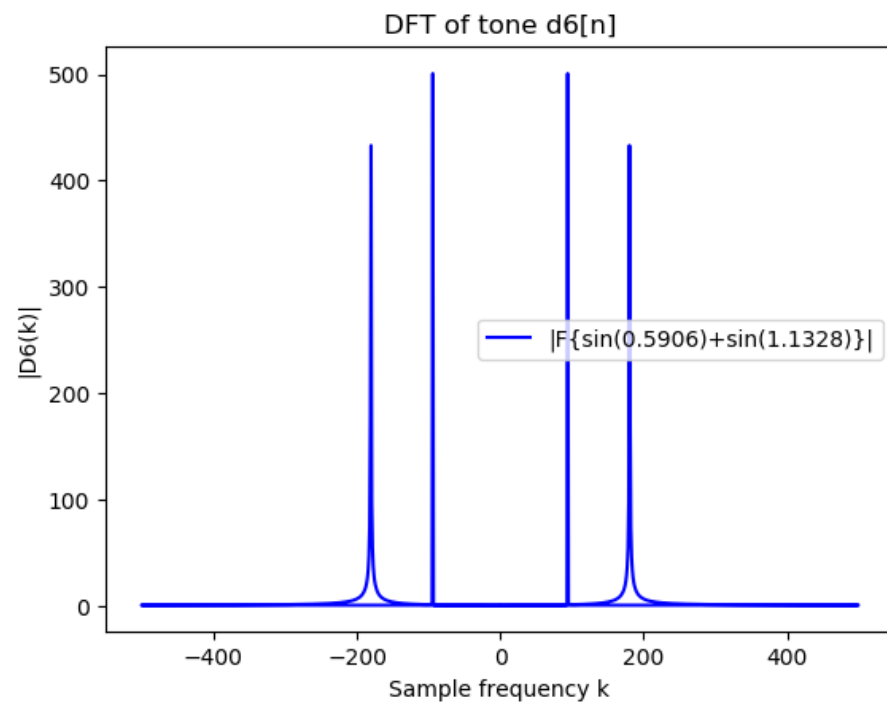
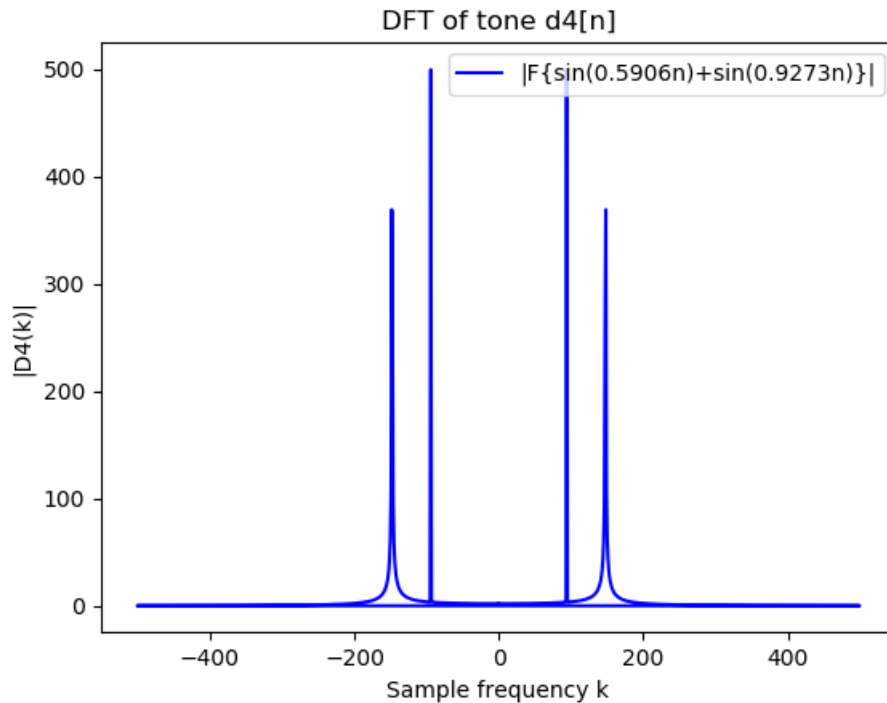
1.2 DFT των σημάτων d4[n] και d6[n]

Ο Διακριτός Μετασχηματισμός Fourier (DFT) $X[k]$ ενός σήματος $x[n]$ δίνεται από τη σχέση:

$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$	Eq. (1.1)
--	-----------

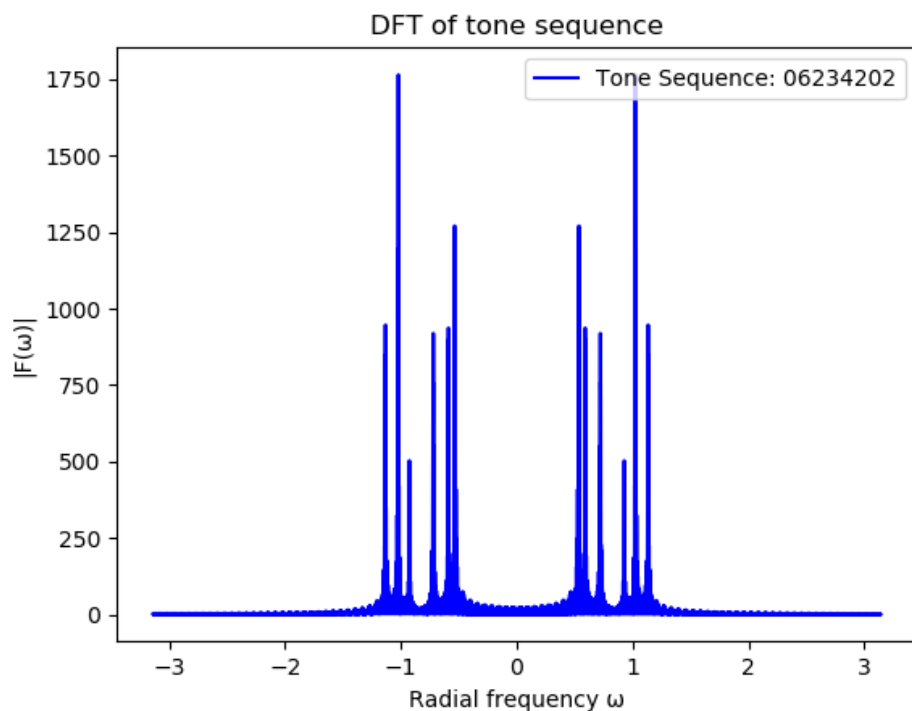
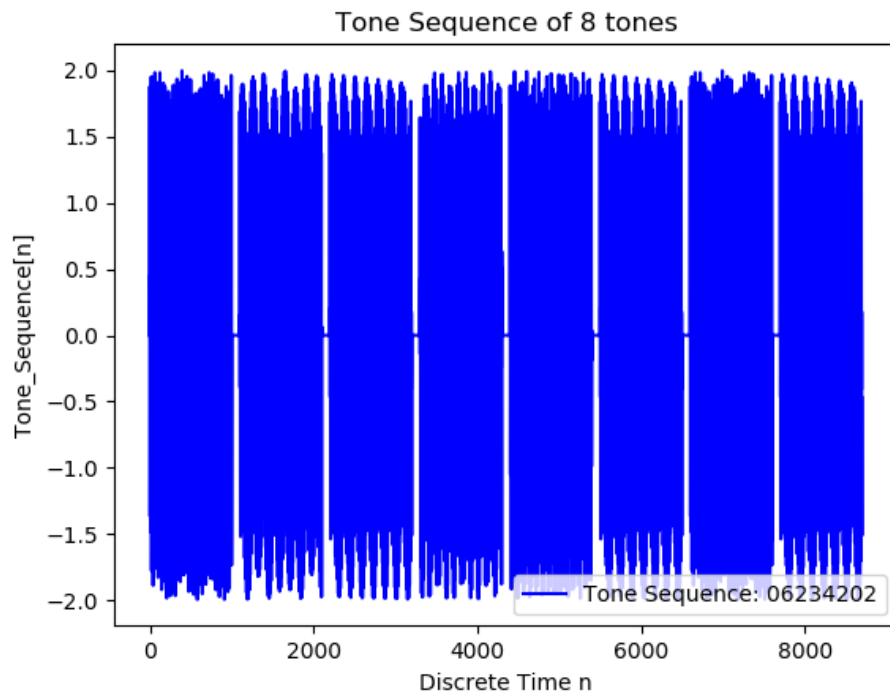
όπου N: αριθμός των δειγμάτων

Κάνοντας χρήση της συνάρτησης `numpy.fft()` της python σχεδιάσαμε τον Διακριτό Μετασχηματισμό Fourier (DFT) των σημάτων d4[n] και d6[n] για N = 1000 δείγματα και στη συνέχεια σχεδιάσαμε τη γραφική παράσταση αυτών. Πιο συγκεκριμένα:



1.3 Δημιουργία ακολουθίας τόνων

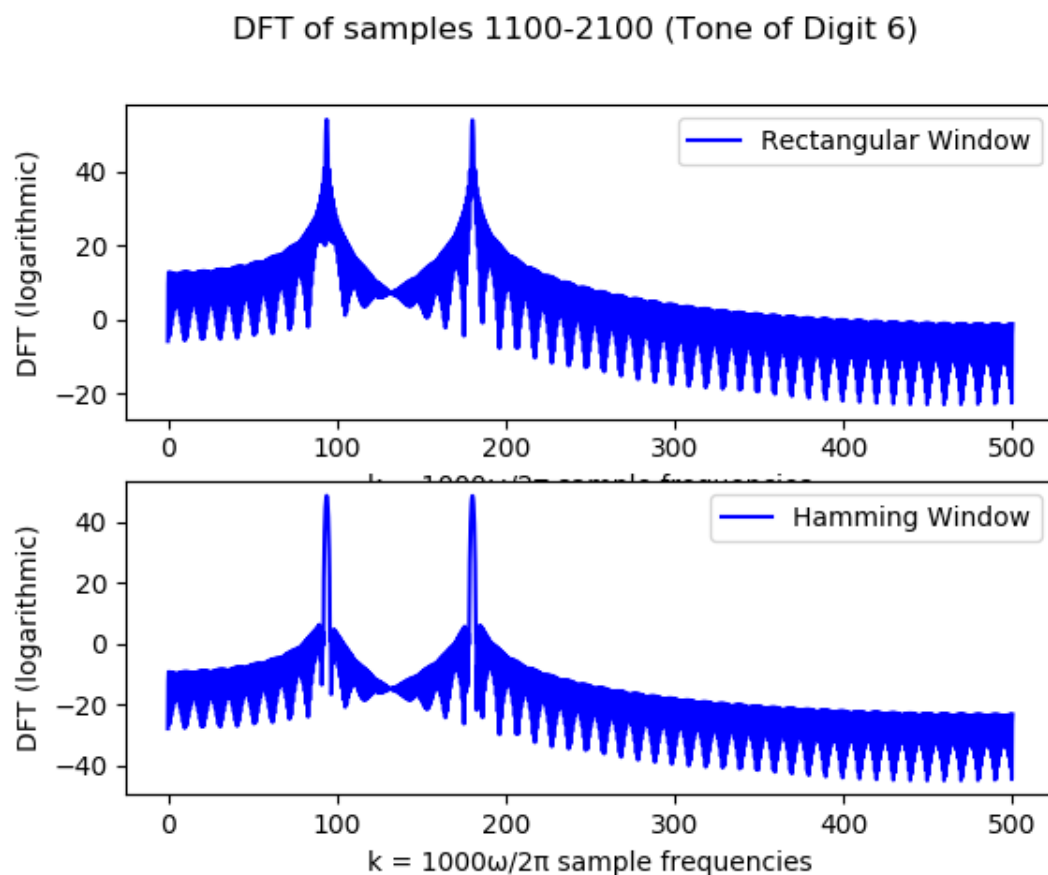
Στο συγκεκριμένο ερώτημα, κληθήκαμε να δημιουργήσουμε και να αποθηκεύσουμε σε αρχείο "tone_sequence.wav", με χρήση της συνάρτησης *write_wav()* της Librosa, ένα σήμα με διαδοχικούς τηλεφωνικούς τόνους. Τα ψηφία του σήματος αυτού προέκυψαν αθροίζοντας τους Αριθμούς Μητρώου μας: $AM1=03117037 + AM2=03117165 = \mathbf{06234202}$. Κάθε ένα ψηφίο της ακολουθίας τόνων διαρκεί 1000 δείγματα και διαχωρίζεται από το προηγούμενο του με 100 μηδενικά δείγματα. Ειδικότερα:



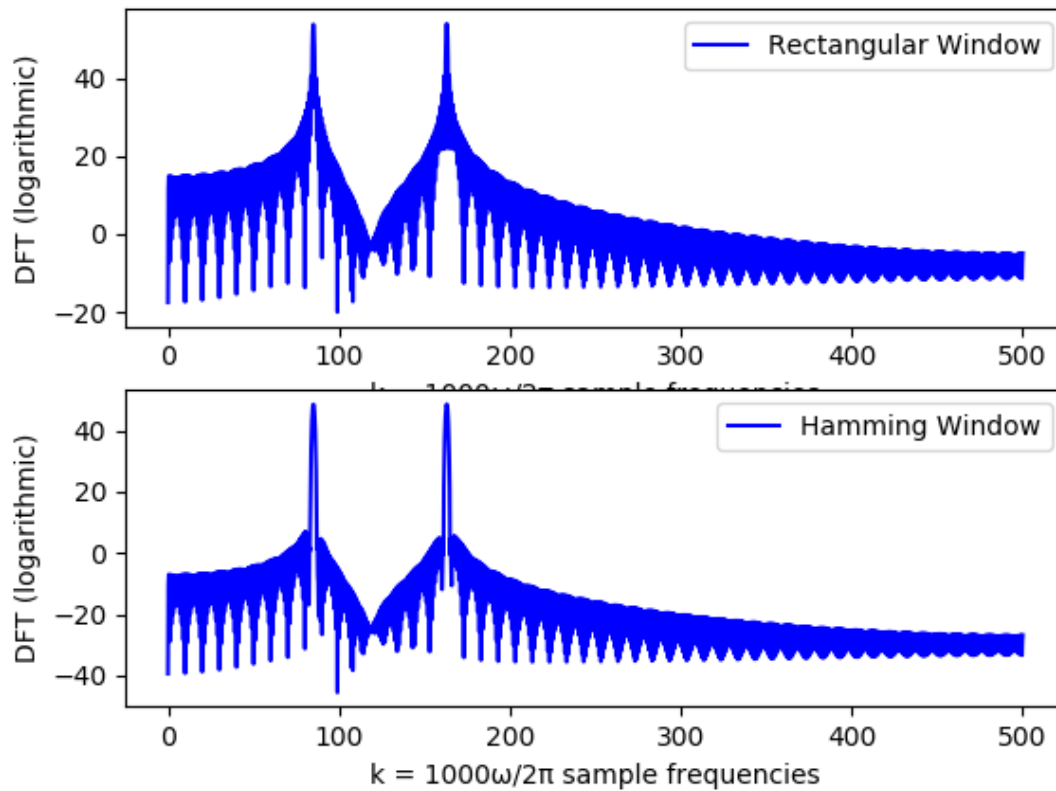
Παρατηρεί κανείς ότι ο Fourier Μετασχηματισμός ολόκληρης της ακολουθίας τόνων καθιστά **δύσκολο** τον ξεχωριστό προσδιορισμό του φασματικού περιεχομένου καθενός εκ των 8 διαφορετικών τόνων. Για τον λόγο αυτό, καθίσταται απαραίτητη η χρήση κατάλληλων παραθύρων.

1.4 Μετασχηματισμός Fourier παραθυροποιημένων σημάτων

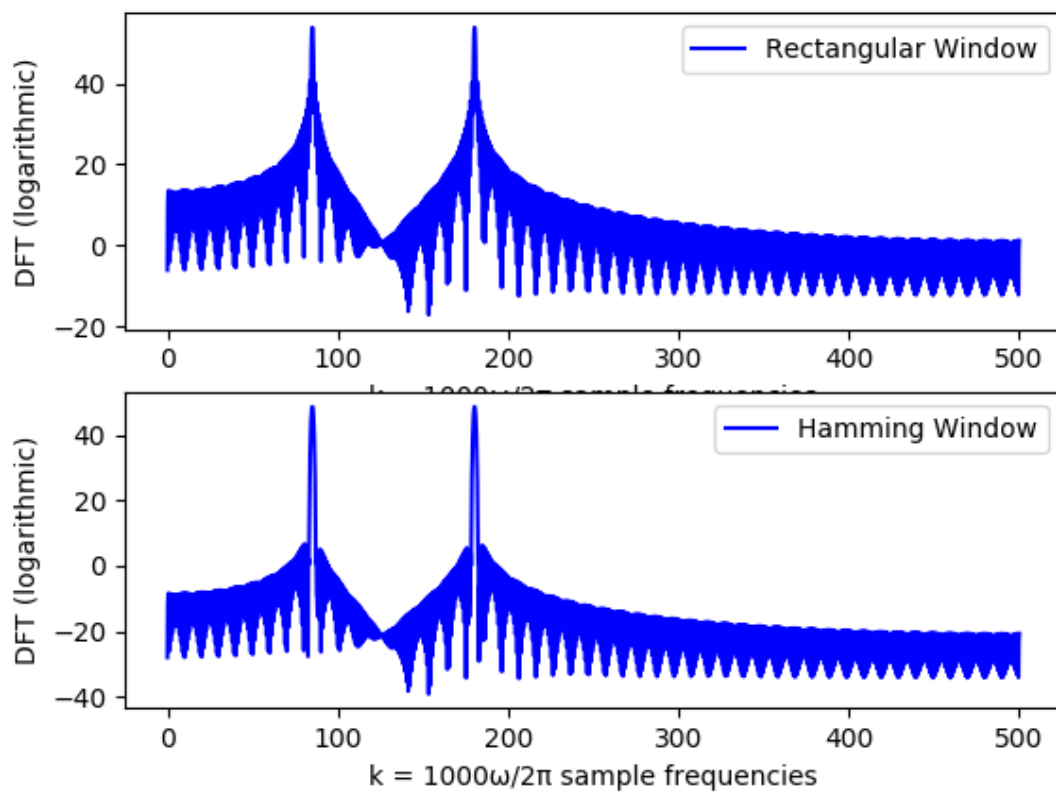
Χρησιμοποιώντας τη συνάρτηση `numpy.fft()` και κατάλληλα χρονικά παράθυρα, (i) **τετραγωνικά** και (ii) **Hamming**, μήκους $N = 1000$ δειγμάτων, υπολογίσαμε τον Διακριτό Μετασχηματισμό Fourier των παραθυροποιημένων σημάτων. Πιο συγκεκριμένα, ολισθαίνουμε στον χρόνο τα προαναφερθέντα παράθυρα, προκειμένου να απομονώσουμε το φασματικό περιεχόμενο των 8 διαφορετικών τόνων που συνθέτουν την ακολουθία τόνων μας. Με αυτόν τον τρόπο μπορούμε να διακρίνουμε με μεγαλύτερη ευκολία το φάσμα του καθενός τόνου ξεχωριστά. Παρακάτω παρατίθενται οι αντίστοιχες γραφικές παραστάσεις σε **λογαριθμική κλίμακα**, ώστε να είναι περισσότερο διακριτή η επίδραση των παραθύρων πάνω στην ακολουθία τόνων. Περιοριζόμαστε μόνο στο διάστημα $(0, 500)$ δείγματα, γνωρίζοντας ότι ο DFT είναι συμμετρικός ως προς το 0 για το διάστημα $(-500, +500)$ των συνολικά 1000 δειγμάτων ενός τόνου. Ειδικότερα:



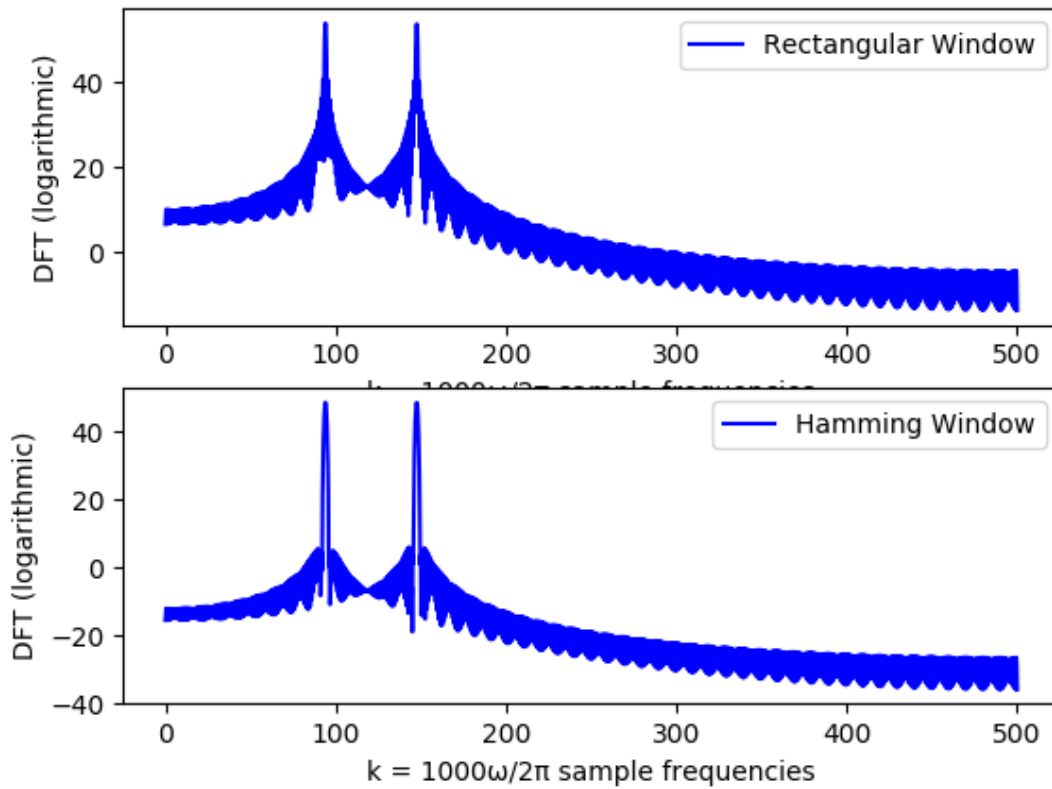
DFT of samples 2200-3200 (Tone of Digit 2)



