



Ψηφιακή Επεξεργασία Σημάτων  
3η εργαστηριακή Άσκηση

Ομάδα των 2 ατόμων

Νικόλαος Δημητριάδης

03114016

HMMY 6° εξάμηνο

el14016@central.ntua.gr

Θανάς Λεράι

03114156

HMMY 6° εξάμηνο

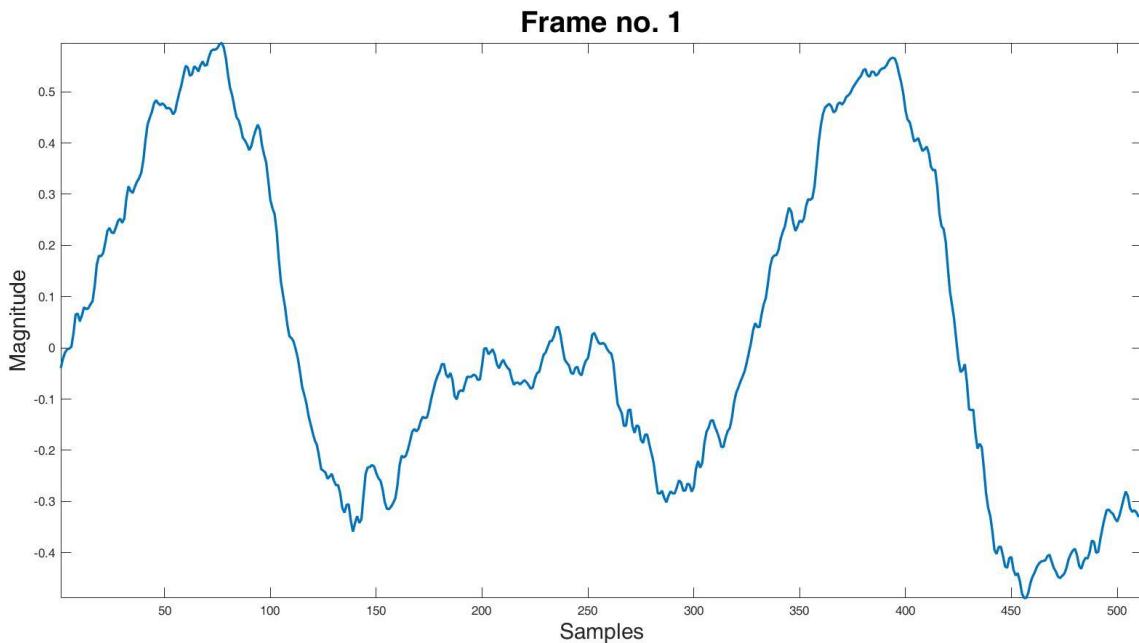
el14156@central.ntua.gr

## Τα παραδοτέα και οι παρατηρήσεις ακολουθούν την επεξήγηση των βημάτων και των αλγορίθμων.

### ΜΕΡΟΣ 1: Ψυχοακουστικό Μοντέλο I

#### Βήμα 1.0: Κανονικοποίηση του σήματος

Παρατηρούμε ότι το σήμα που μας δόθηκε, *music.wav*, αποτελείται από 2 κανάλια. Προκειμένου να πάρουμε, λοιπόν, ένα τελικό διάνυσμα, το *sig\_final*, υπολογίσαμε τη μέση τιμή των 2 καναλιών. Στη συνέχεια, διαιρέσαμε το *sig\_final* με τη μέγιστη απόλυτη τιμή του έτσι ώστε να επιτευχθεί η κανονικοποίηση του σήματος, δηλαδή να αποκτήσει τιμές εντός του διαστήματος [-1,1].