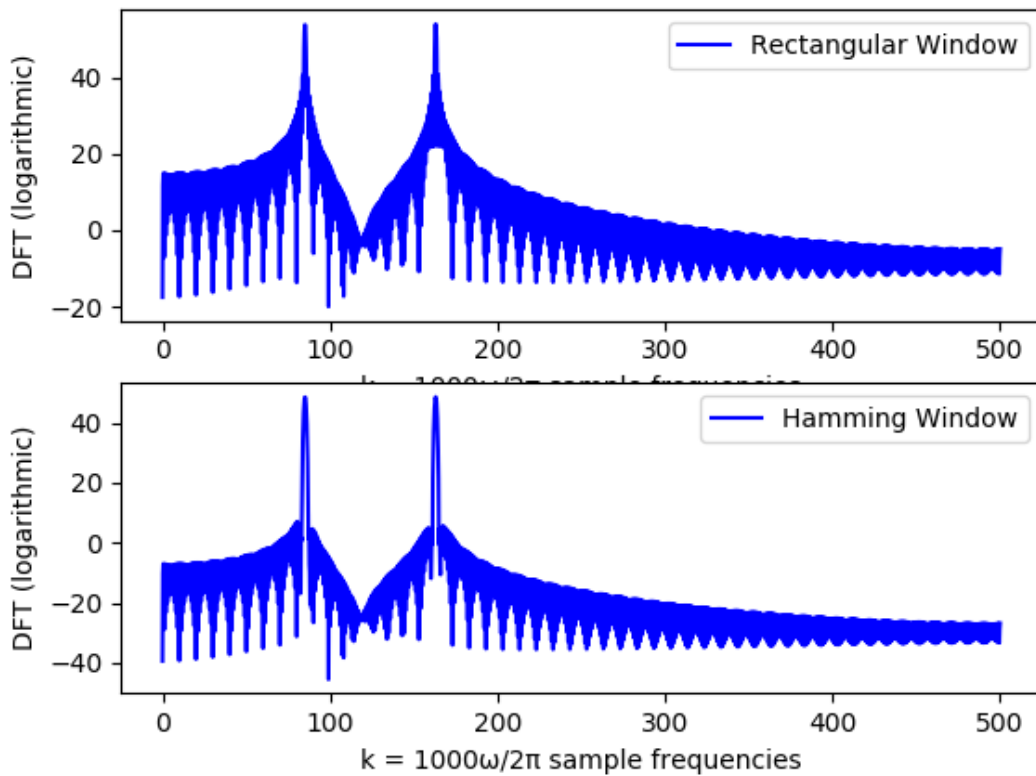
DFT of samples 3300-4300 (Tone of Digit 3)



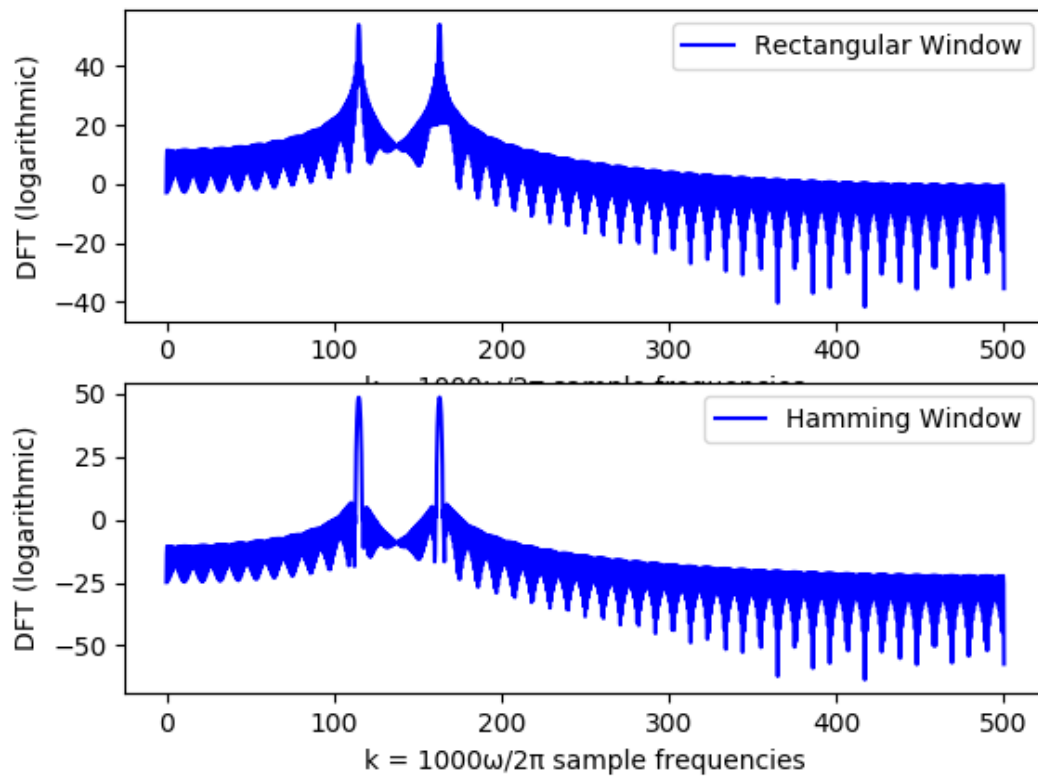
DFT of samples 4400-5400 (Tone of Digit 4)



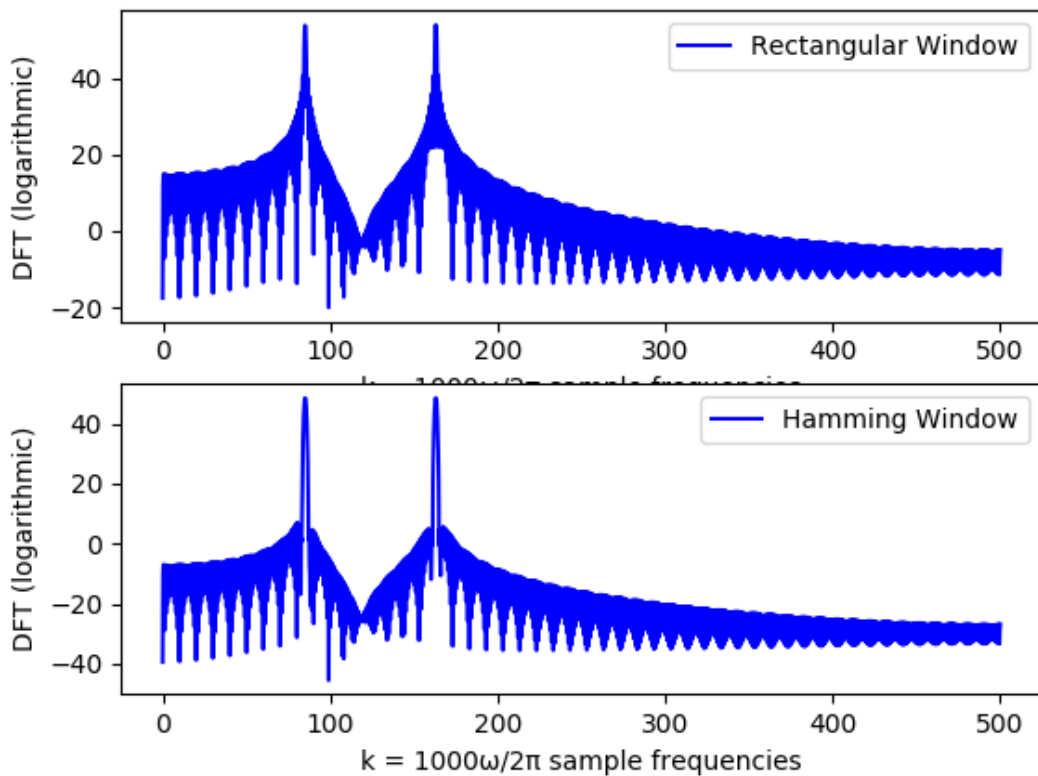
DFT of samples 5500-6500 (Tone of Digit 2)



DFT of samples 6600-7600 (Tone of Digit 0)



DFT of samples 7700-8700 (Tone of Digit 2)



Παρατηρώντας τις παραπάνω γραφικές παραστάσεις, συμπαιρνουμε ότι με χρήση **τετραγωνικών παραθύρων** ο κύριος λοβός του Μετασχηματισμού Fourier των παραθυροποιημένων σημάτων είναι **στενότερος**, ωστόσο οι πλευρικοί λοβοί έχουν **μεγαλύτερο πλάτος**. Αντίθετα, με χρήση **παραθύρων Hamming**, λαμβάνουμε τον Μετασχηματισμού Fourier των παραθυροποιημένων σημάτων με **πιο ευρύ** τον κύριο λοβό, αλλά με **μικρότερο πλάτος** των πλευρικών λοβών.

1.5 Πίνακας συχνοτήτων με δείκτες k

Γνωρίζουμε ότι ο Διακριτός Μετασχηματισμός Fourier (DFT) αποτελεί δειγματοληψία του DTFT για κάθε $\omega = 2k\pi/N$, όπου N ο συνολικός αριθμός δειγμάτων. Στη συγκεκριμένη περίπτωση, καλούμαστε να υπολογίσουμε μια λίστα από δείκτες k για τις αντίστοιχες συχνότητες που βρίσκονται εγγύτερα στις touch-tone συχνότητες, όπως αυτές ορίστηκαν στον πίνακα του ερωτήματος 1.1. Έχοντας, λοιπόν, $N = 1000$ δείγματα για κάθε τόνο κάνουμε χρήση του τύπου:

$k = \omega N / 2\pi$	Eq. (1.2)
-----------------------	-----------

όπου $N = 1000$ ο αριθμός των δειγμάτων

Δεδομένου ότι οι δείκτες k είναι ακέραιοι αριθμοί, χρειάζεται να κάνουμε χρήση της συνάρτησης *numpy.round*, ώστε να λάβουμε τις αντίστοιχες συχνότητες που βρίσκονται **εγγύτερα** στις touch-tone συχνότητες. Ως αποτέλεσμα, παίρνουμε τον παρακάτω πίνακα:

	kcolumn		
krow	148	163	180
85	1	2	3
94	4	5	6
104	7	8	9
115		0	

Δηλαδή:

Values k of rows = [85. 94. 104. 115.]

Values k of columns = [148. 163. 180.]

1.6 Υλοποίηση συνάρτησης ttdecode

Στο συγκεκριμένο ερώτημα καλούμαστε να υλοποιήσουμε μια συνάρτηση με το όνομα `ttdecode`, η οποία θα δέχεται ως όρισμα εισόδου ένα τονικό σήμα, αντίστοιχο με αυτό που δημιουργήσαμε στο Ερώτημα 1.3, και θα επιστρέφει ένα διάνυσμα με τα αντίστοιχα ψηφία. Θέτοντας, λοιπόν, ως είσοδο της συνάρτησης την τονική ακολουθία που δημιουργήσαμε παραπάνω, παίρνουμε ως έξοδο το διάνυσμα $\mathbf{AM1+AM2} = [0.6.2.3.4.2.0.2.]$.

Η λειτουργία της συνάρτησης `ttdecode` περιγράφεται παρακάτω:

- i) Αρχικά, προσδιορίζουμε το μέγεθος της τονικής ακολουθίας εισόδου, καθώς και τη θέση των τόνων, καθένας των 1000 δειγμάτων, μέσα σε αυτήν. Χρειάζεται, μάλιστα, να λάβουμε υπόψη ότι ο αριθμός των μηδενικών μεταξύ δύο διαδοχικών τόνων δεν παραμένει απαραίτητα σταθερός και ίσος με 100, όπως θα φανεί και στο επόμενο ερώτημα.
- ii) Στη συνέχεια, κάνοντας χρήση παραθύρων Hamming των 1000 δειγμάτων, υπολογίζουμε τον Διακριτό Μετασχηματισμό Fourier των παραθυροποιημένων σημάτων που αντιστοιχούν στον κάθε τόνο ξεχωριστά.
- iii) Υπολογίζουμε την ενέργεια του κάθε τόνου των 1000 δειγμάτων κάνοντας χρήση της σχέσης $E_k = |X[k]|^2$, όπου $|X[k]|$ το μέτρο του DFT του τόνου.
- iv) Προσδιορίζουμε τις θέσεις των *sampling frequencies*, έστω k_1 και k_2 , στις οποίες η ενέργεια του τόνου, όπως ορίστηκε παραπάνω, παρουσιάζει μέγιστα.
- v) Τέλος, κάνοντας χρήση του πίνακα δεικτών k που ορίσαμε στο Ερώτημα 1.5, προσδιορίζουμε σε ποιούς δείκτες είναι εγγύτερα οι συχνότητες k_1 και k_2 του προηγούμενου βήματος. Ο συνδυασμός αυτός των δεικτών θα μας οδηγήσει στο ψηφίο που αντιστοιχεί ο κάθε τόνος της ακολουθίας.

1.7 Αποκωδικοποίηση των σημάτων `easySig` και `hardSig`

Στο τελευταίο ερώτημα του 1^{ου} μέρους αρχικά φορτώνουμε τα αρχεία `easySig.npy` και `hardSig.npy` από το συμπληρωματικό υλικό της άσκησης με χρήση της εντολής `numpy.load`. Κάθε ένα από τα αρχεία αυτά θα μας δώσει μια τονική ακολουθία, των 8 τόνων και 10 τόνων αντιστοίχως.

Αξίζει να σημειωθεί ότι η τονική ακολουθία του σήματος `easySig` αποτελείται από 8 τόνους, ο καθένας εκ των οποίων απέχει από τον προηγούμενο του 100 μηδενικά δείγματα, όπως ακριβώς η τονική ακολουθία που ορίσαμε στο ερώτημα 1.3. Αντιθέτως, η τονική ακολουθία του σήματος `hardSig` αποτελείται από 10 τόνους, ο καθένας εκ των οποίων **δεν** απέχει από τον προηγούμενο κάποιον σταθερό αριθμό μηδενικών δειγμάτων.

Θέτοντας ως είσοδο τις δύο τονικές ακολουθίες στη συνάρτηση `ttdecode` που ορίσαμε παραπάνω, προσδιορίζουμε τα ψηφία στα οποία αντιστοιχούν οι τόνοι των δύο σημάτων. Ειδικότερα:

Easy Tone Sequence: [1. 3. 2. 6. 3. 9. 0. 0.]

Hard Tone Sequence: [9. 0. 9. 6. 3. 2. 1. 1. 9. 1.]

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

Σκοπός Άσκησης

Στο δεύτερο μέρος της εργαστηριακής άσκησης, αφού δειγματοληπτήσουμε δύο ξεχωριστά σήματα κάνουμε χρήση των μετασχηματισμών STFT και CWT για την ανάλυση του συχνοτικού τους περιεχομένου. Αναλυτικά, παρατίθενται οι δύο μετασχηματισμοί:

$\text{STFT}(\tau, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n - \tau]e^{-j2\pi kn/M}$	Eq. (2.1)
--	-----------

Όπου M το μήκος του DFT μεγαλύτερο ή ίσο του μήκος L του παραθύρου.

$\text{CWT}(\tau, s) = \frac{1}{\sqrt{ s }} \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-\tau}{s}\right) dt$	Eq. (2.2)
---	-----------

Όπου $\psi(t)$ είναι η βασική-μητρική συνάρτηση, η οποία μπορεί να μετατοπιστεί κατά τ και να μεγεθυνθεί ή σμικρυνθεί κατά s .

Αξίζει να σημειωθεί ότι στον Μετασχηματισμό STFT το μήκος L του παραθύρου w είναι σταθερό και επιλέγεται εξ αρχής, ενώ στον CWT μετασχηματισμό δεν υπάρχει ο περιορισμός αυτός.

2.1 Ανάλυση Συχνοτικού περιεχομένου 1^{ου} σήματος

Δειγματοληπτούμε, με συχνότητα δειγματοληψίας $F_s = 1000 \text{ Hz}$, το σήμα:

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

στο χρονικό διάστημα 0 έως 2 sec και προκύπτει το σήμα διακριτού χρόνου:

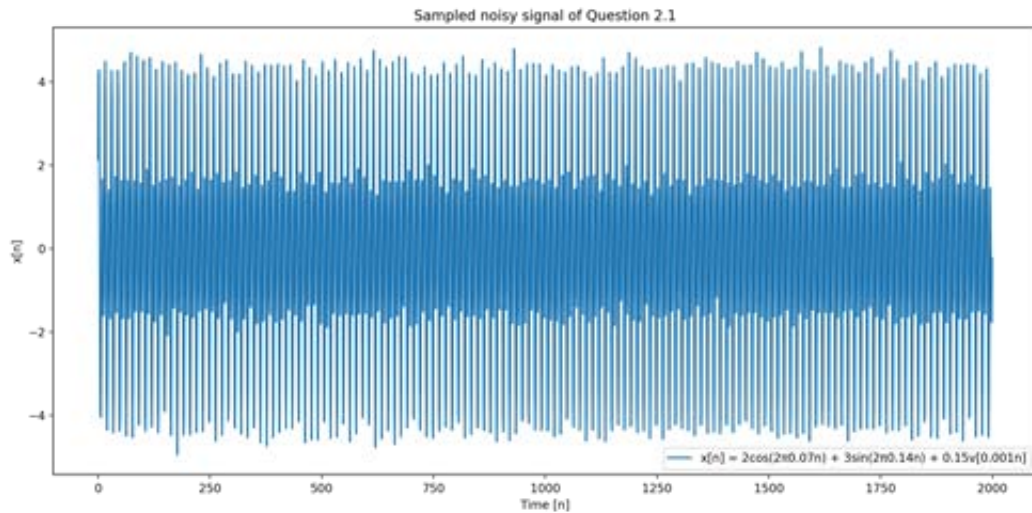
$x[n] = 2\cos[2\pi 0.07n] + 3\sin[2\pi 0.14n] + 0.15u[0.001n]$
--

Εμφανώς, ισχύει: $x(t) = x[nT_s]$, όπου $T_s = 1/F_s = 0.001 \text{ sec}$.

Η χρονική συνάρτηση $u(t)$ – που στη συνέχεια διακριτοποιείται – συμβολίζει λευκό Gaussian θόρυβο μηδενικής μέσης τιμής, τον οποίο και έχουμε υλοποιήσει με χρήση της συνάρτησης `normal()` της `numpy`, με ορίσματα:

- ✓ `loc = 0.0` (μηδενική μέση τιμή)
- ✓ `scale = 1.0` (διασπορά ίση με 1)
- ✓ `size = 2000` ($\frac{2\text{sec}-0\text{sec}}{T_s} = 2000$ δείγματα)

α) Γραφική παράσταση του σήματος $x[n]$



β) STFT Μετ/σμός του σήματος $x[n]$

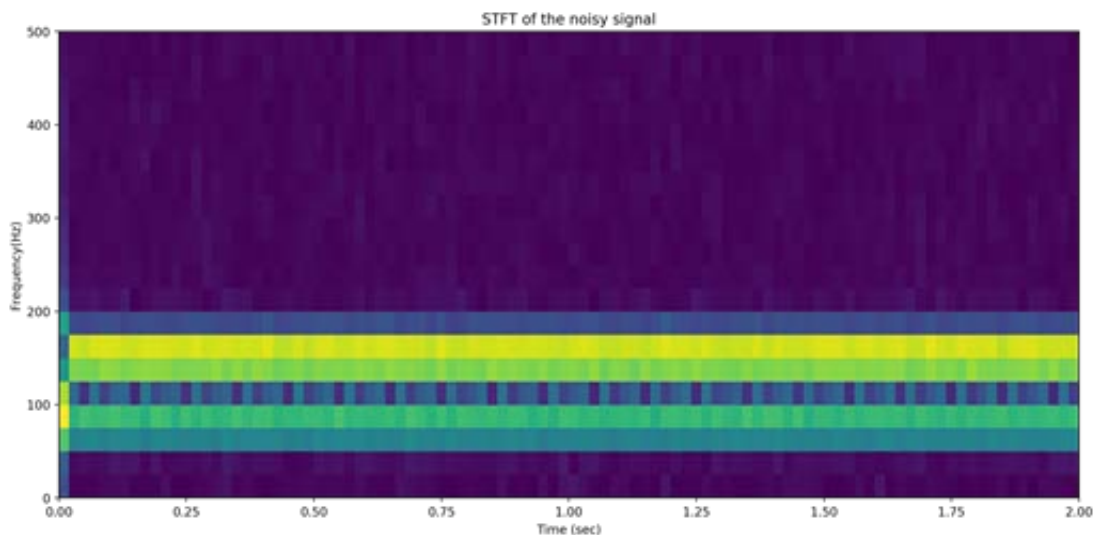
Στη συνέχεια, υπολογίζουμε τον ΜΣ Fourier Βραχέως Χρόνου του σήματος $x[n]$ (με χρήση της ρουτίνας `stft()` της `Librosa`) και τον αποθηκεύουμε σε μια μεταβλητή `G`. Για αυτή την υλοποίηση τέθηκαν οι εξής παράμετροι:

- ✓ Μήκος παραθύρου = 0.04 sec (δηλ. όρισμα $0.04 \times F_s = 40$)
- ✓ Επικάλυψη = 0.02 sec (δηλ. όρισμα $0.02 \times F_s = 20$)

Με αυτά, λοιπόν, τα δεδομένα η συνάρτηση `G.shape` μας επιστρέφει διαστάσεις του προκύπτοντος μιγαδικού πίνακα ίσες με 21×101 .

Αυτές δίνονται ως ορίσματα δειγμάτων στους γραμμικούς χώρους t (χρόνος - 101) και f (συχνότητα - 21).

Εν τέλει, με χρήση της `rcolormesh()` αναπαριστούμε το πλάτος της `G`, δηλαδή του μιγαδικού STFT. Το φασματογράφημα που προκύπτει έχει ως οριζόντιο άξονα τον γραμμικό χώρο t και ως κατακόρυφο τον f . Ειδικότερα:



γ) Discretized CWT του σήματος $x[n]$

Προκειμένου να ορίσουμε τις κλίμακες s , όπως ζητείται, κάνουμε τις εξής κινήσεις. Καταρχάς, δεν είναι δυνατό να οριστούν απ'ευθείας ως ένας γραμμικός χώρος δειγμάτων που ισαπέχουν, καθώς ζητούμε να έχουμε 20 κυματίδια ανά οκτάβα, δηλ. 20 δείγματα κλιμάκων s ανά δύναμη του 2 (οκτάβα: μετάβαση από το $s = 2^k$ στο $s = 2^{k+1}$).

Η συχνότητα συνδέεται με την αντίστοιχη κλίμακα s μέσω της σχέσης:

$f = F_s/s$	Eq. (2.3)
-------------	-----------

Είναι: $F_{\max} = 500 \text{ Hz} = F_s/2$, οπότε λόγω της (1): $s_{\min} = 2^1$

και $F_{\min} = 15.625 \text{ Hz} = F_s/64$, άρα λόγω της (1): $s_{\max} = 2^6$

Επομένως, θα έχουμε $6 - 1 = 5$ οκτάβες.

Άρα, θέλουμε έναν άξονα s , τέτοιο ώστε:

$$s = [2, 2^{21/20}, 2^{22/20}, 2^{23/20}, \dots, 2^2, 2^{41/20}, 2^{42/20}, \dots, 2^6]$$

Δηλαδή:

$$s = 2^{\log_2 [\text{από } 2 \text{ έως } 64 \text{ με } \frac{20 \text{ δειγματα}}{\text{οκτάβα}} \cdot 5 \text{ οκτάβες}]}$$

Δηλαδή:

$$s = 2^{\text{[από } 1 \text{ έως } 6 \text{ με } 100 \text{ δείγματα}]}$$

Οπότε, έχουμε το επιθυμητό αποτέλεσμα με έναν χώρο της μορφής:

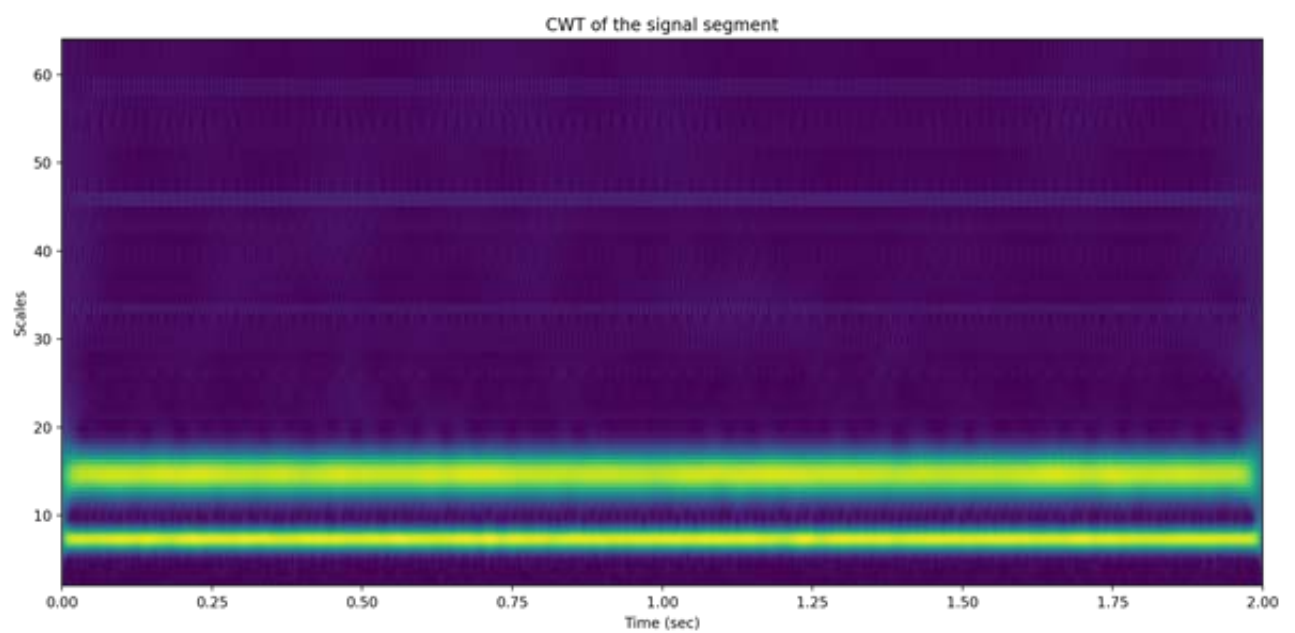
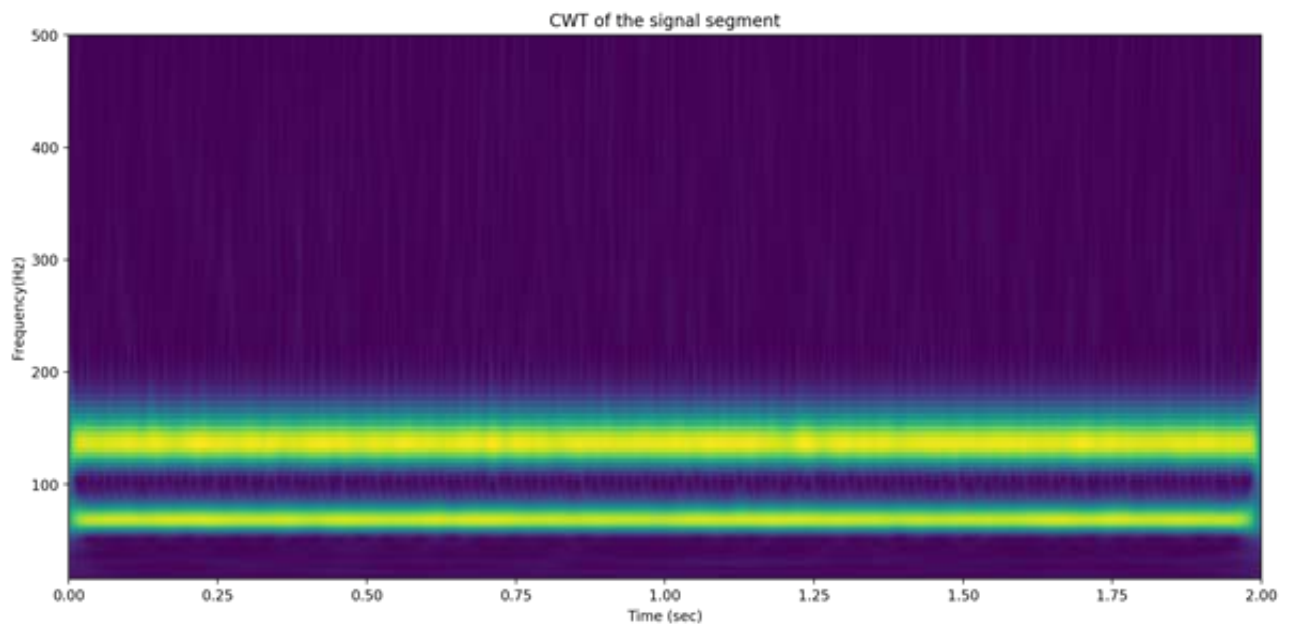
$s = \text{np.power}(2, \text{np.linspace}(1, 6, 100))$

Με χρήση, λοιπόν, της συνάρτησης `cwt()` (εγινε χρήση του ζητούμενου κυματιδίου "cmor3.0-1.0"):

- ✓ αποθηκεύουμε τους επιστρεπτέους συντελεστές πλάτους στη μεταβλητή `coeffs`
- ✓ αποθηκεύουμε τις επιστρεπτέες συχνότητες που αντιστοιχούν στις δοθείσες στη συνάρτηση κλίμακες στη μεταβλητή `freqs`.

Λόγω του τρόπου με τον οποίο κλήθηκε, η συνάρτηση `cwt()` έχει πραγματοποιήσει τους υπολογισμούς θεωρώντας $F_s = 1$. Για να πάρουμε τις σωστές συχνότητες - βάσει της σχέσης (1) – πολλαπλασιάζουμε την `freqs` επί F_s .

Παρακάτω, δίνονται τα προκύπτοντα Scalograms, πρώτα συναρτήσει των συχνοτήτων σε Hz και ύστερα συναρτήσει των κλιμάκων του μετασχηματισμού:



δ) Συμπερασματικές Παρατηρήσεις

Παρατηρούμε στο σημείο αυτό πως το Spectrogram του σήματος (η αναπαράσταση του πλάτους του STFT του), έχει σταθερή ανάλυση, δηλαδή σταθερές διαστάσεις στα χρωματισμένα ορθογώνια από τα οποία αποτελείται. Συγκρίνοντάς το με τα Scalograms (αναπαραστάσεις του πλάτους του Discretized CWT συναρτήσει της συχνότητας και των κλιμάκων), θα λέγαμε ότι τα δεύτερα μας ικανοποιούν περισσότερο, λόγω της μεγαλύτερης ευκρίνειας και συχνοτικής ακρίβειας που πετυχαίνουν. Αναμενόμενο, βέβαια, καθώς ο STFT χρησιμοποιεί παράθυρο σταθερού μήκους, σε αντίθεση με τον ΜΣ Wavelet που χρησιμοποιεί μεταβλητού (κυματίδια με διαφορετικές κλιμακώσεις). Από την άλλη, δε, η σύγκριση μεταξύ των αναπαραστάσεων $|CWT(\tau, f)|$ και $|CWT(\tau, s)|$, φανερώνει μεγαλύτερη ακρίβεια στην αναπαράσταση χαμηλών συχνοτήτων στο σχέδιο του $|CWT(\tau, f)|$ και το αντίστροφο στο σχέδιο του $|CWT(\tau, s)|$, γεγονός συνεφές με την σχέση αντίστροφης αναλογίας (2.3).

2.1 Ανάλυση Συχνοτικού περιεχομένου 2^{ου} σήματος

Δειγματοληπτούμε, με συχνότητα δειγματοληψίας $F_s = 1000 \text{ Hz}$, το σήμα

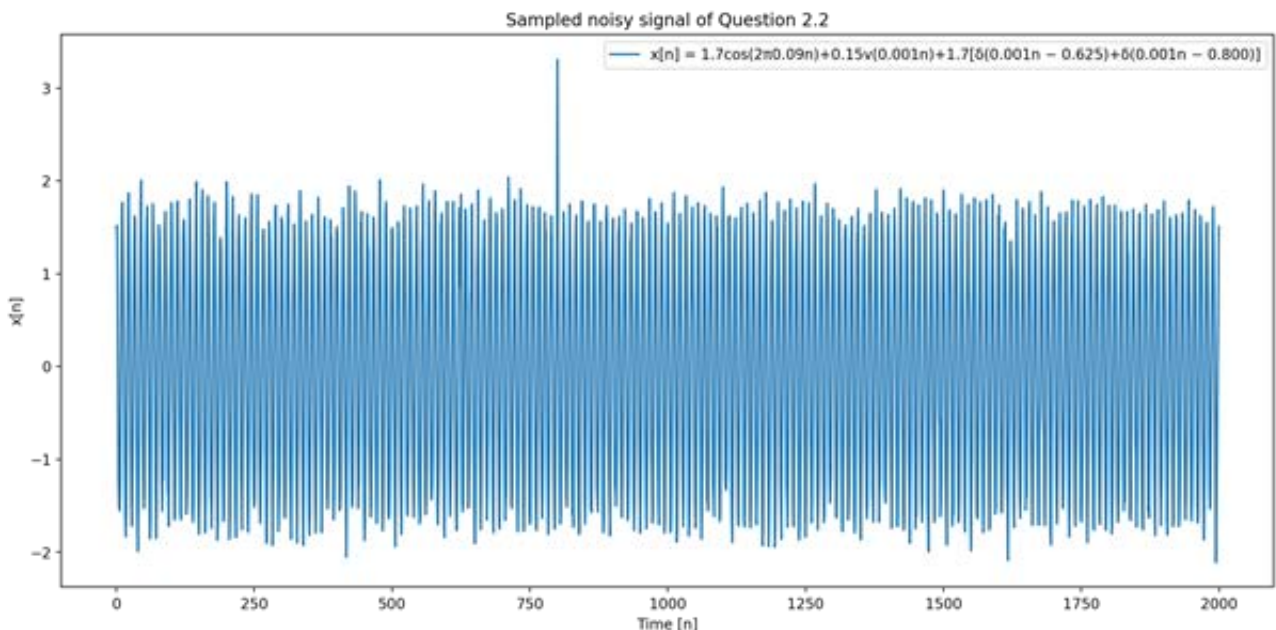
$$x(t) = 1.7\cos(2\pi 90t) + 0.15u(t) + 1.7[\delta(t - 0.625) + \delta(t - 0.800)]$$

στο χρονικό διάστημα 0 έως 2 sec και προκύπτει το σήμα διακριτού χρόνου:

$$x[n] = 1.7\cos(2\pi 0.09n) + 0.15v(0.001n) + 1.7[\delta(0.001n - 0.625) + \delta(0.001n - 0.800)]$$

Εμφανώς, ισχύει: $x(t) = x[nT_s]$, όπου $T_s = 1/F_s = 0.001 \text{ sec}$.

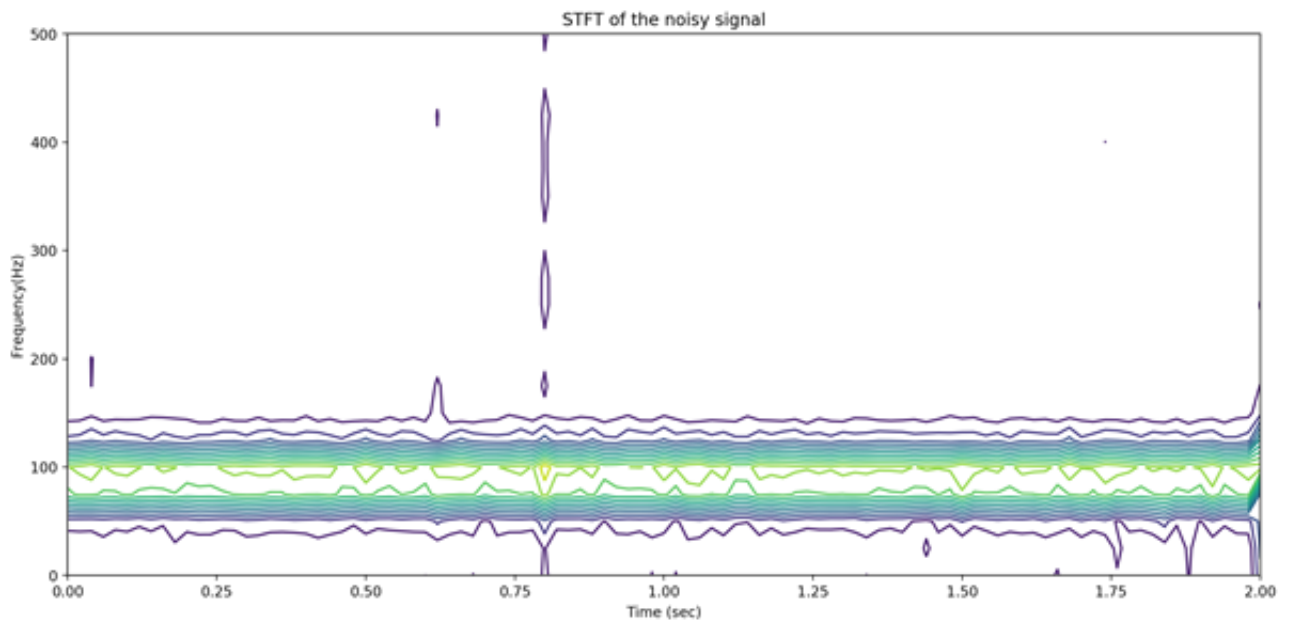
α) Γραφική παράσταση του σήματος $x[n]$



Παρατηρούμε, εδώ, ότι διαφαίνεται ένα spike που αντιστοιχεί στην μια εκ των δύο κρουστικών που υπερτίθενται στο σήμα. Δεν φαίνεται το δεύτερο, καθώς (όπως διαπιστώθηκε ύστερα από δοκιμή με μικρές τροποποιήσεις στον κώδικα) έχει προστεθεί σε αρνητική τιμή πλάτους του υπόλοιπου σήματος.

β) STFT Μετ/σμός του σήματος $x[n]$

Το ζητούμενο ταυτίζεται με αυτό του (β) ερωτήματος του πρώτου σκέλους, απλώς έχουμε διαφορετικό σήμα $x[n]$ και διαφορετική συνάρτηση αναπαράστασης του σχεδίου. Με τη λογική που χρησιμοποιήθηκε προηγουμένως για τη διαμόρφωση των ορισμάτων των συναρτήσεων, υλοποιούμε και τις συναρτήσεις αυτού του σκέλους. Επιπλέον, στην συνάρτηση `contour()` έχουμε θέσει ως όρισμα των `levels` τον αριθμό 16, ώστε να πάρουμε τα επιθυμητά αποτελέσματα. Ειδικότερα:

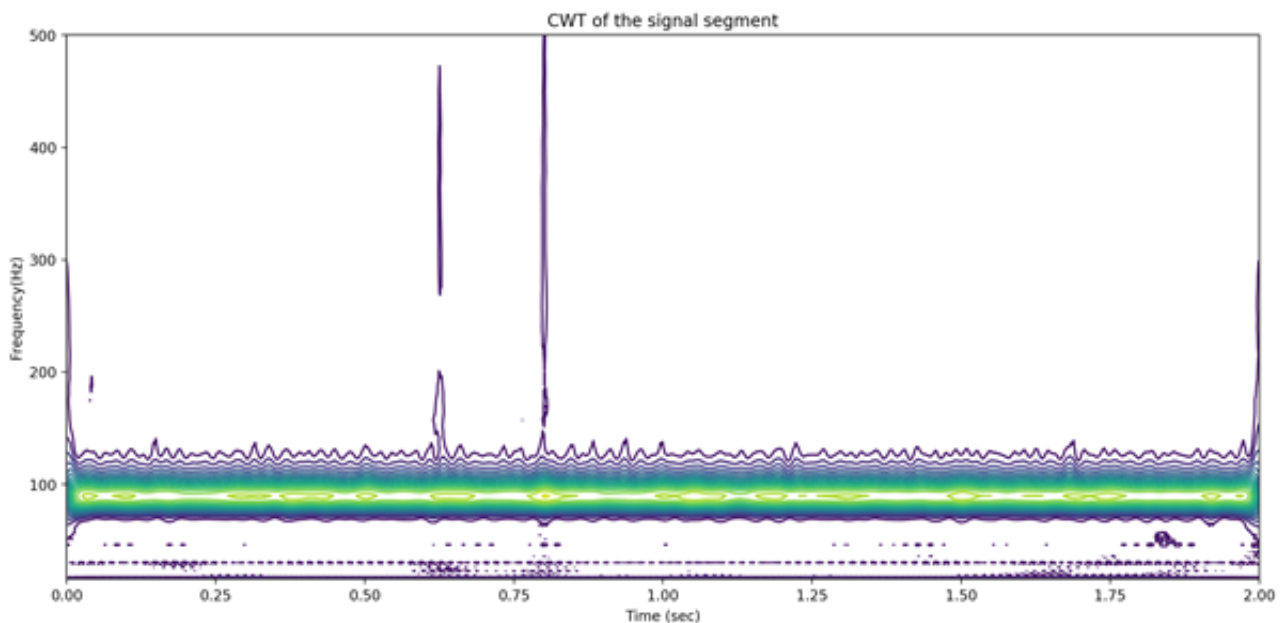


Παρατηρούμε ότι ο θόρυβος φαίνεται στο διάγραμμα, υπό τη μορφή των **κυματισμών** που παρουσιάζουν οι περιφέρειες των χρωματισμένων χωρίων, ενώ οι δύο διαταραχές διαφαίνονται κοντά στις χρονικές στιγμές ορισμού τους και είναι απροσδιόριστες συχνотικά, όπως θα περιμέναμε, άλλωστε, αφού οι κρουστικές στο πεδίο της συχνότητας είναι σταθερές (απροσδιόριστη συχνότητα).