#### Βήμα 1.1: Φασματική Ανάλυση

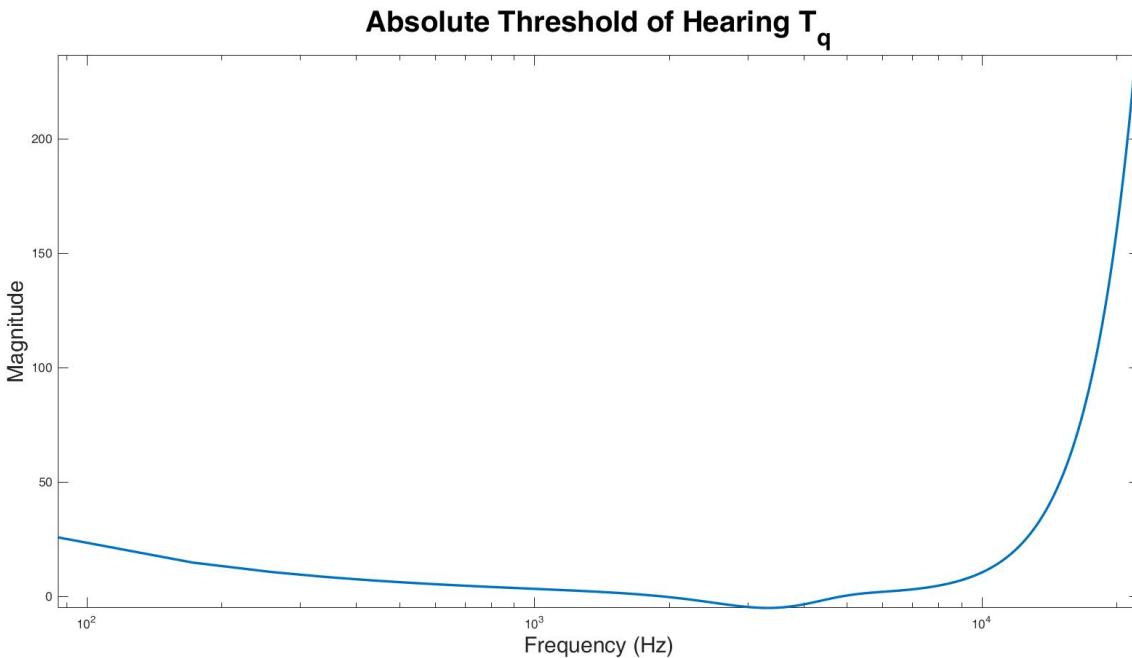
Στο στάδιο αυτό έχουμε ως στόχο να αποκτήσουμε μια υψηλής ανάλυσης εκτίμηση του φάσματος του σήματος εκφρασμένο σε μονάδες SPL (Sound Pressure Level), όπως δηλαδή εκφράζεται η πίεση του αέρα στο τύμπανο του αυτιού. Πρώτα, λοιπόν, ορίζουμε την κλίμακα Bark σύμφωνα με την παρακάτω εξίσωση:

$$b(f) = 13 \arctan(0.00076f) + 3.5 \arctan[(f/7500)^2]$$

Για την υλοποίηση στο MATLAB, ορίσαμε το διάνυσμα φυσικών συχνοτήτων  $f$  ως εξής:

$$f = linspace(0, \frac{F_s}{2}, \frac{L}{2})$$

όπου  $F_s = 44100\text{Hz}$  και  $L = 512$ . Επίσης η συνάρτηση  $\arctan$  υλοποιήθηκε με τη συνάρτηση του MATLAB  $\text{atan}$ .