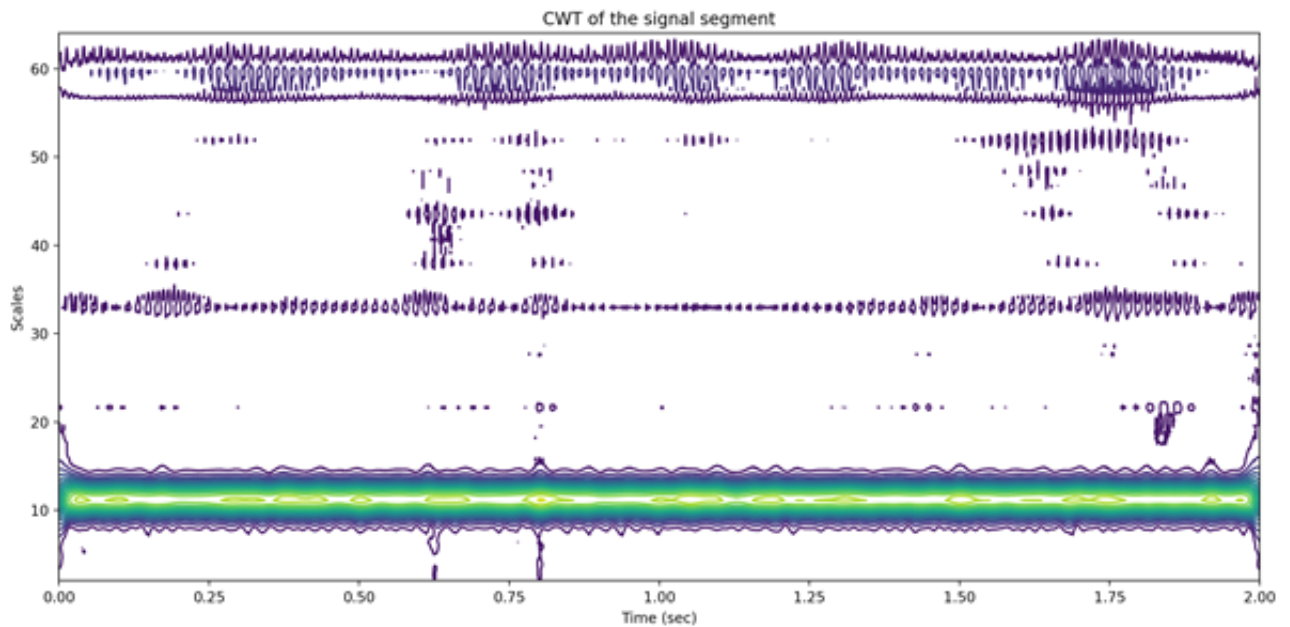
γ) Discretized CWT του σήματος $x[n]$

Το ζητούμενο ταυτίζεται με αυτό του (γ) ερωτήματος του πρώτου σκέλους, απλώς έχουμε διαφορετικό σήμα $x[n]$ και διαφορετική συνάρτηση αναπαράστασης των σχεδίων. Με τη λογική που χρησιμοποιήθηκε προηγουμένως για τη διαμόρφωση των ορισμάτων των συναρτήσεων, υλοποιούμε και τις συναρτήσεις αυτού του σκέλους. Επιπλέον, στην συνάρτηση `contour()` έχουμε θέσει ως όρισμα των `levels` τον αριθμό 16, ώστε να πάρουμε τα επιθυμητά αποτελέσματα. Παρακάτω, παραδίδονται τα Scalograms του σήματος $x[n]$.

Ειδικότερα:



Παρατηρούμε στην αναπαράσταση $|CWT(\tau, f)|$, τη συχνотική αναπαράσταση του θορύβου με τον ίδιο τρόπο με πριν, όπως και τη συχνотική αναπαράσταση των κρουτικών συναρτήσεων. Γνωρίζουμε, δε, ότι τα χαμηλότερα συχνотικά περιεχόμενα των σημάτων θα φαίνονται με υψηλότερη ακρίβεια, μόνο που έχουμε μόνο μια ημιτονοειδή συνάρτηση στον χρόνο, επομένως αυτό το συμπέρασμα **δεν** γίνεται φανερό στο σχέδιο.



Στην αναπαράσταση $|CWT(\tau, s)|$, η αναπαράσταση των κρουστικών **δεν** διαφαίνεται πια. Τα χωρία, πλην του έντονου οριζόντιου πακέτου χωρίων κοντά στα 10 Hz, είναι αποτελέσματα της `contour()` λόγω της υλοποίησης και της ευαισθησίας της και συνεχίζουν να υπάρχουν με ή χωρίς τις κρουστικές και τον θόρυβο.

δ) Συμπερασματικές Παρατηρήσεις

Οι παρατηρήσεις αναφορικά με τη σύγκριση των παραπάνω διαγραμμάτων μεταξύ τους, την ευκρίνεια και τη χωροχρονική ακρίβεια της αναπαράστασης των επιμέρους συμμάτων που υπερτίθενται είναι ανάλογες με αυτές του ερωτήματος (δ) του πρώτου σκέλους.

Μέρος 3^ο – Χαρακτηριστικά Βραχέος Χρόνου Σημάτων Φωνής και Μουσικής (Ενέργεια και Ρυθμός Εναλλαγής Προσήμου)

Σκοπός Άσκησης

Στο 3ο μέρος της εργαστηριακής άσκησης κληθήκαμε μέσω μετρήσεων βραχέος χρόνου, δηλαδή μετρήσεων που γίνονται σε ένα μετακινούμενο παράθυρο, σε σήματα φωνής και μουσικής, να προσδιορίσουμε την ενέργεια και τον ρυθμό εναλλαγής προσήμου.

Πιο συγκεκριμένα, ως **ενέργεια βραχέος χρόνου (short-time energy)** ορίζεται το μέγεθος:

$E_n = \sum_{m=-\infty}^{\infty} [x(m)]^2 w(n-m)$	Eq. (3.1)
---	-----------

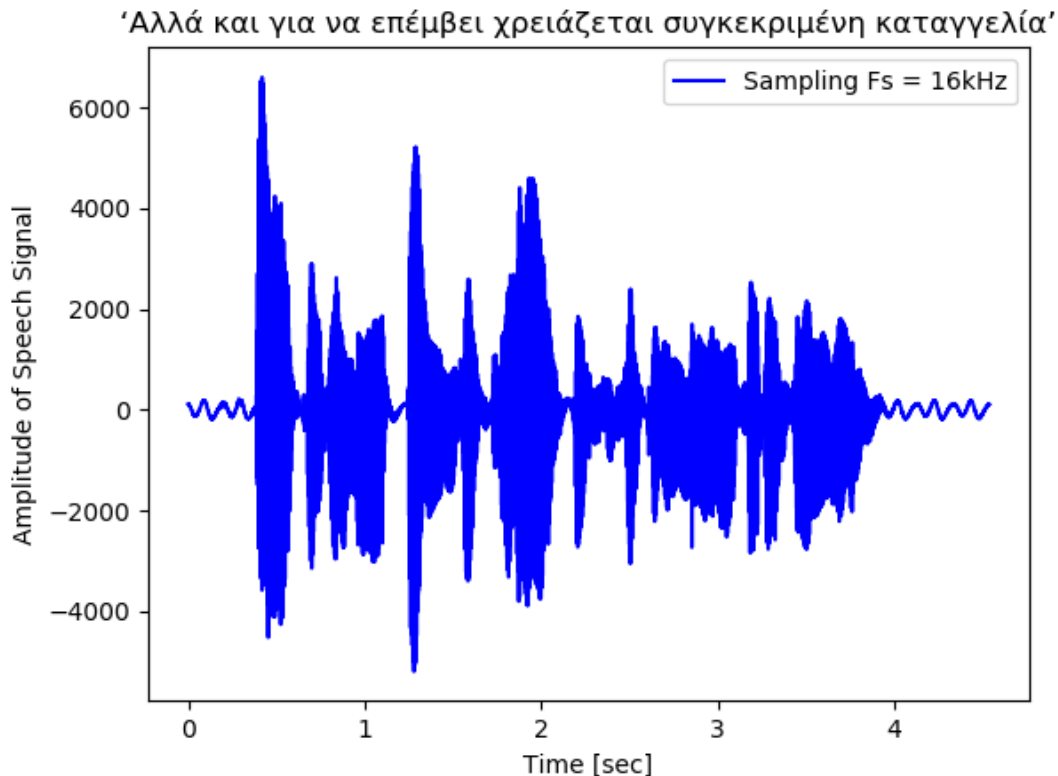
Όπου w ένα παράθυρο της επιλογής μας, συνήθως το Hamming παράθυρο.

Αντιστοίχως, ο **ρυθμός εναλλαγής προσήμου (zero crossing rate)**, για την περίπτωση που χρησιμοποιείται τετραγωνικό παράθυρο N δειγμάτων και πλάτους $1/2N$, ορίζεται ως:

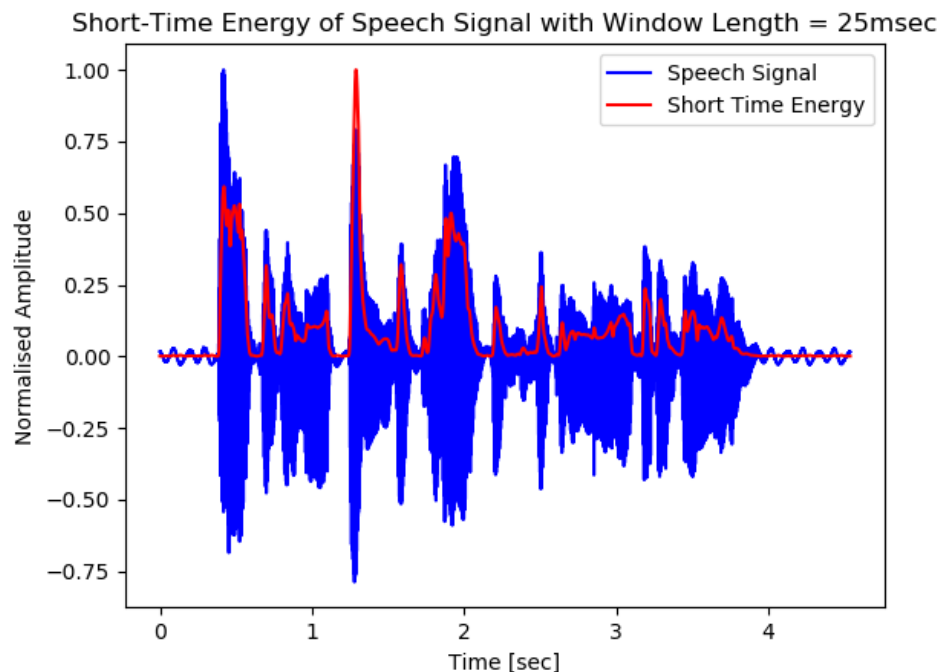
$Z_n = \sum_{m=-\infty}^{\infty} \text{sgn}[x(m)] - \text{sgn}[x(m-1)] w(n-m)$	Eq. (3.2)
--	-----------

3.1 Χαρακτηριστικά Βραχέος Χρόνου Σήματος Φωνής

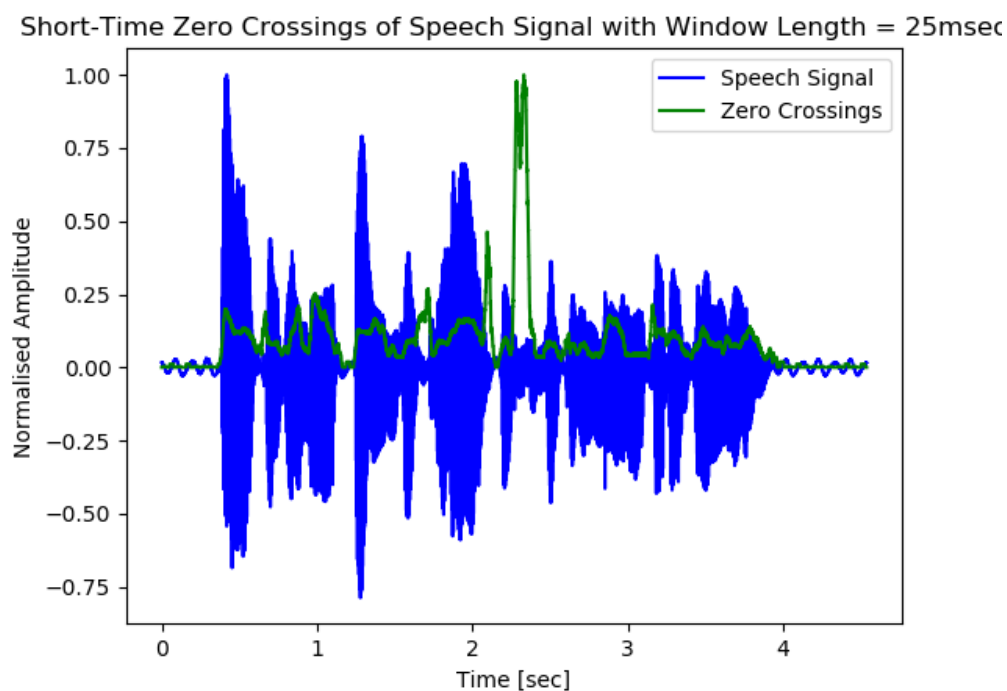
Κάνοντας χρήση της συνάρτησης `wavfile.read` της βιβλιοθήκης `scipy`, ανοίγουμε το αρχείο “speech_utterance.wav” του συμπληρωματικού υλικού, το οποίο περιέχει το σήμα φωνής της πρότασης ‘Αλλά και για να επέμβει χρειάζεται συγκεκριμένη καταγγελία’ με συχνότητα δειγματοληψίας $f_s = 16\text{kHz}$. Παρακάτω φαίνεται η γραφική παράσταση του σήματος φωνής σε συνάρτηση με τον χρόνο:



Στη συνέχεια, κάνοντας χρήση της Εξίσωσης (3.1) υπολογίζουμε την ενέργεια βραχέος χρόνου του εν λόγω σήματος φωνής. Χρησιμοποιούμε **παράθυρο Hamming διάρκειας 25 msec**, δηλαδή $25 \cdot f_s = 400$ **δειγμάτων**. Παρατίθεται σε κοινό διάγραμμα το σήμα φωνής και η ενέργεια βραχέος χρόνου του με **κανονικοποιημένο** τον κατακόρυφο άξονα, προκειμένου να είναι περισσότερο ορατές οι αντίστοιχες περιοχές του σήματος φωνής που αλλάζει η ενέργεια βραχέος χρόνου. Ειδικότερα:

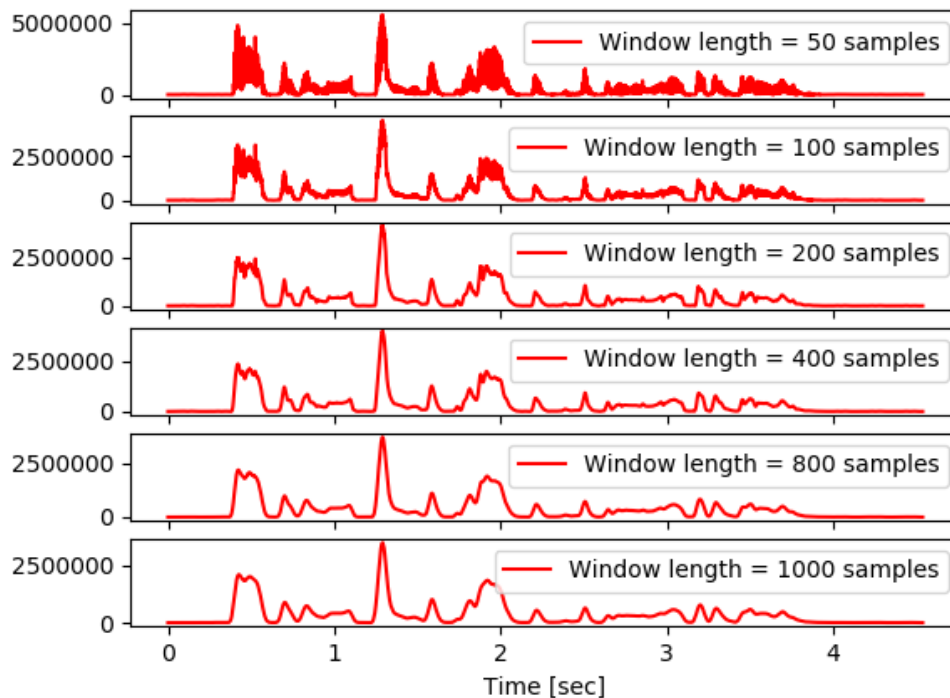


Ομοίως, υπολογίζουμε με χρήση της Εξίσωσης (3.2) τον ρυθμό εναλλαγής προσήμου, χρησιμοποιώντας **τετραγωνικό παράθυρο διάρκειας 25 msec**, δηλαδή $25 \cdot f_s = 400$ **δειγμάτων**. Όπως πριν, παρατίθεται σε κοινό διάγραμμα το σήμα φωνής και ο ρυθμός εναλλαγής προσήμου του με **κανονικοποιημένο** τον κατακόρυφο άξονα:

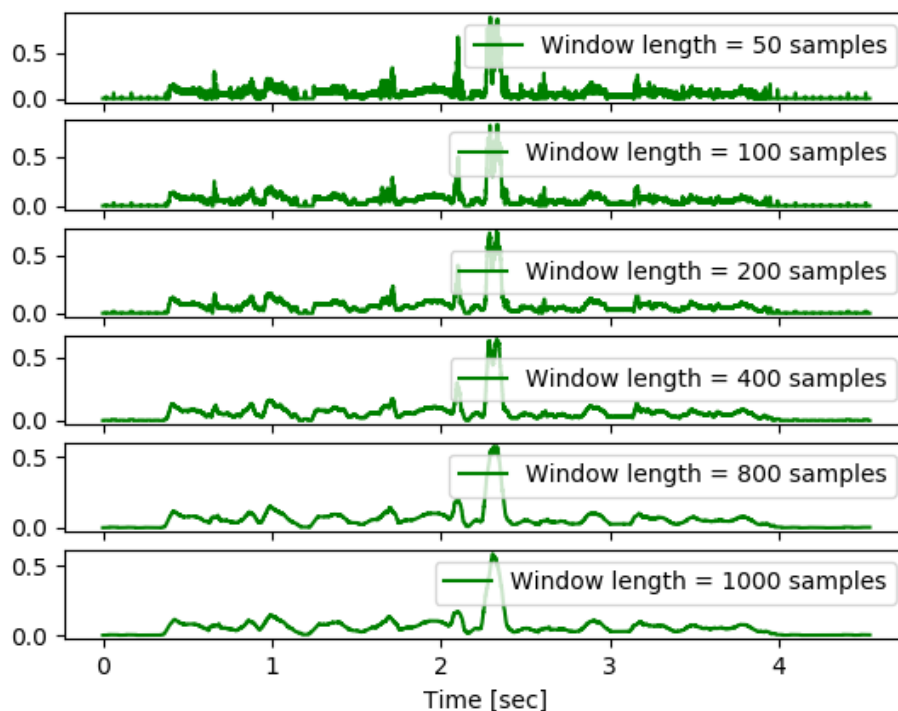


Ακολουθως, αυξανουμε το μήκος των παραθύρων που χρησιμοποιήσαμε, προκειμένου να παρατηρήσουμε τη συμπεριφορά της ενέργειας βραχέος χρόνου και του ρυθμού εναλλαγής προσήμου του σήματος φωνής. Ειδικότερα, χρησιμοποιήσαμε παράθυρα **50, 100, 200, 400, 800 και 1000 δειγμάτων**:

Short-Time Energy of Speech Signal for different Window Lengths



Short-Time Zero Crossings of Speech Signal for different Window Lengths



Όπως παρατηρούμε, λοιπόν, καθώς αυξάνεται το μήκος του χρησιμοποιούμενου παραθύρου, τόσο η ενέργεια βραχέος χρόνου, όσο και ο ρυθμός των zero crossings **εξομαλύνονται**, καθώς παρουσιάζουν λιγότερες κυματώσεις. Συγχρόνως, άυξηση του μήκους παραθύρου επιφέρει **μικρή μείωση του πλάτους** της ενέργειας βραχέος χρόνου, ενώ δεν φαίνεται να επηρεάζει τόσο το πλάτος του ρυθμού εναλλαγής προσήμου. Τέλος, η εξομάλυνση γίνεται πολύ πιο γρήγορα αντιληπτή στα μικρά μήκη παραθύρων, όπως διαφαίνεται και στη μετάβαση από 50 σε 100 δείγματα.

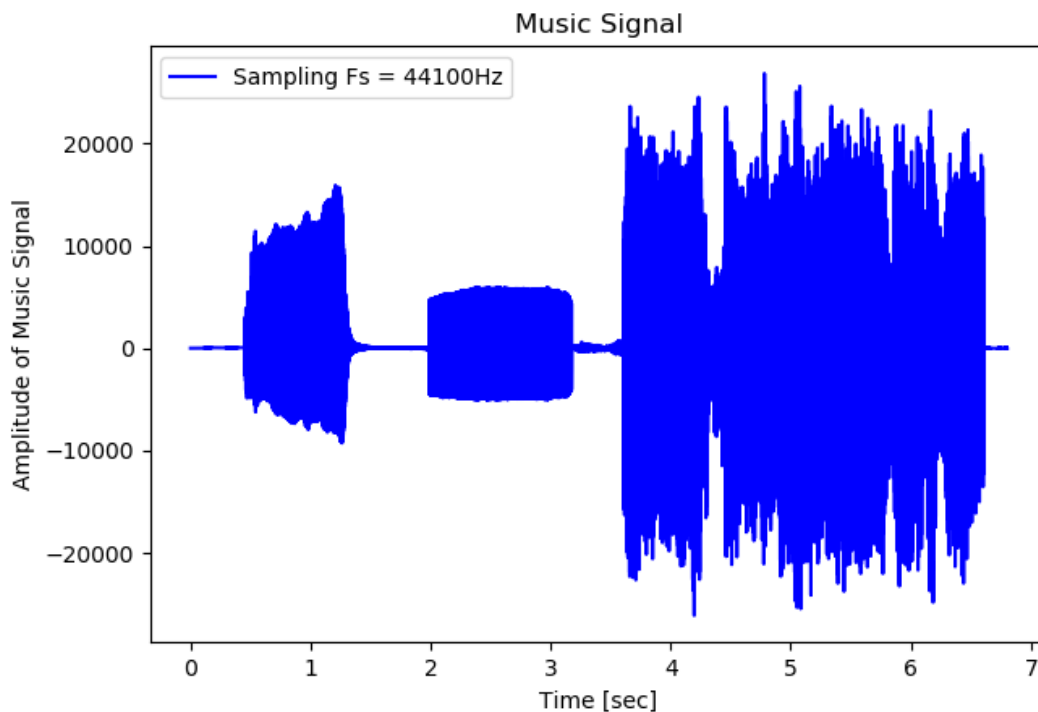
Αξίζει να σημειωθεί ότι ένα μεγάλο σε μήκος παράθυρο θα εξομάλυνε την ενέργεια βραχέος χρόνου και τον ρυθμό εναλλαγής προσήμου σε τέτοιο βαθμό, ώστε αυτά τα μεγέθη να μην παρουσιάζουν αισθητές αλλαγές στις μετρήσεις του. Αντιθέτως, ένα πολύ μικρό παράθυρο – άρα και η πρακτικώς αμελητέα εξομάλυνση που αυτό μας προσφέρει - θα είχε ως αποτέλεσμα την εμφάνιση έντονων κυματώσεων που θα δυσχαίρεναν την ερμηνεία των αποτελεσμάτων. Ως εκ τούτου, χρειάζεται να βρούμε μια **“χρυσή τομή”** όσον αφορά το μήκος του παραθύρου, ώστε να καθίσταται εφικτή η καλύτερη και ποιοτικότερη ερμηνεία των δύο χαρακτηριστικών βραχέος χρόνου. Για τον λόγο αυτό, προτιμάμε παράθυρα μήκους περίπου 400 δειγμάτων, δηλαδή εντός της συνιστάμενης περιοχής των 20-30 msec.

Έπειτα, όσον αφορά τη διάκριση του σήματος φωνής σε περιοχές έμφωνων ήχων, άφωνων ήχων και σιωπής μπορούμε να συμπεράνουμε τα εξής:

- 1) Στις **περιοχές έμφωνων ήχων**, οι οποίες αντιστοιχούν κατά κανόνα σε (τονισμένα) φωνήεντα, η **ενέργεια βραχέος χρόνου λαμβάνει υψηλές τιμές**, σε αντίθεση με τον **ρυθμό εναλλαγής προσήμου ο οποίος είναι σχετικά χαμηλός**. Ενδεικτικές περιοχές του σήματος φωνής μας που αντιστοιχούν σε έμφωνους ήχους εντοπίζονται για περίπου **0.4 sec** (φωνήεν ‘α’ της λέξης “αλλά”), για **1.3 sec** (φωνήεν ‘ε’ της λέξης “επ’Εμβει”) και για **1.9 sec** (φωνήεν ‘α’ της λέξης “χρειάζεται”).
- 2) Στις **περιοχές άφωνων ήχων**, οι οποίες αντιστοιχούν κατά κανόνα σε σύμφωνα του αλφαβήτου, η **ενέργεια βραχέος χρόνου παρουσιάζει χαμηλές αλλά μη μηδενικές τιμές**, ενώ ο **ρυθμός εναλλαγής προσήμου λαμβάνει υψηλές τιμές**. Ενδεικτική περιοχή άφωνου ήχου του σήματος φωνής μας εντοπίζεται περίπου στα **2.3 msec** (σύμωνο ‘ζ’ της λέξης “χρειάζεται”), οπότε και ο ρυθμός εναλλαγής προσήμου του σήματος φωνής λαμβάνει και τη μέγιστη τιμή του.
- 3) Στις **περιοχές σιωπής**, αναμένουμε τόσο η ενέργεια βραχέος χρόνου, όσο και ο ρυθμός εναλλαγής προσήμου να λαμβάνουν **μηδενικές τιμές**. Κάτι τέτοιο διαφαίνεται κυρίως στην αρχή και στο τέλος του σήματος φωνής που μας δόθηκε. Στην πράξη, βέβαια, τα δύο αυτά μεγέθη - κυρίως το zero crossing rate – ενδέχεται να λάβουν μη μηδενικές τιμές λόγω θορύβου του μικροφώνου, κάτι που δεν συμβαίνει στην προκείμενη περίπτωση τόσο έντονα.

3.2 Χαρακτηριστικά Βραχέος Χρόνου Σήματος Μουσικής

Κάνοντας χρήση της συνάρτησης `wavfile.read` της βιβλιοθήκης `scipy`, ανοίγουμε το αρχείο “music.wav” του συμπληρωματικού υλικού, το οποίο περιέχει ένα σήμα μουσικής με συχνότητα δειγματοληψίας $f_s = 44100\text{Hz}$. Παρακάτω φαίνεται η γραφική παράσταση του σήματος μουσικής σε συνάρτηση με τον χρόνο:

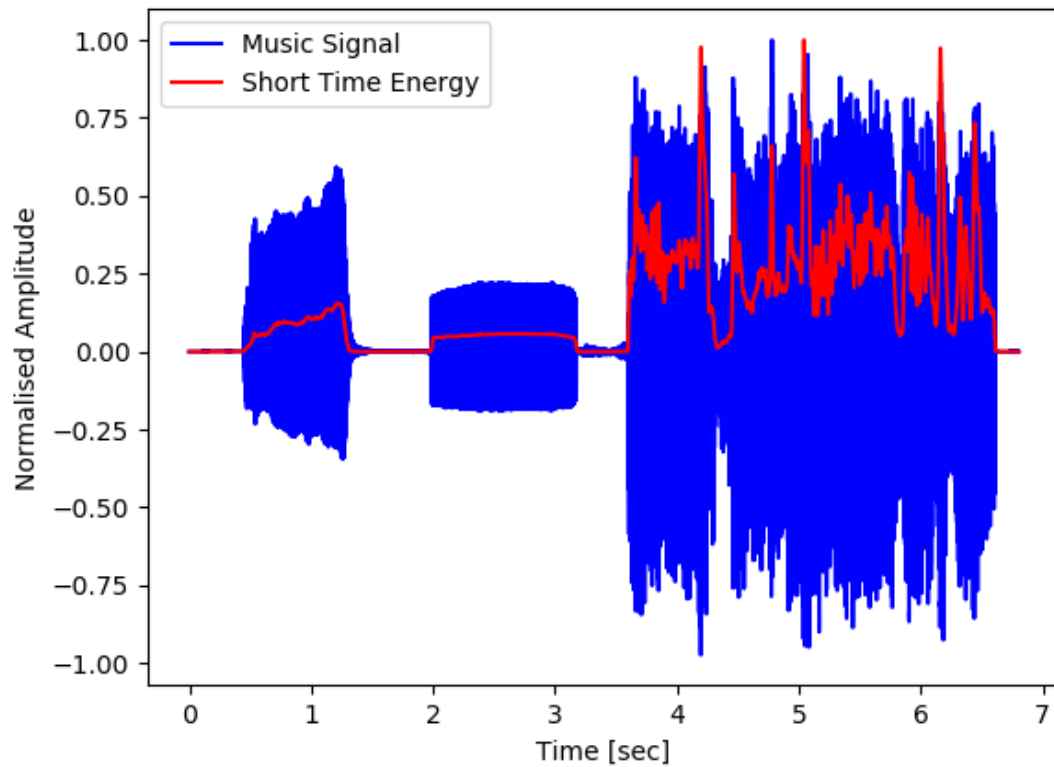


Στη συνέχεια, όπως πριν, κάνοντας χρήση της Εξίσωσης (3.1) υπολογίζουμε την ενέργεια βραχέος χρόνου του εν λόγω σήματος μουσικής. Χρησιμοποιούμε **παράθυρο Hamming διάρκειας 25 msec**, δηλαδή $25 \cdot f_s = 400$ δειγμάτων. Σχεδιάζεται σε κοινό διάγραμμα το σήμα μουσικής και η ενέργεια βραχέος χρόνου του με **κανονικοποιημένο** τον κατακόρυφο άξονα, προκειμένου να είναι περισσότερο ορατές οι αντίστοιχες περιοχές του σήματος φωνής που αλλάζει η ενέργεια βραχέος χρόνου.

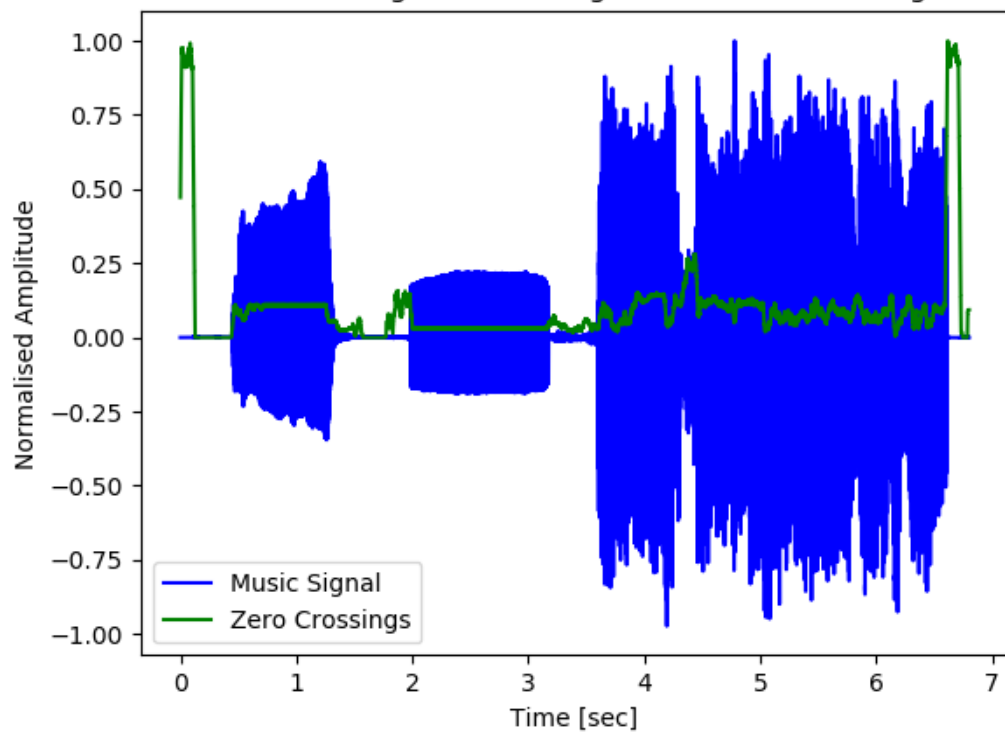
Ομοίως, υπολογίζουμε με χρήση της Εξίσωσης (3.2) τον ρυθμό εναλλαγής προσήμου, χρησιμοποιώντας **τετραγωνικό παράθυρο διάρκειας 25 msec**, δηλαδή $25 \cdot f_s = 400$ δειγμάτων. Όπως πριν, σχεδιάζεται σε κοινό διάγραμμα το σήμα φωνής και ο ρυθμός εναλλαγής προσήμου του με **κανονικοποιημένο** τον κατακόρυφο άξονα.

Παρακάτω παρατίθενται τα δύο διαγράμματα:

Short-Time Energy of Music Signal with Window Length = 25msec

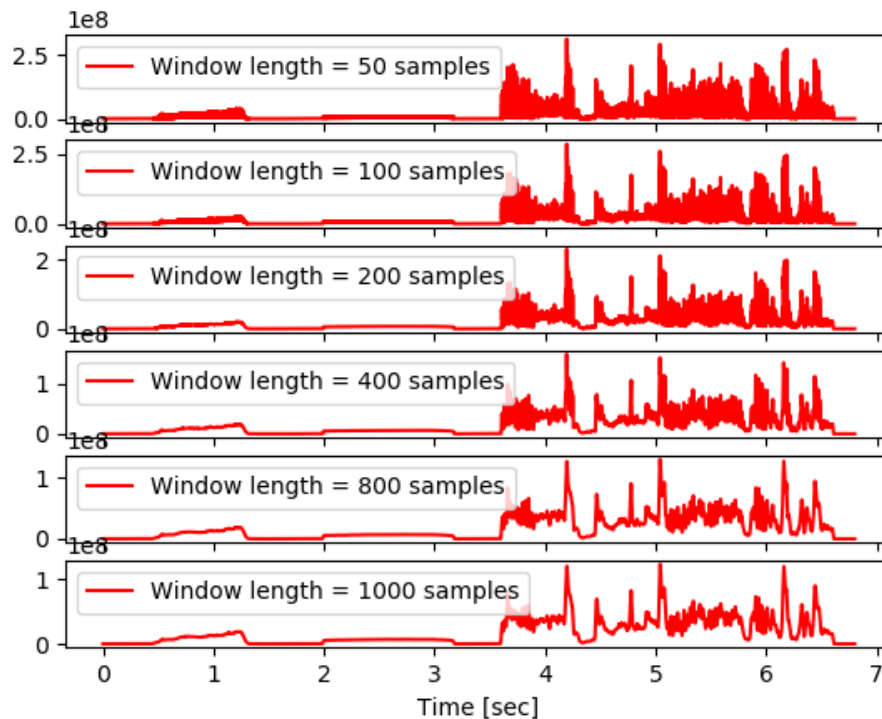


Short-Time Zero Crossings of Music Signal with Window Length = 25msec

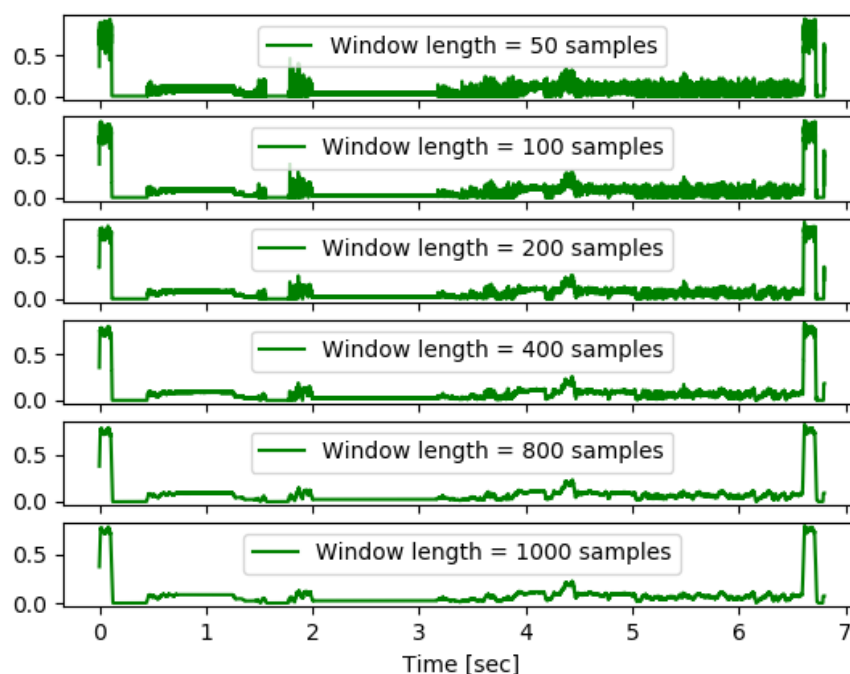


Ακολουθώς, αυξάνουμε το μήκος των παραθύρων που χρησιμοποιήσαμε, προκειμένου να παρατηρήσουμε τη συμπεριφορά της ενέργειας βραχέος χρόνου και του ρυθμού εναλλαγής προσήμου του σήματος φωνής. Ειδικότερα, χρησιμοποιήσαμε παράθυρα **50, 100, 200, 400, 800 και 1000 δειγμάτων**:

Short-Time Energy of Music Signal for different Window Lengths



Short-Time Zero Crossings of Music Signal for different Window Lengths



Εντελώς ανάλογα με το σήμα φωνής του προηγούμενου ερωτήματος, παρατηρούμε ότι καθώς αυξάνεται το μήκος του χρησιμοποιούμενου παραθύρου, τόσο η ενέργεια βραχέος χρόνου, όσο και ο ρυθμός των zero crossings **εξομαλύνονται**, καθώς παρουσιάζουν λιγότερες κυματώσεις. Συγχρόνως, αύξηση του μήκους παραθύρου επιφέρει **μικρή μείωση του πλάτους** της ενέργειας βραχέος χρόνου, ενώ δεν φαίνεται να επηρεάζει τόσο το πλάτος του ρυθμού εναλλαγής προσήμου. Τέλος, η εξομάλυνση γίνεται πολύ πιο γρήγορα αντιληπτή στα μικρά μήκη παραθύρων, όπως διαφαίνεται και στη μετάβαση από 50 σε 100 δείγματα.

Έπειτα, όσον αφορά τη διάκριση του σήματος φωνής σε περιοχές έμφωνων ήχων, άφωνων ήχων και σιωπής μπορούμε να συμπεράνουμε τα εξής:

- 1) Στις **περιοχές σιωπής**, ανάμεσα δηλαδή στα δύο μουσικά όργανα και το απόσπασμα pop κομματιού, παρατηρούμε ότι, όπως και στο σήμα φωνής, τόσο η ενέργεια βραχέος χρόνου, όσο και ο ρυθμός εναλλαγής προσήμου να λαμβάνουν **μηδενικές τιμές**. Αξίζει να σημειωθεί, ωστόσο, ότι σε ορισμένα σημεία το zero crossings rate λαμβάνει πολύ **υψηλές μη μηδενικές τιμές**, κάτι που όπως αναφέρθηκε οφείλεται στον **θόρυβο** του μικροφώνου.
- 2) Στις περιοχές άφωνων ήχων, οι οποίες αντιστοιχούν κατά κανόνα σε **καθαρές νότες**, η **ενέργεια βραχέος χρόνου παρουσιάζει χαμηλές αλλά μη μηδενικές τιμές**. Ενδεικτική περιοχή είναι η καθαρή νότα που εντοπίζεται περίπου στα **2 sec**, οπότεν και η ενέργεια βραχέος χρόνου λαμβάνει σταθερή τιμή χαμηλή ως προς το πλάτος. Στην ίδια περιοχή παρατηρούμε ότι ο ρυθμός εναλλαγής προσήμου λαμβάνει επίσης πάρα πολύ χαμηλή τιμή.
- 3) Στις περιοχές έμφωνων ήχων, οι οποίες αντιστοιχούν κατά κανόνα σε **συνδυασμό από πολλές νότες και beats**, η **ενέργεια βραχέος χρόνου παρουσιάζει πολύ υψηλές τιμές**. Ενδεικτική περιοχή είναι αυτή που εμφανίζεται το απόσπασμα pop κομματιού περίπου στα **3,8 sec**, καθώς και η πρώτη περιοχή ήχου που αντιστοιχεί σε άγριο παίξιμο βιολιού(?) περίπου στα **0.5 sec**, οπότεν και η ενέργεια βραχέος χρόνου λαμβάνει πολύ υψηλές τιμές ως προς το πλάτος, ενώ ο ρυθμός εναλλαγής προσήμου λαμβάνει επίσης σχετικά μεγαλύτερες τιμές.