Στη συνέχεια υπολογίζουμε το  $N$ -σημείων φάσμα ισχύος  $P(k)$  του σήματος με  $N=512$  δείγματα, όπως έχει καθιερωθεί στο πρότυπο MPEG Layer-1. Ο υπολογισμός αυτός γίνεται με βάση τον παρακάτω τύπο:

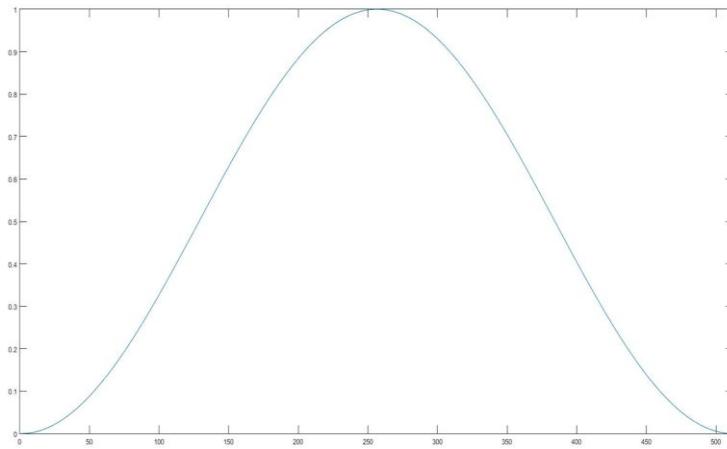
$$P(k) = PN + 10 \log_{10} \left| \sum_{n=0}^{N-1} w(n)x(n)e^{-j\frac{2\pi kn}{N}} \right|^2, \quad 0 \leq k \leq \frac{N}{2}$$

όπου το παράθυρο  $w(n)$  είναι παράθυρο hanning και δίνεται από τον τύπο:

$$w(n) = \frac{1}{2} \left[ 1 - \cos \left( \frac{2\pi n}{N} \right) \right]$$

Για την υλοποίηση στο MATLAB ακολουθήσαμε τα παρακάτω βήματα:

1. Με χρήση της συνάρτησης *buffer* παραθυροποιήσαμε το σήμα *sig\_final* σε παράθυρα μήκους  $L=512$  δειγμάτων με μηδενική επικάλυψη. Επισημαίνουμε ότι η συνάρτηση *buffer* χρησιμοποιεί τετραγωνικά παράθυρα.
2. Προκειμένου να αποδυναμώσουμε τα άκρα, πολλαπλασιάζουμε κάθε παράθυρο του σήματος *sig\_final* με το αντίστοιχο παράθυρο *hanning*. Ο λόγος για τον οποίο επιτυγχάνεται αυτή η αποδυνάμωση των άκρων είναι διότι το παράθυρο *hanning* έχει την παρακάτω μορφή:



3. Ύστερα υπολογίζουμε τον την απόλυτη τιμή του fft του κάθε hanning παραθυροποιημένου σήματος με μήκος 512 δείγματα.
4. Υψώνουμε στο τετράγωνο κάθε τέτοιο fft μετασχηματισμό και παίρνουμε το δεκαδικό λογάριθμο του αποτελέσματος.
5. Προσθέτουμε τη σταθερά κανονικοποίησης PN, η οποία ισούται με 90.302dB.
6. Τέλος, κρατάμε μόνο τα πρώτα 256 δείγματα από το κάθε διάνυσμα  $P(k)$  καθώς αυτό είναι συμμετρικό ως προς τον άξονα  $k=256$ .

### Βήμα 1.2: Εντοπισμός μασκών τόνων και θορύβου (Maskers)

Έχοντας υπολογίσει το φάσμα ισχύος  $P(k)$  εντοπίζουμε τώρα τοπικά μέγιστα (μάσκες) ανά critical band, τα οποία είναι μεγαλύτερα από τις γειτονικές τους συχνότητες τουλάχιστον κατά 7dB. Το εύρος της γειτονιάς υπολογισμού των μασκών διαφέρει ανά διακριτή συχνότητα  $k$  σύμφωνα με την παρακάτω εξίσωση:

$$\Delta_k \in \begin{cases} 2, & 2 < k < 63 \quad (0.17 - 5.5\text{kHz}) \\ [2, 3] & 63 \leq k < 127 \quad (5.5 - 11\text{kHz}) \\ [2, 6] & 127 \leq k \leq 250 \quad (11 - 20\text{kHz}) \end{cases}$$