Συνεπώς, μπορούμε να συμπεράνουμε ότι τα δύο χαρακτηριστικά μεγέθη βραχέος χρόνου μας βοηθάνε να ξεχωρίσουμε ένα σήμα μουσικής καλύτερα σε περιοχές σιωπής, καθώς και σε περιοχές όπου οι νότες παίζονται περισσότερο ή λιγότερο καθαρά και με μεγαλύτερη ή μικρότερη ένταση.

Μέρος 4^ο – Πολυκλιμακωτή Ανάλυση Σημάτων μέσω του Διακριτού Μετασχηματισμού Wavelets για τη Ρυθμική Ανάλυση Τραγουδιών

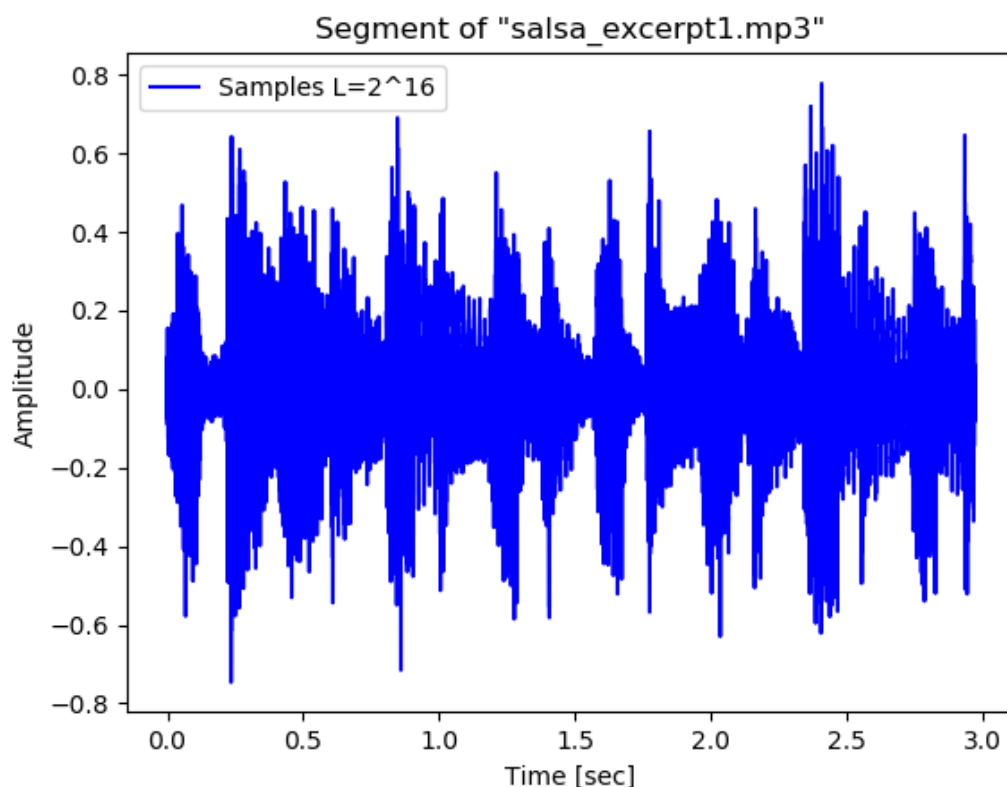
Σκοπός Άσκησης

Το τελευταίο μέρος της εργαστηριακής άσκησης αποσκοπεί στην μελέτη των μουσικών σημάτων και πιο συγκεκριμένα τις δύο κυριότερες προσεγγιστικά περιοδικές συμπεριφορές που αυτά εμφανίζουν σε μια πληθώρα χρονικών κλιμάκων. Αυτές είναι η περιοδικότητα σε επίπεδο νότας (όπου η συχνότητα μετράται σε Hertz), και η ρυθμική ομοιότητα, που αφορά στη διαδοχή παλμών στο μουσικό σήμα, η οποία μετράται σε παλμούς ανά λεπτό (BPM). Για την ανάλυση αυτών των συμπεριφορών, κάνουμε χρήση του Διακριτού Μετασχηματισμού Wavelet (DWT), ο οποίος αναλύει το φασματικό περιεχόμενο του εκάστοτε μουσικού σήματος σε πολλαπλές κλίμακες, των τιμών 2^k , $k=1, \dots, N$, για N επίπεδα ανάλυσης.

Πολυκλιμακωτή Ανάλυση μουσικού σήματος Salsa 1

4.1 Απομόνωση αποσπάσματος μουσικού σήματος:

Κάνοντας χρήση της εντολής *librosa.load()* φορτώνουμε στον υπολογιστή το αρχείο “salsa_excerpt1.mp3” από το συμπληρωματικό υλικό της άσκησης, με συχνότητα δειγματοληψίας $f_s = 22.05\text{kHz}$. Στη συνέχεια, απομονώνουμε ένα απόσπασμα μήκους $L=2^{16}$ δειγμάτων (περίπου 3 δευτερολέπτων) από αυτό, το οποίο αντιστοιχεί στα δείγματα **10000:75536**. Παρακάτω διαφαίνεται η γραφική παράσταση του αποσπάσματος σε συνάρτηση με τον χρόνο:



Παρατηρούμε ότι το απόσπασμα του μουσικού σήματος παρουσιάζει peaks κάθε περίπου 0.2sec. Συνεπώς, εκτιμάμε πως η παλμική περιοδικότητα του αποσπάσματος είναι κατά προσέγγιση **5Hz**.

4.2 Πολυκλιμακωτή ανάλυση με χρήση DWT

Με τη βοήθεια της εντολής `pywt.wavedec()` εφαρμόζουμε στο άνωθι απόσπασμα τον Διακριτό Μετασχηματισμό Wavelet, ώστε να εξάγουμε την πολυκλιμακωτή του ανάλυση σε συνιστώσες. Χρησιμοποιούμε **κυματίδια Daubechies-4** και συνολικά **7 επίπεδα ανάλυσης**. Ως αποτέλεσμα, λαμβάνουμε 7 υψίσυχνες συνιστώσες λεπτομέρειας (detail) του αποσπάσματος, $y_{di}[n]$, $i = 1, \dots, 7$, και μία χαμηλόσυχη συνιστώσα προσέγγισης (approximation) $y_{a7}[n]$. Οι παραπάνω συνιστώσες έχουν μέγεθος δυνάμεις του 2, το οποίο ολοένα και μειώνεται καθώς περνάμε σε μεγαλύτερα επίπεδα ανάλυσης. Επιπλέον, προκειμένου να διατηρήσουμε τα σήματα στην ίδια κλίμακα μετά την εφαρμογή του DWT, διαιρούμε καθεμία εκ των συνιστωσών με τον παράγοντα $\sqrt{2}$.

4.3 Εξαγωγή Περιβάλλουσας Συνιστωσών

Επιχειρούμε να εξάγουμε την περιβάλλουσα κάθε μιας εκ των συνιστωσών ακολουθώντας τα εξής βήματα:

1. Υπολογίζουμε την απόλυτη τιμή του σήματος, $z_i[n] = |y_i[n]|$ με χρήση της εντολής `numpy.abs`
2. Φιλτράρουμε το προκύπτον σήμα με χρήση βαθυπερατού φίλτρου, ώστε:

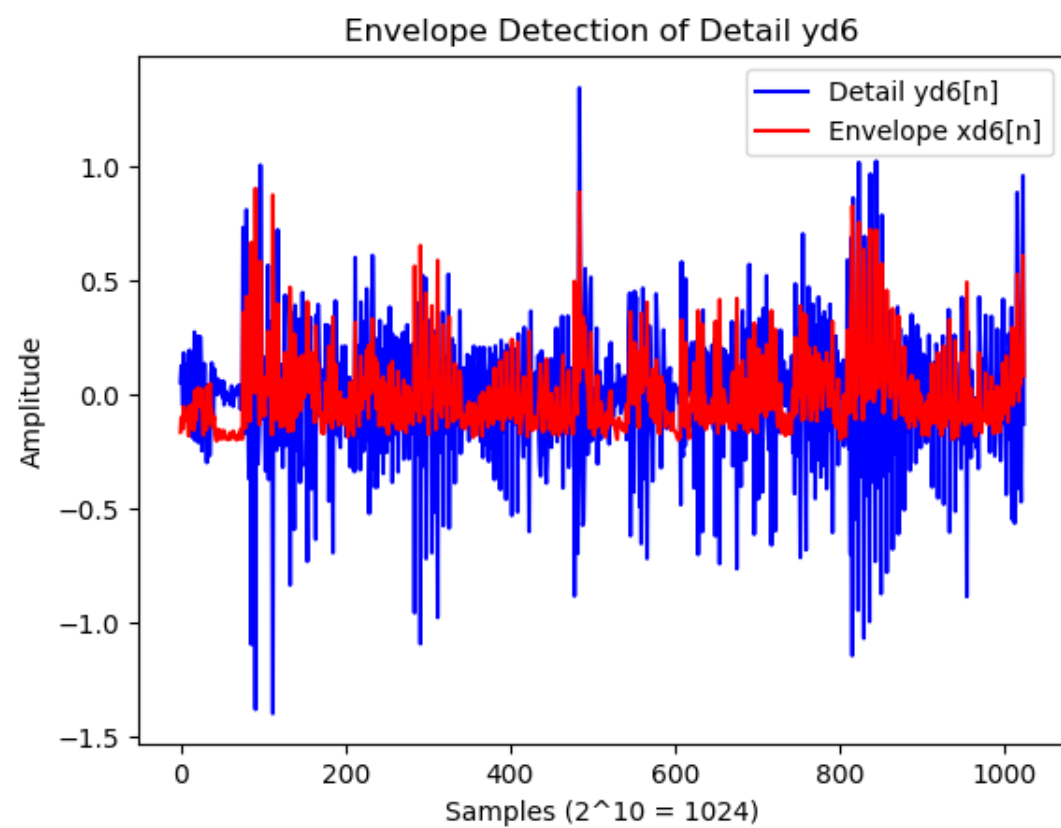
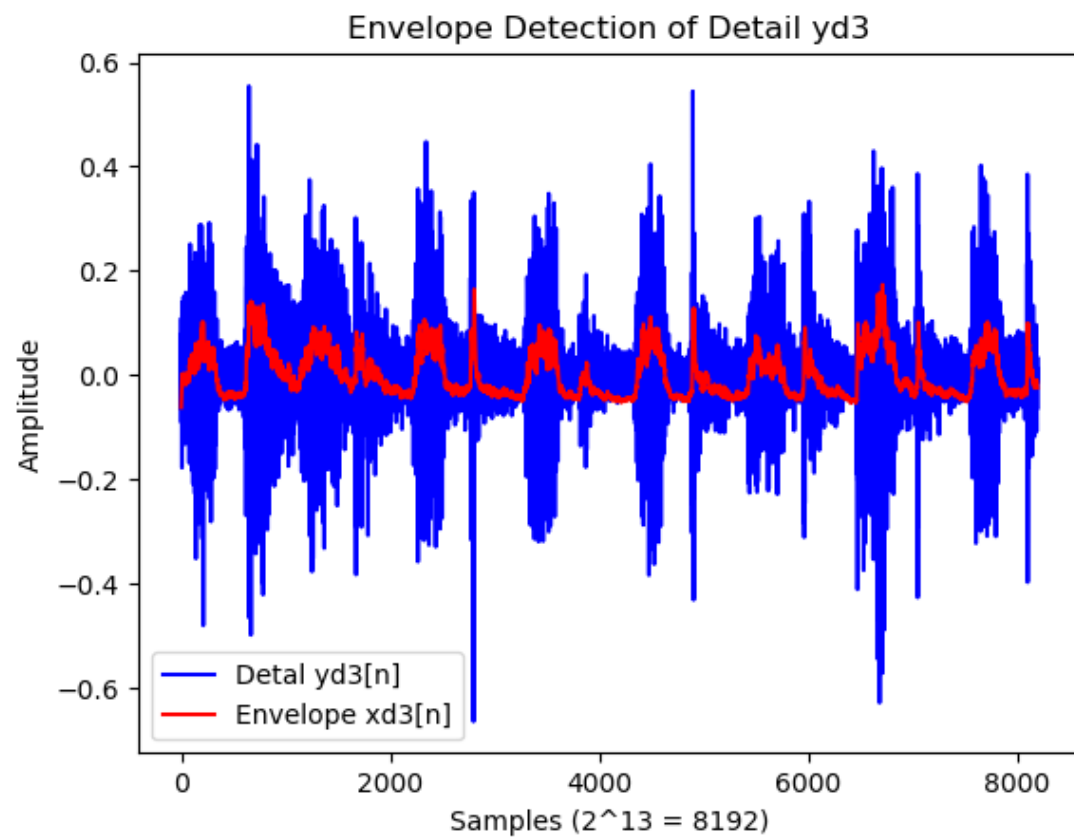
$$x_i[n] = (1 - \alpha)x_i[n-1] + \alpha z_i[n]$$

όπου $\alpha = \alpha_0 2^k$, k : το επίπεδο ανάλυσης που βρισκόμαστε. Η επιλογή αυτή μας εξασφαλίζει ομοιομορφία στη χρονική κλίμακα του βαθυπερατού φίλτρου. Στην προκειμένη περίπτωση επιλέγουμε $\alpha_0 = \mathbf{0.006}$.
3. Κεντράρουμε το προκύπτον σήμα $x_i[n]$, αφαιρώντας από αυτό τη μέση τιμή του με χρήση της εντολής `numpy.mean`.

Ακολουθώντας την παραπάνω διαδικασία, εξάγουμε τις περιβάλλουσες $x_{di}[n]$, $i = 1, \dots, 7$ για κάθε μία από τις 7 συνιστώσες λεπτομέρειας, καθώς και την περιβάλλουσα $x_{a7}[n]$ της συνιστώσας προσέγγισης.

Παρατίθενται, ενδεικτικά, οι γραφικές παραστάσεις των σημάτων $y_{d3}[n]$, $y_{d6}[n]$, μαζί με τις περιβάλλουσες τους $x_{d3}[n]$, $x_{d6}[n]$, σε συνάρτηση με τον αριθμό δειγμάτων τους.

Ειδικότερα:

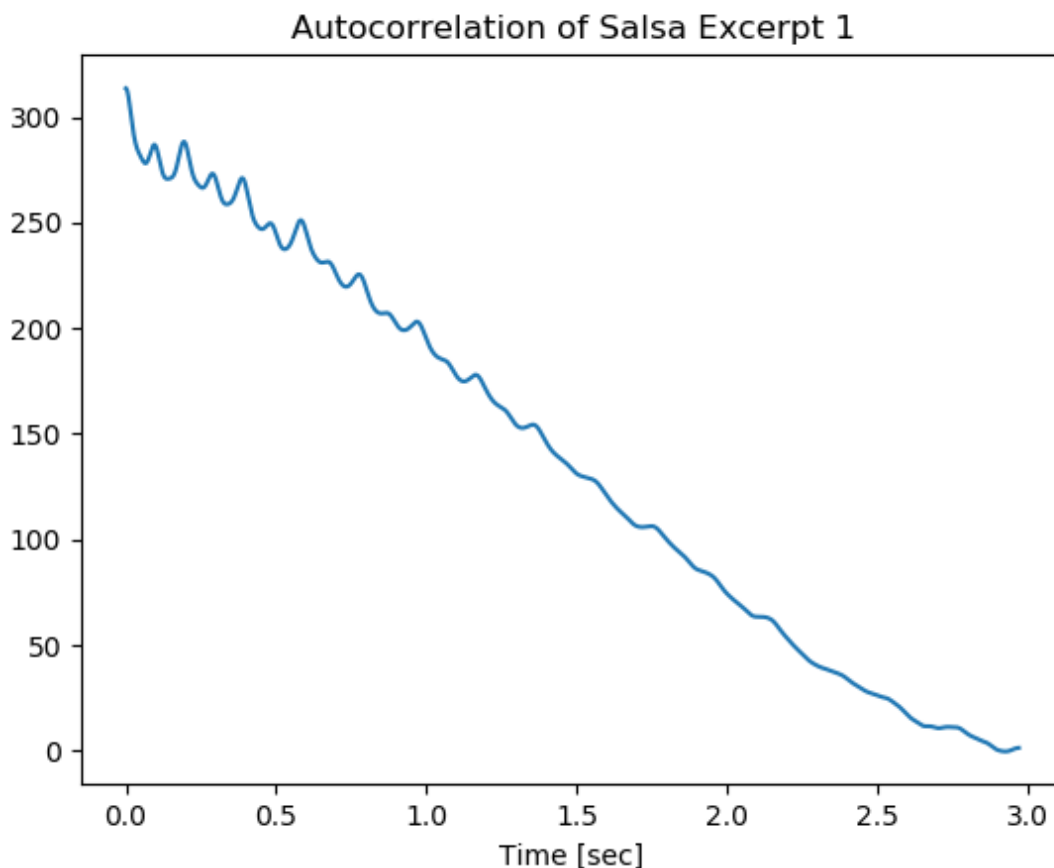


4.4 Άθροιση περιβάλλουσών με γραμμική παρεμβολή

Προκειμένου οι περιβάλλουσες των συνιστωσών να αποκτήσουν το ίδιο μήκος, εφαρμόζουμε **γραμμική παρεμβολή** με χρήση της εντολής `numpy.interp()`, αναγάγοντας κάθε περιβάλλουσα σε μήκος 32768 δειγμάτων. Στη συνέχεια, αθροίζουμε όλες τις ιδίου μήκους πλέον μήκους περιβάλλουσες και παίρνουμε το σήμα $x_{sum}[n]$.

Ακολούθως, με χρήση της εντολής `numpy.correlate()`, υπολογίζουμε την αυτοσυσχέτιση του σήματος $x_{sum}[n]$. Παρατηρούμε, ωστόσο, ότι η συνάρτηση αυτοσυσχέτισης παρουσιάζει **τραχύτητες**, γι' αυτό και την εξομαλύνουμε, εφαρμόζοντας σε αυτήν Γκαουσιανό φίλτρο διασποράς $\sigma^2 = 150$, με χρήση της εντολής `scipy.ndimage.gaussian.filter1d()`.

Παρακάτω φαίνεται σε γράφημα το δεξί συμμετρικό μέρος της αυτοσυσχέτισης του σήματος $x_{sum}[n]$ σε συνάρτηση με τον χρόνο:



Σημείωση:

Η μετατροπή του οριζόντιου άξονα από αριθμό δειγμάτων σε χρόνο έγινε λαμβάνοντας υπόψη ότι η συχνότητα δειγματοληψίας του σήματος είναι ίση με $f_s/2 = 11.025\text{kHz}$, καθώς το σήμα μας έχει πλέον το μισό αριθμό δειγμάτων από ότι το αρχικό απόσπασμα.

4.5 Ρυθμός (BPM) του μουσικού αποσπάσματος

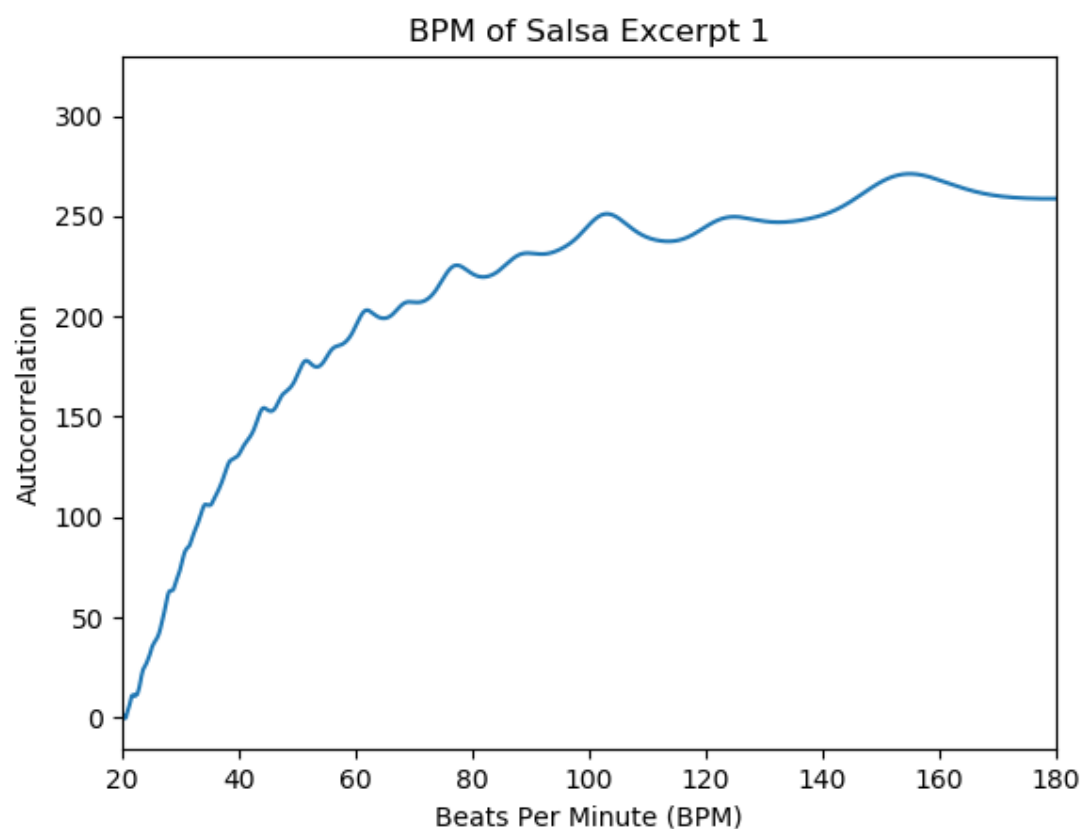
Με τη βοήθεια της συνάρτησης `scipy.argrelextrema()`, μπορούμε να προσδιορίσουμε τα τοπικά μέγιστα της άνωθι συνάρτησης αυτοσυσχέτισης. Από αυτά, εκείνα που μπορούν να μας δώσουν έγκυρα αποτελέσματα για τον ρυθμό του μουσικού αποσπάσματος, εμφανίζονται ανά 2 (τα peaks με το μεγαλύτερο πλάτος). Ειδικότερα, παίρνουμε τα εξής αποτελέσματα:

'Salsa1: Autocorrelation peaks every 4059 samples.'

'Salsa1: Autocorrelation peaks approximately every 368 msec.'

'Salsa1: Beats Per Minute Rate = 162 bpm.'

Λαμβάνοντας υπόψη ότι κάθε παλμός/λεπτό (BPM) αντιστοιχεί σε 60Hz, μπορούμε να δούμε τα παραπάνω αποτελέσματα στο παρακάτω γράφημα της αυτοσυσχέτισης σε συνάρτησης με τους παλμούς ανά λεπτό, στο διάστημα 0-180 bpm:

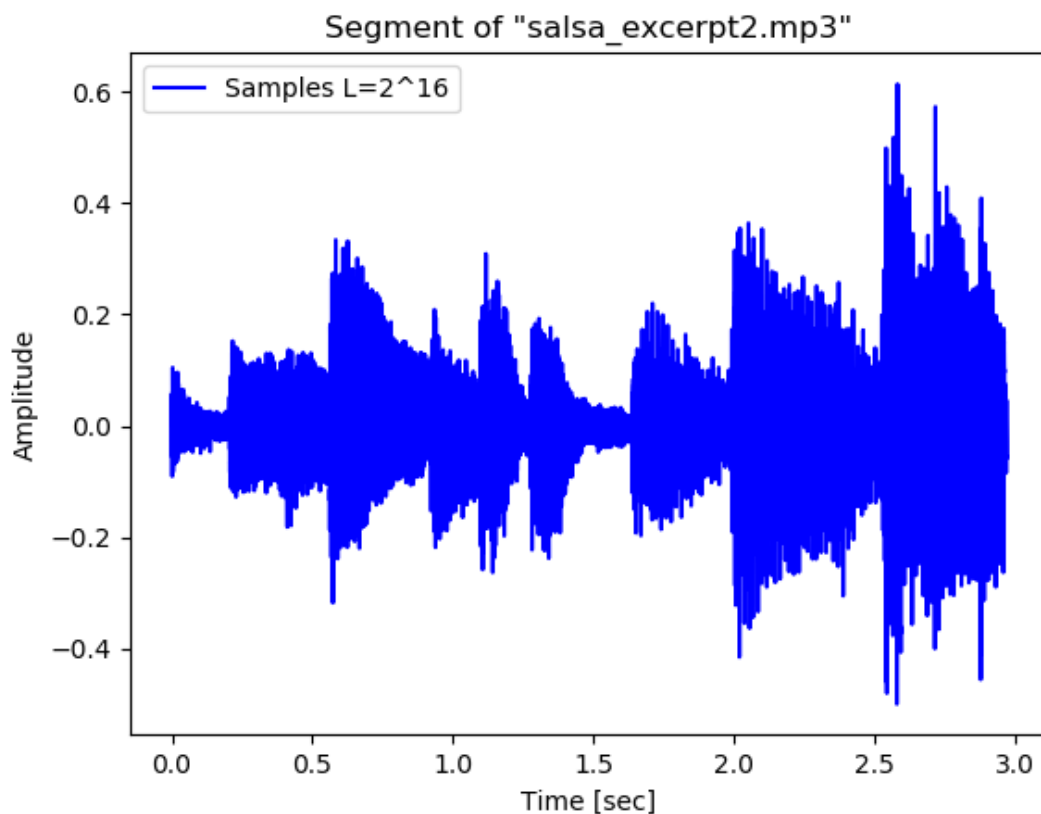


Πράγματι, παρατηρούμε ότι και η γραφική παράσταση παρουσιάζει peak στην περιοχή κοντά στα 160 bpm, επιβεβαιώνοντας, έτσι, το αποτέλεσμα ότι ο ρυθμός του συγκεκριμένου αποσπάσματος από το αρχείο `"salsa_excerpt1.mp3"` είναι **162 bpm**.

Πολυκλιμακωτή Ανάλυση μουσικού σήματος Salsa 2

4.1 Απομόνωση αποσπάσματος μουσικού σήματος:

Όμοια με πριν, κάνοντας χρήση της εντολής *librosa.load()* φορτώνουμε στον υπολογιστή το αρχείο *"salsa_excerpt2.mp3"* από το συμπληρωματικό υλικό της άσκησης, με συχνότητα δειγματοληψίας $f_s = 22.05\text{kHz}$. Στη συνέχεια, απομονώνουμε ένα απόσπασμα μήκους $L=2^{16}$ δειγμάτων (περίπου 3 δευτερολέπτων) από αυτό, το οποίο αντιστοιχεί στα δείγματα **60000:125536**. Παρακάτω διαφαίνεται η γραφική παράσταση του αποσπάσματος σε συνάρτηση με τον χρόνο:



Παρατηρούμε ότι το απόσπασμα του μουσικού σήματος παρουσιάζει peaks κάθε περίπου 0.25sec. Συνεπώς, εκτιμάμε πως η παλμική περιοδικότητα του αποσπάσματος είναι κατά προσέγγιση **4Hz**.

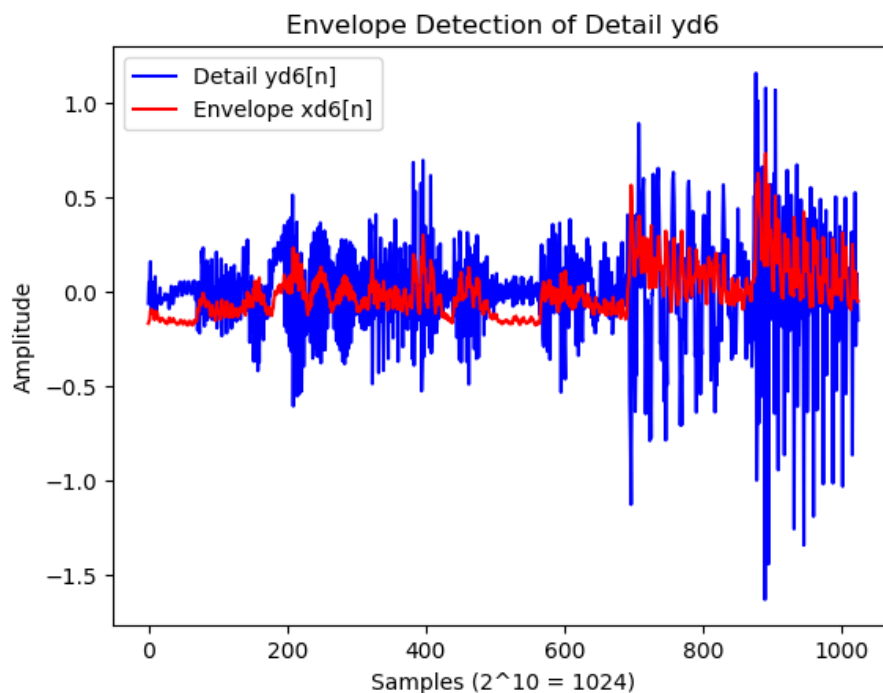
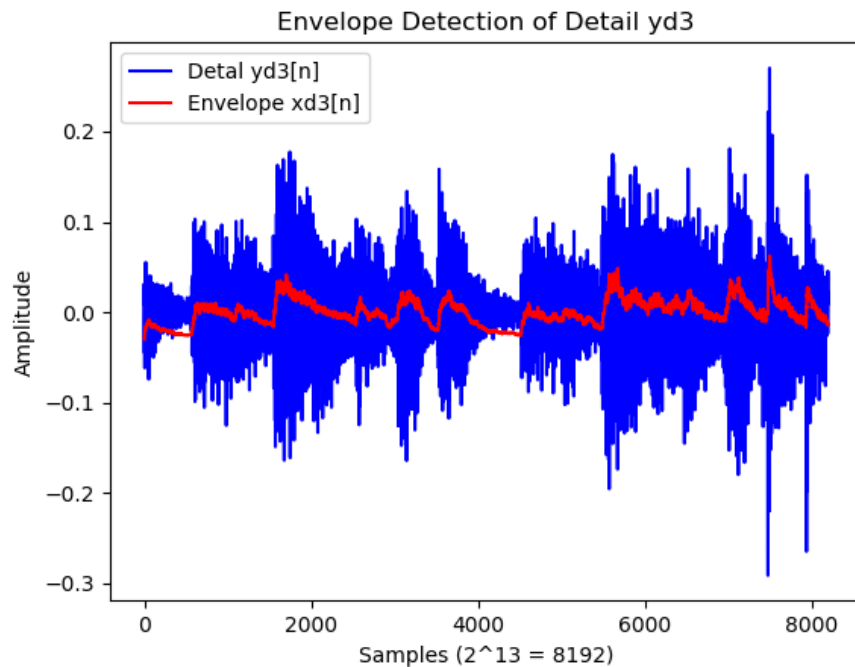
4.2 Πολυκλιμακωτή ανάλυση με χρήση DWT

Ακολουθούμε ακριβώς τα ίδια βήματα με πριν. Με τη βοήθεια της εντολής *pywt.wavedec()* εφαρμόζουμε στο άνωθι απόσπασμα τον Διακριτό Μετασχηματισμό Wavelet, ώστε να εξάγουμε την πολυκλιμακωτή του ανάλυση σε συνιστώσες. Χρησιμοποιούμε **κυματίδια Daubechies-4** και συνολικά **7 επίπεδα ανάλυσης**. Ως αποτέλεσμα, λαμβάνουμε 7 υψίσυχνες συνιστώσες λεπτομέρειας (detail) του αποσπάσματος, $\text{ydi}[n]$, $i = 1, \dots, 7$, και μία χαμηλόσυχη συνιστώσα προσέγγισης (approximation) $\text{ya7}[n]$. Επιπλέον, διαιρούμε καθεμία εκ των συνιστωσών με τον παράγοντα $\sqrt{2}$.

4.3 Εξαγωγή Περιβάλλουσας Συνιστωσών

Επιχειρούμε να εξάγουμε την περιβάλλουσα κάθε μιας εκ των συνιστωσών ακολουθώντας τα ίδια βήματα που ακολουθήσαμε στο προηγούμενο μουσικό απόσπασμα. Στην προκείμενη περίπτωση, επιλέγουμε $\alpha_0 = 0.003$ και εξάγουμε τις περιβάλλουσες $x_{di}[n]$, $i = 1, \dots, 7$ για κάθε μία από τις 7 συνιστώσες λεπτομέρειας, καθώς και την περιβάλλουσα $x_{a7}[n]$ της συνιστώσας προσέγγισης.

Παρατίθενται, ενδεικτικά, οι γραφικές παραστάσεις των σημάτων $y_{d3}[n]$, $y_{d6}[n]$, μαζί με τις περιβάλλουσες τους $x_{d3}[n]$, $x_{d6}[n]$, σε συνάρτηση με τον αριθμό δειγμάτων τους. Ειδικότερα:

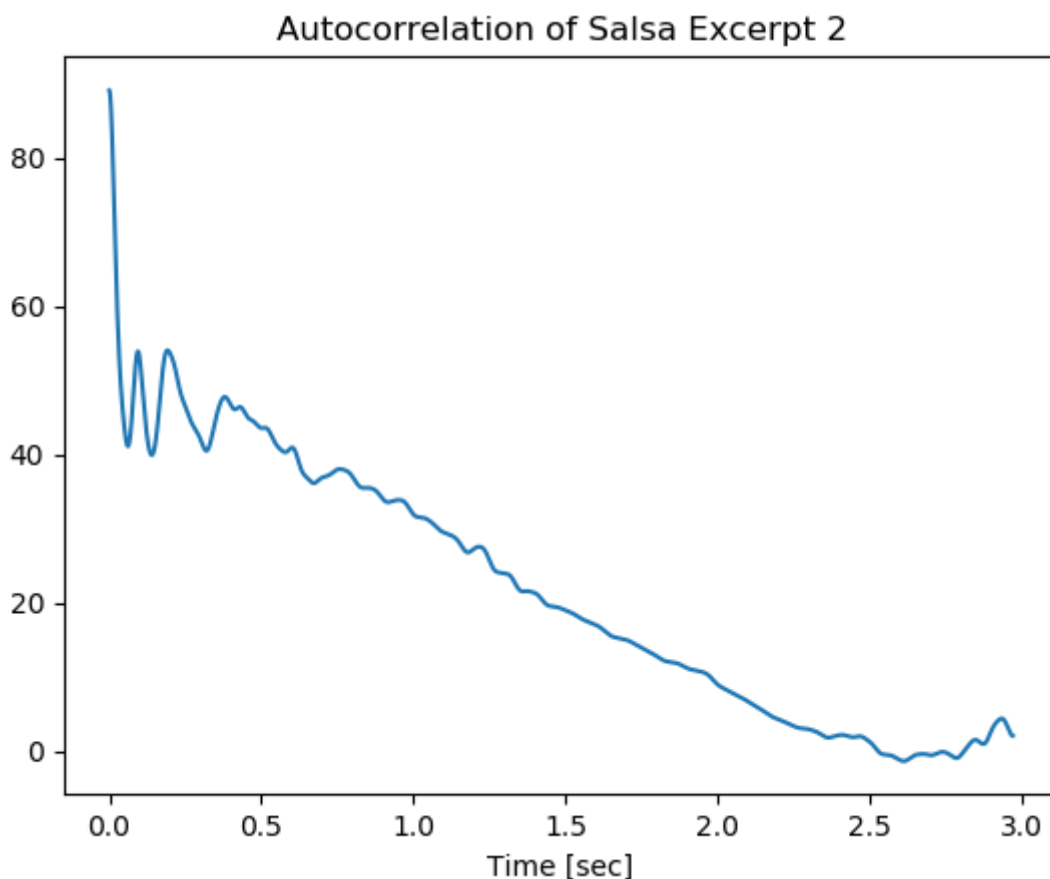


4.4 Άθροιση περιβάλλουσών με γραμμική παρεμβολή

Προκειμένου οι περιβάλλουσες των συνιστωσών να αποκτήσουν το ίδιο μήκος, εφαρμόζουμε **γραμμική παρεμβολή** με χρήση της εντολής `numpy.interp()`, αναγάγοντας κάθε περιβάλλουσα σε μήκος 32768 δειγμάτων. Στη συνέχεια, αθροίζουμε όλες τις ιδίου μήκους πλέον μήκους περιβάλλουσες και παίρνουμε το σήμα $x_{sum}[n]$.

Ακολουθώντας, με χρήση της εντολής `numpy.correlate()`, υπολογίζουμε την αυτοσυσχέτιση του σήματος $x_{sum}[n]$. Παρατηρούμε, ωστόσο, ότι η συνάρτηση αυτοσυσχέτισης παρουσιάζει **τραχύτητες**, γι' αυτό και την εξομαλύνουμε, εφαρμόζοντας σε αυτήν Γκαουσιανό φίλτρο διασποράς $\sigma^2 = 130$, με χρήση της εντολής `scipy.ndimage.gaussian.filter1d()`.

Παρακάτω φαίνεται σε γράφημα το δεξί συμμετρικό μέρος της αυτοσυσχέτισης του σήματος $x_{sum}[n]$ σε συνάρτηση με τον χρόνο:



Σημείωση:

Η μετατροπή του οριζόντιου άξονα από αριθμό δειγμάτων σε χρόνο έγινε λαμβάνοντας υπόψη ότι η συχνότητα δειγματοληψίας του σήματος είναι ίση με $f_s/2 = 11.025\text{kHz}$, καθώς το σήμα μας έχει πλέον το μισό αριθμό δειγμάτων από ότι το αρχικό απόσπασμα.

4.5 Ρυθμός (BPM) του μουσικού αποσπάσματος

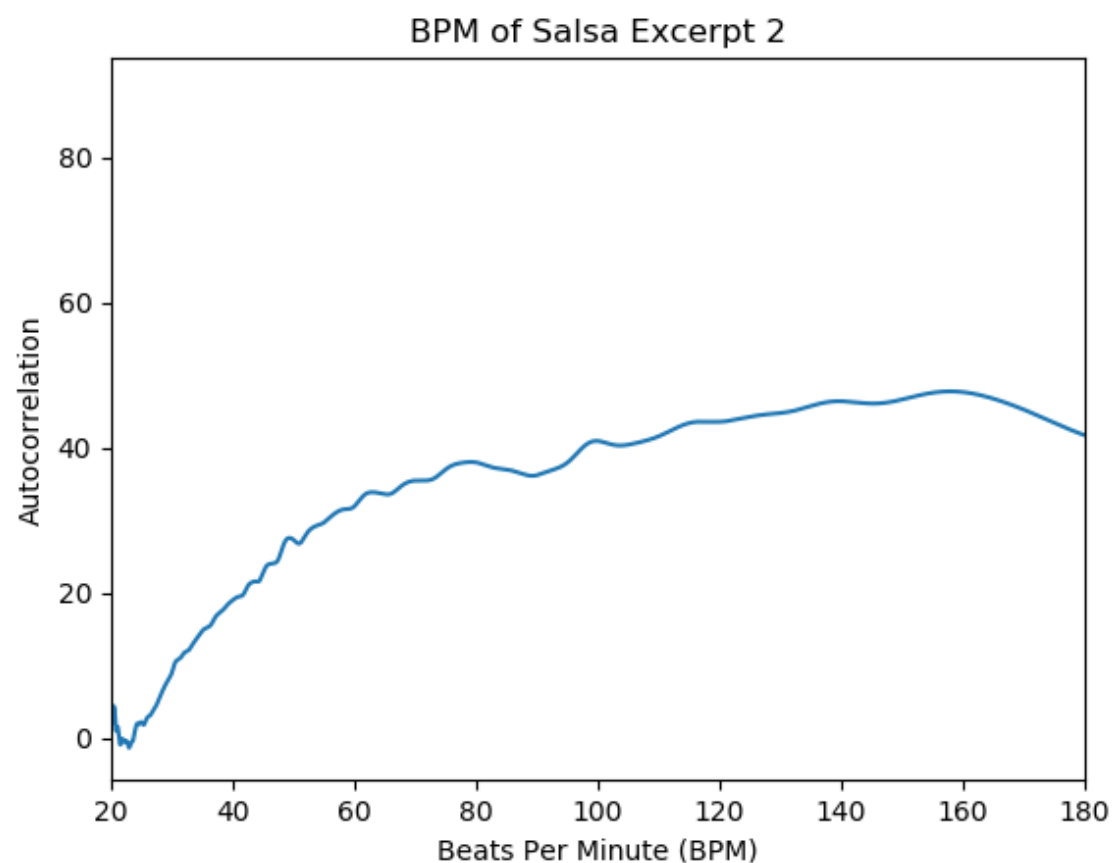
Με τη βοήθεια της συνάρτησης `scipy.argrextrema()`, μπορούμε να προσδιορίσουμε τα τοπικά μέγιστα της άνωθι συνάρτησης αυτοσυσχέτισης. Από αυτά, εκείνα που μπορούν να μας δώσουν έγκυρα αποτελέσματα για τον ρυθμό του μουσικού αποσπάσματος, εμφανίζονται ανά 2 (τα peaks με το μεγαλύτερο πλάτος). Ειδικότερα, παίρνουμε τα εξής αποτελέσματα:

'Salsa2: Autocorrelation peaks every 3915 samples.'

'Salsa2: Autocorrelation peaks approximately every 355 msec.'

'Salsa2: Beats Per Minute Rate = 168 bpm.'