Προκειμένου να βρούμε τις μάσκες του θορύβου (noise maskers) κάθε παράθυρου του σήματος  $sig\_final$ , χρησιμοποιούμε την έτοιμη συνάρτηση:

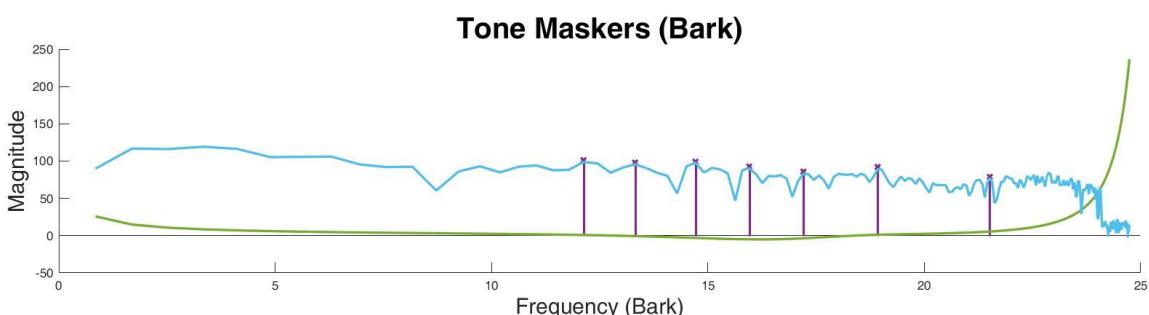
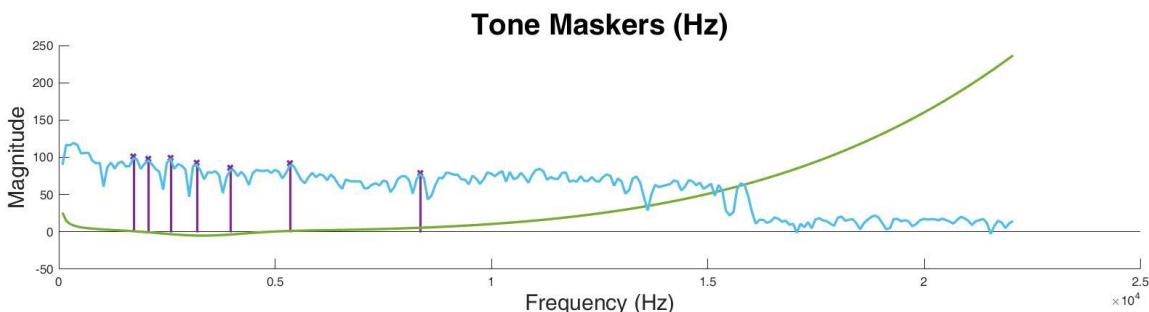
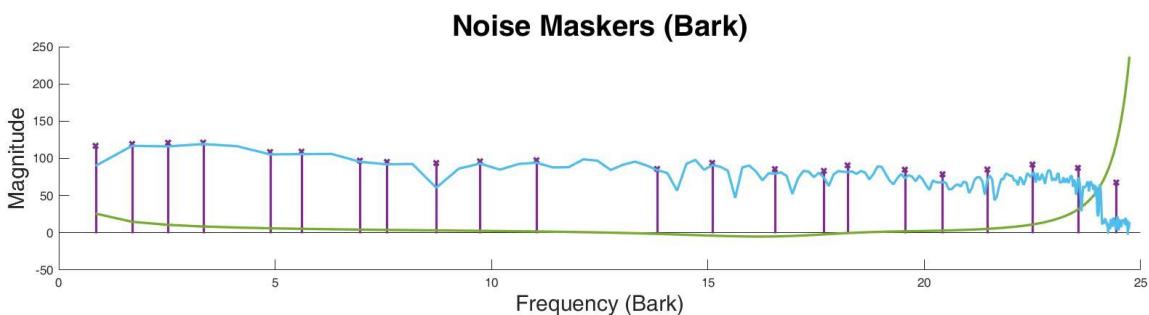
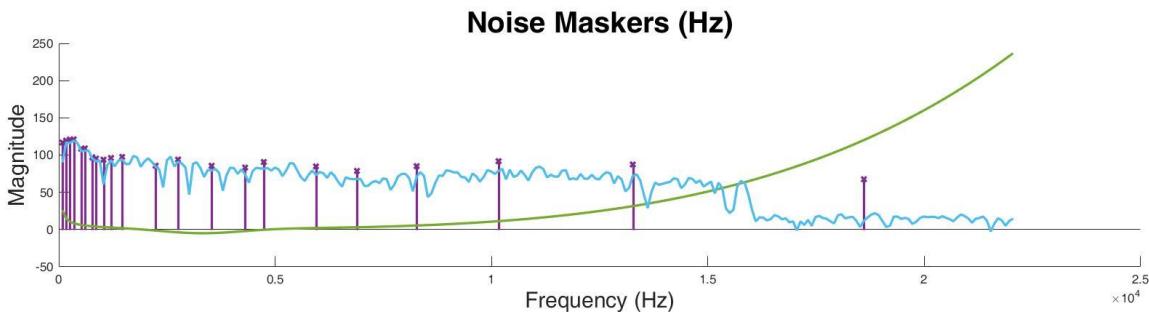
$$[P_{NM}] = \text{findNoiseMaskers}(P, P_{TM}, b),$$