Λαμβάνοντας υπόψη ότι κάθε παλμός/λεπτό (BPM) αντιστοιχεί σε 60Hz, μπορούμε να δούμε τα παραπάνω αποτελέσματα στο παρακάτω γράφημα της αυτοσυσχέτισης σε συνάρτησης με τους παλμούς ανά λεπτό, στο διάστημα 0-180 bpm:

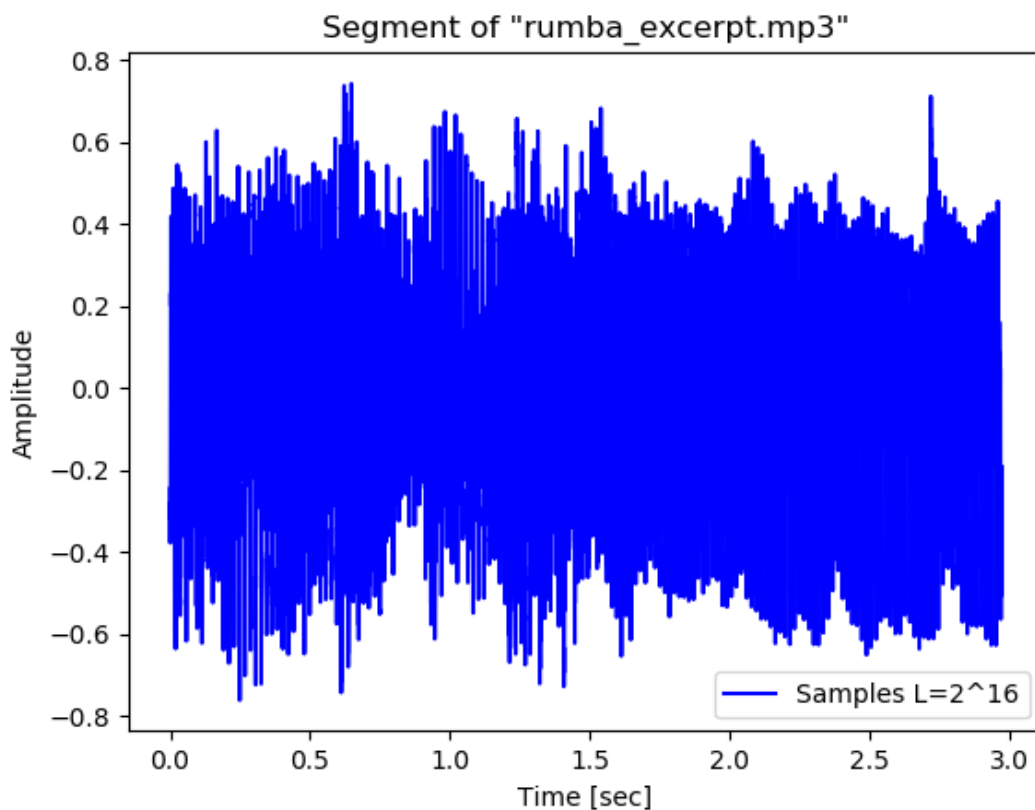


Πράγματι, παρατηρούμε ότι και η γραφική παράσταση παρουσιάζει peak στην περιοχή κοντά στα 160 bpm, επιβεβαιώνοντας, έτσι, το αποτέλεσμα ότι ο ρυθμός του συγκεκριμένου αποσπάσματος από το αρχείο `“salsa_excerpt2.mp3”` είναι **168 bpm**.

Πολυκλιμακωτή Ανάλυση μουσικού σήματος Rumba

4.1 Απομόνωση αποσπάσματος μουσικού σήματος:

Όμοια με πριν, κάνοντας χρήση της εντολής *librosa.load()* φορτώνουμε στον υπολογιστή το αρχείο *"rumba_excerpt.mp3"* από το συμπληρωματικό υλικό της άσκησης, με συχνότητα δειγματοληψίας $f_s = 22.05\text{kHz}$. Στη συνέχεια, απομονώνουμε ένα απόσπασμα μήκους $L=2^{16}$ δειγμάτων (περίπου 3 δευτερολέπτων) από αυτό, το οποίο αντιστοιχεί στα δείγματα **350000:415536**. Παρακάτω διαφαίνεται η γραφική παράσταση του αποσπάσματος σε συνάρτηση με τον χρόνο:



Παρατηρούμε ότι το απόσπασμα του μουσικού σήματος παρουσιάζει peaks κάθε περίπου 0.5sec. Συνεπώς, εκτιμάμε πως η παλμική περιοδικότητα του αποσπάσματος είναι κατά προσέγγιση **2Hz**.

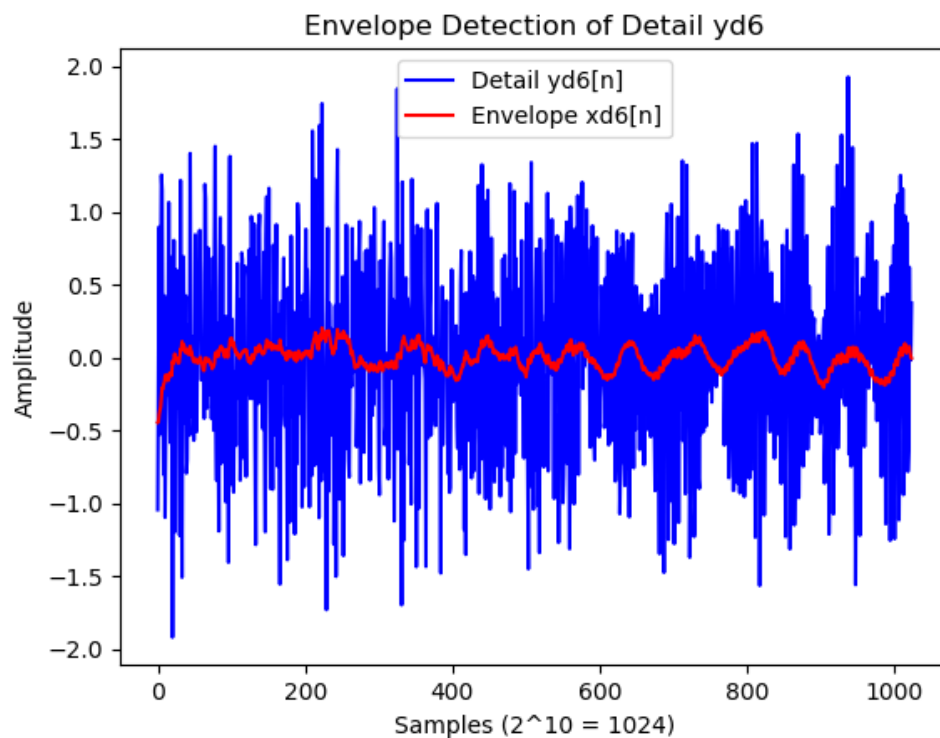
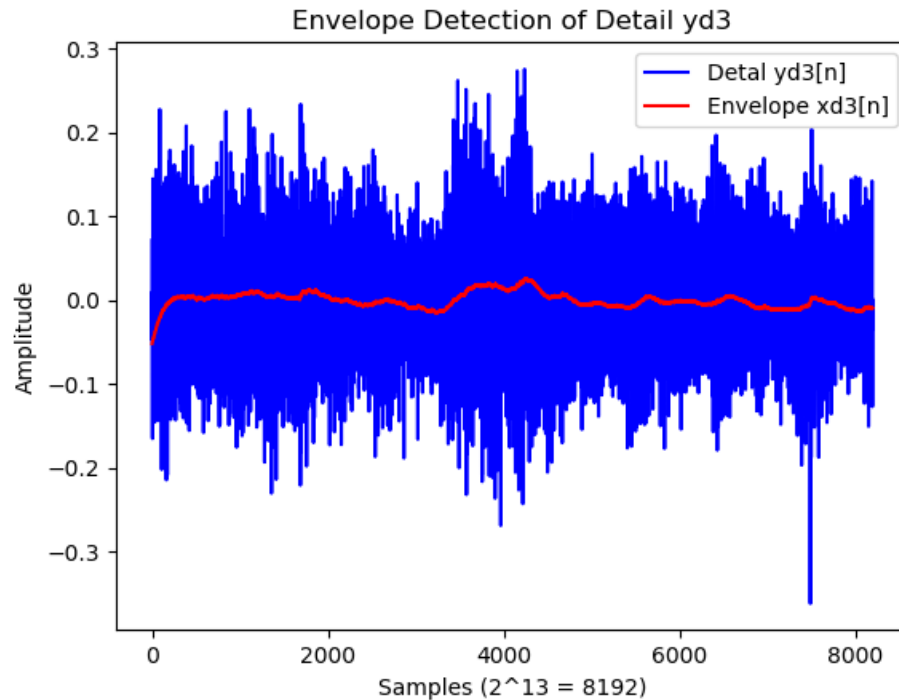
4.2 Πολυκλιμακωτή ανάλυση με χρήση DWT

Ακολουθούμε ακριβώς τα ίδια βήματα με πριν. Με τη βοήθεια της εντολής *pywt.wavedec()* εφαρμόζουμε στο άνωθι απόσπασμα τον Διακριτό Μετασχηματισμό Wavelet, ώστε να εξάγουμε την πολυκλιμακωτή του ανάλυση σε συνιστώσες. Χρησιμοποιούμε **κυματίδια Daubechies-4** και συνολικά **7 επίπεδα ανάλυσης**. Ως αποτέλεσμα, λαμβάνουμε 7 υψίσυχνες συνιστώσες λεπτομέρειας (detail) του αποσπάσματος, $y_{di}[n]$, $i = 1, \dots, 7$, και μία χαμηλόσυχη συνιστώσα προσέγγισης (approximation) $y_{a7}[n]$. Επιπλέον, διαιρούμε καθεμία εκ των συνιστωσών με τον παράγοντα $\sqrt{2}$.

4.3 Εξαγωγή Περιβάλλουσας Συνιστωσών

Επιχειρούμε να εξάγουμε την περιβάλλουσα κάθε μιας εκ των συνιστωσών ακολουθώντας τα ίδια βήματα που ακολουθήσαμε στο προηγούμενο μουσικό απόσπασμα. Στην προκείμενη περίπτωση, επιλέγουμε $\alpha_0 = 0.0005$ και εξάγουμε τις περιβάλλουσες $x_{di}[n]$, $i = 1, \dots, 7$ για κάθε μία από τις 7 συνιστώσες λεπτομέρειας, καθώς και την περιβάλλουσα $x_{a7}[n]$ της συνιστώσας προσέγγισης.

Παρατίθενται, ενδεικτικά, οι γραφικές παραστάσεις των σημάτων $y_{d3}[n]$, $y_{d6}[n]$, μαζί με τις περιβάλλουσες τους $x_{d3}[n]$, $x_{d6}[n]$, σε συνάρτηση με τον αριθμό δειγμάτων τους. Ειδικότερα:

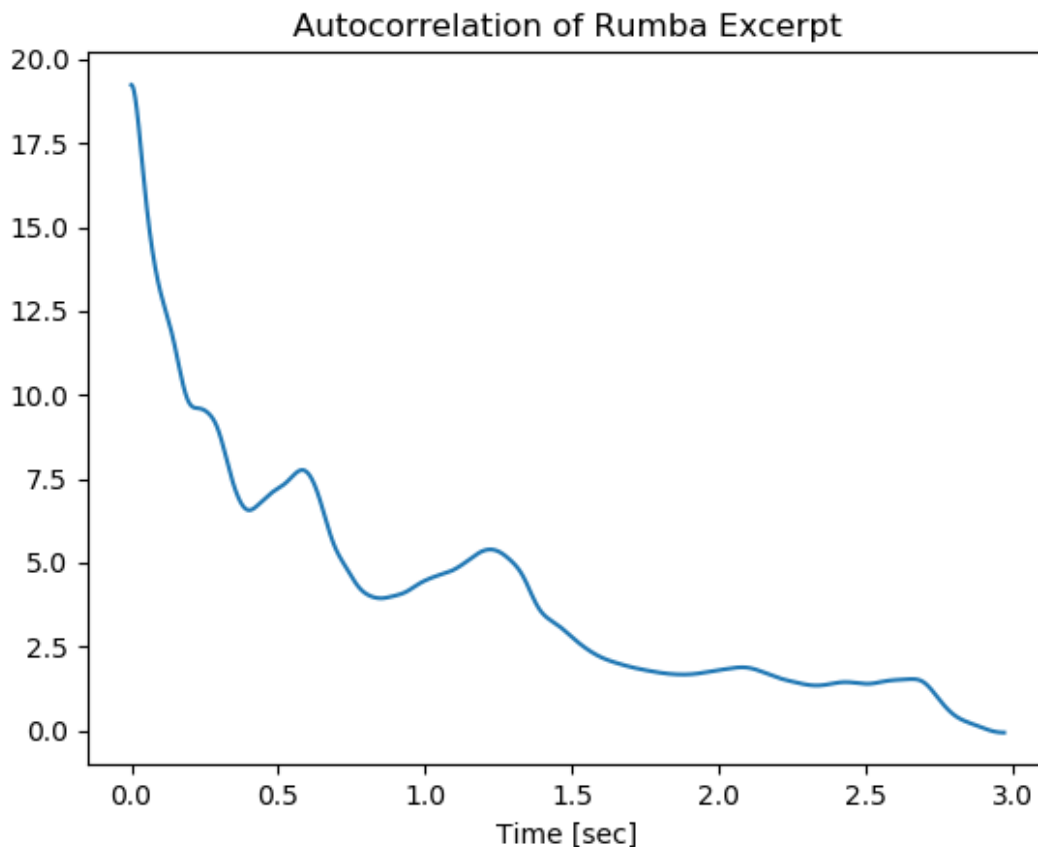


4.4 Άθροιση περιβάλλουσών με γραμμική παρεμβολή

Προκειμένου οι περιβάλλουσες των συνιστωσών να αποκτήσουν το ίδιο μήκος, εφαρμόζουμε **γραμμική παρεμβολή** με χρήση της εντολής `numpy.interp()`, αναγάγοντας κάθε περιβάλλουσα σε μήκος 32768 δειγμάτων. Στη συνέχεια, αθροίζουμε όλες τις ιδίου μήκους πλέον μήκους περιβάλλουσες και παίρνουμε το σήμα $x_{sum}[n]$.

Ακολούθως, με χρήση της εντολής `numpy.correlate()`, υπολογίζουμε την αυτοσυσχέτιση του σήματος $x_{sum}[n]$. Παρατηρούμε, ωστόσο, ότι η συνάρτηση αυτοσυσχέτισης παρουσιάζει **τραχύτητες**, γι' αυτό και την εξομαλύνουμε, εφαρμόζοντας σε αυτήν Γκαουσιανό φίλτρο διασποράς $\sigma^2 = 250$, με χρήση της εντολής `scipy.ndimage.gaussian.filter1d()`.

Παρακάτω φαίνεται σε γράφημα το δεξί συμμετρικό μέρος της αυτοσυσχέτισης του σήματος $x_{sum}[n]$ σε συνάρτηση με τον χρόνο:



Σημείωση:

Η μετατροπή του οριζόντιου άξονα από αριθμό δειγμάτων σε χρόνο έγινε λαμβάνοντας υπόψη ότι η συχνότητα δειγματοληψίας του σήματος είναι ίση με $f_s/2 = 11.025\text{kHz}$, καθώς το σήμα μας έχει πλέον το μισό αριθμό δειγμάτων από ότι το αρχικό απόσπασμα.

4.5 Ρυθμός (BPM) του μουσικού αποσπάσματος

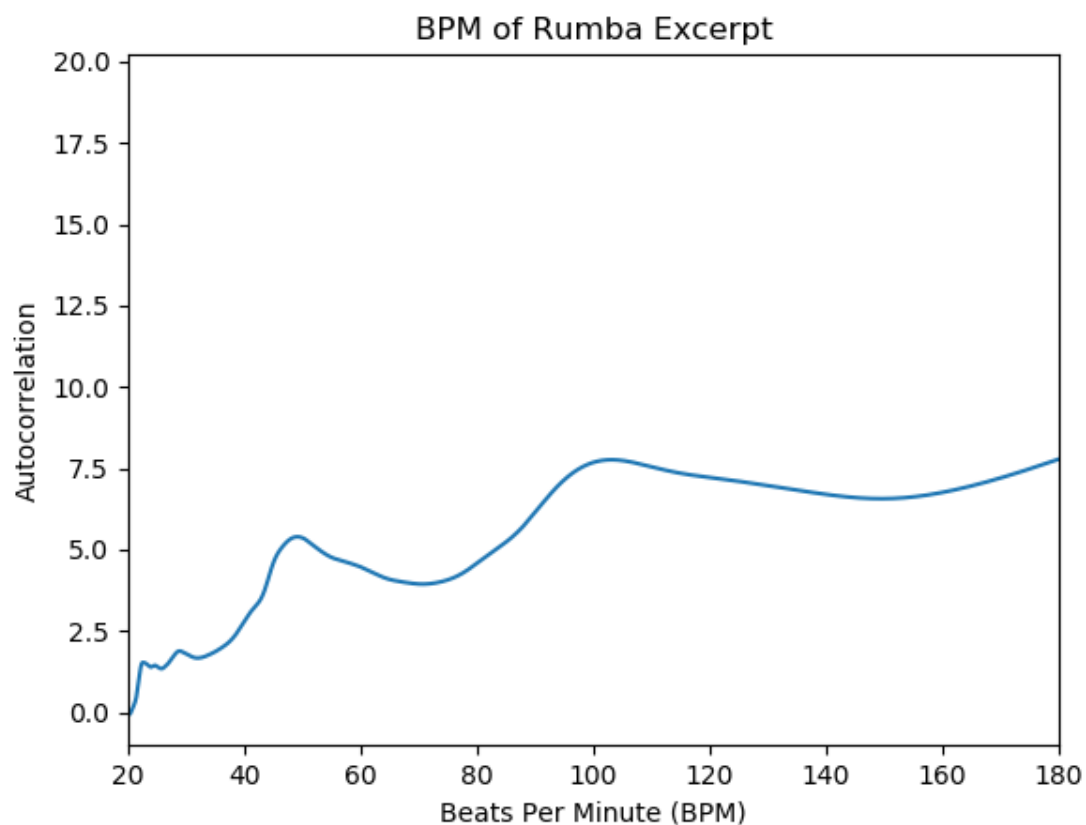
Με τη βοήθεια της συνάρτησης `scipy.argrelextrema()`, μπορούμε να προσδιορίσουμε τα τοπικά μέγιστα της άνωθι συνάρτησης αυτοσυσχέτισης. Από αυτά, εκείνα που μπορούν να μας δώσουν έγκυρα αποτελέσματα για τον ρυθμό του μουσικού αποσπάσματος, εμφανίζονται ανά 2 (τα peaks με το μεγαλύτερο πλάτος). Ειδικότερα, παίρνουμε τα εξής αποτελέσματα:

'Rumba: Autocorrelation peaks every 5725 samples.'

'Rumba: Autocorrelation peaks approximately every 519 msec.'

'Rumba: Beats Per Minute Rate = 115 bpm.'

Λαμβάνοντας υπόψη ότι κάθε παλμός/λεπτό (BPM) αντιστοιχεί σε 60Hz, μπορούμε να δούμε τα παραπάνω αποτελέσματα στο παρακάτω γράφημα της αυτοσυσχέτισης σε συνάρτησης με τους παλμούς ανά λεπτό, στο διάστημα 0-180 bpm:



Πράγματι, παρατηρούμε ότι και η γραφική παράσταση παρουσιάζει peak στην περιοχή κοντά στα 110 bpm, επιβεβαιώνοντας, έτσι, το αποτέλεσμα ότι ο ρυθμός του συγκεκριμένου αποσπάσματος από το αρχείο `"rumba_excerpt.mp3"` είναι **115 bpm**.

Συμπερασματικές Παρατηρήσεις

Έχοντας ακολουθήσει την ίδια διαδικασία για τα τρία μουσικά αρχεία του συμπληρωματικού υλικού της άσκησης, συμπεραίνουμε ότι μπορούμε να χρησιμοποιήσουμε τον Διακριτό Μετασχηματισμό Wavelet για να προσδιορίσουμε καλύτερα την περιοδική συμπεριφορά ενός μουσικού σήματος και εν συνεχεία να το κατατάξουμε σε ένα μουσικό είδος.

Χαρακτηριστικά, τα δύο πρώτα αρχεία *'salsa_excerpt1.mp3'*, *'salse_excerpt2.mp3'*, παρουσίασαν πολύ κοντινούς ρυθμούς (**162bpm** και **168bpm** αντίστοιχα), γεγονός που οφείλεται στο ότι ανήκουν στο ίδιο μουσικό είδος (salsa). Αντιθέτως, το αρχείο *'rumba_excerpt.mp3'* μας έδωσε ως αποτέλεσμα ρυθμό των **115bpm**, ο οποίος είναι αρκετά μικρότερος από αυτόν των άλλων δύο μουσικών αποσπασμάτων.

Συνεπώς, ο DWT αποτελεί ένα ισχυρό εργαλείο για τον προσδιορισμό της περιοδικής συμπεριφοράς ενός μουσικού σήματος και την κατηγοριοποίηση του ανά είδος με βάση αυτήν.

Βιβλιογραφία

- [1] [Ψηφιακή Επεξεργασία Σήματος](#), Alan V. Oppenheim, Ronald W. Schaffer, Εκδόσεις Φούντας, 2012
- [2] [Ψηφιακή Επεξεργασία Σήματος](#), Monson H. Hayes, Εκδόσεις Τζιόλα, 2014
- [3] [Ψηφιακή Επεξεργασία Φωνής: Θεωρία και Εφαρμογές](#), Lawrence R. Rabiner, Ronald W. Schaffer, 1η Έκδοση, Εκδόσεις Π.Χ.Πασχαλίδης, 2011