όπου  $P$  το φάσμα ισχύος του κάθε παράθυρου του σήματος,  $b$  η κλίμακα bark που ορίστηκε στο βήμα 1.1 και  $P_{TM}(k)$  η ισχύς της μάσκας στη θέση  $k$ , η οποία δίνεται από τον τύπο:

$$P_{TM}(k) = \begin{cases} 10 \log_{10}(10^{0.1(P(k-1))} + 10^{0.1(P(k))} + 10^{0.1(P(k+1))})(\text{dB}), & \text{av } S_T(k) = 1 \\ 0, & \text{av } S_T(k) = 0 \end{cases}$$

ΟΠΟΥ:

$$S_T = \begin{cases} 0, & \text{αν } k \notin [3, 250] \\ P(k) > P(k \pm 1) \wedge P(k) > P(k \pm \Delta_k) + 7\text{dB}, & \text{αν } k \in [3, 250] \end{cases}$$

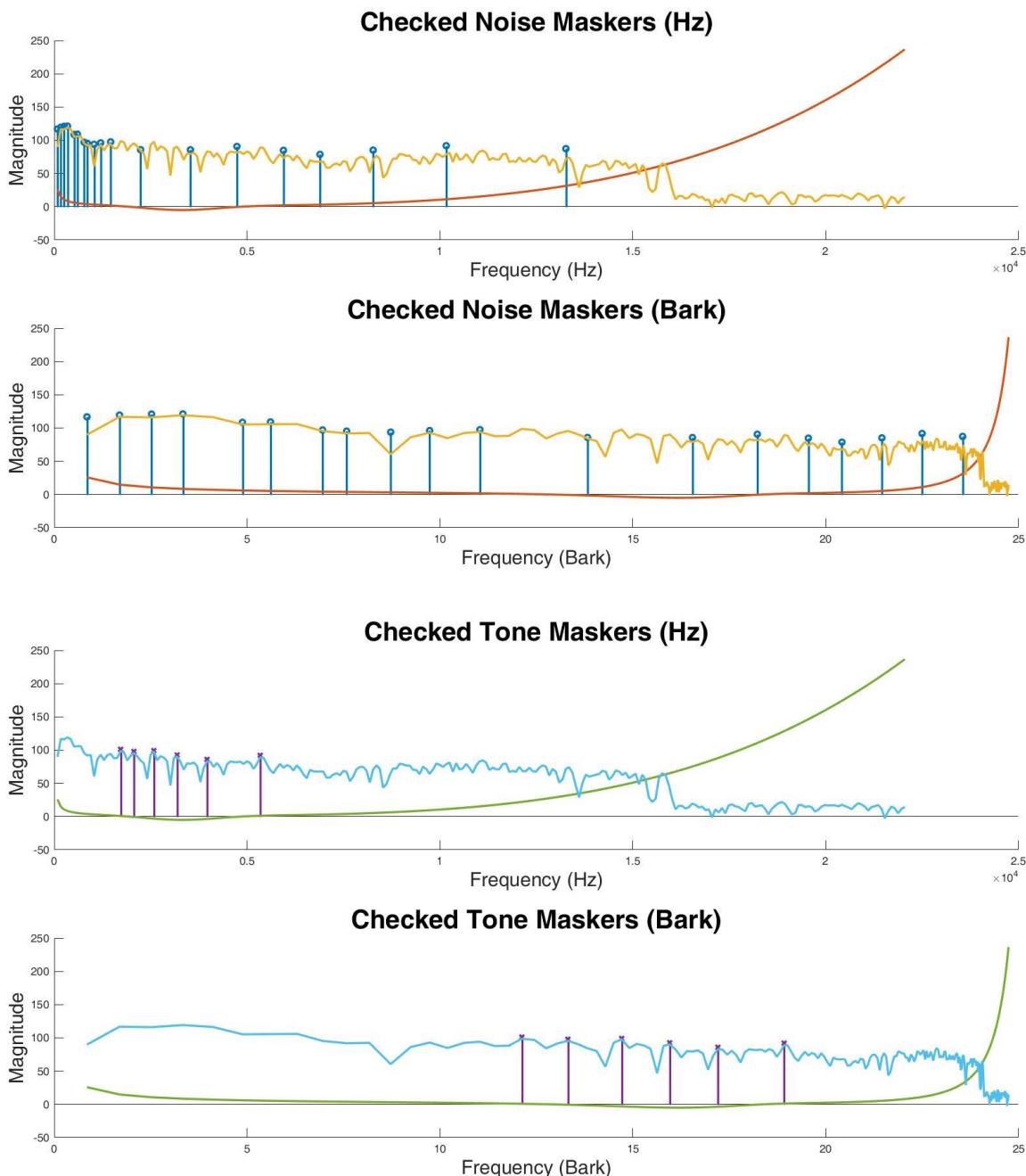


### Βήμα 1.3: Μείωση και αναδιοργάνωση των μασκών

Στο στάδιο αυτό θέλουμε να μειώσουμε τον αριθμό των μασκών χρησιμοποιώντας 2 διαφορετικά κριτήρια. Για το σκοπό αυτό κάνουμε χρήση της έτοιμης συνάρτησης:

$$[P_{TM}, P_{NM}] = \text{checkMaskers}(P_{TM}, P_{NM}, T_q, b)$$

όπου όλες οι παράμετροι έχουν οριστεί στο παραπάνω βήμα.



## Βήμα 1.4: Υπολογισμός των 2 διαφορετικών κατωφλίων κάλυψης (Individual Masking Thresholds)

Έχοντας ολοκληρώσει την μείωση του αριθμού των μασκών στο Βήμα 1.3, θα υπολογίσουμε τώρα τα 2 διαφορετικά κατώφλια κάλυψης. Καθένα από αυτά αντιπροσωπεύει το ποσοστό κάλυψης στο σημείο i το οποίο προέρχεται από την μάσκα τόνου ή θορύβου στο σημείο j. Τα 2 αυτά κατώφλια υπολογίζονται με βάση τους τύπους:

$$T_{TM}(i, j) = P_{TM}(j) - 0.275b(j) + SF(i, j) - 6.025(\text{dB SPL})$$

$$T_{NM}(i, j) = P_{NM}(j) - 0.175b(j) + SF(i, j) - 2.025(\text{dB SPL})$$

όπου το  $P_{TM}(j)$  και  $P_{NM}(j)$  η ισχύς των μασκών (τόνων και θορύβου αντίστοιχα) στο σημείο j και  $b(j)$  οι συχνότητες στην κλίμακα Bark. Η συνάρτηση  $SF(i, j)$  υπολογίζει την έκταση της κάλυψης από το σημείο j στο οποίο βρίσκεται η μάσκα έως το σημείο i το οποίο υφίσταται κάλυψη και μοντελοποιείται ως εξής:

$$SF(i, j) = \begin{cases} 17\Delta_b - 0.4P_{TM}(j) + 11, & -3 \leq \Delta_b < -1 \\ (0.4P_{TM}(j) + 6)\Delta_b, & -1 \leq \Delta_b < 0 \\ -17\Delta_b, & 0 \leq \Delta_b < 1 \\ (0.15P_{TM}(j) - 17)\Delta_b - 0.15P_{TM}(j), & 1 \leq \Delta_b < 8 \end{cases}$$

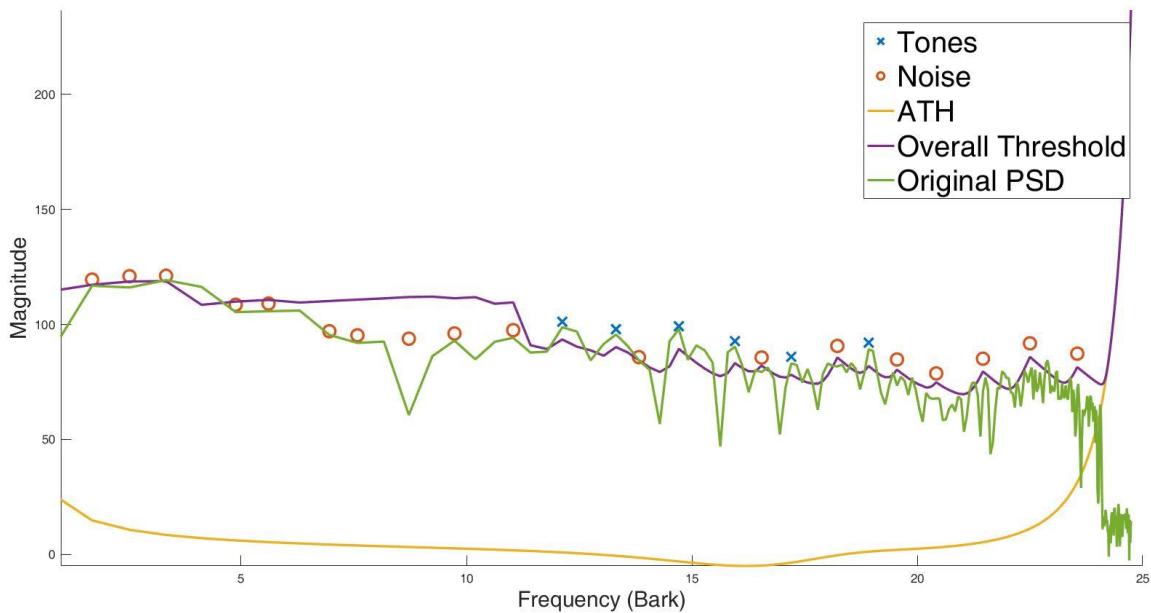
Η συνάρτηση  $SF(i, j)$  προσεγγίζει το ελάχιστο επίπεδο ισχύος το οποίο πρέπει να έχουν οι γειτονικές συχνότητες προκειμένου να γίνουν αντιληπτές από τον άνθρωπο. Η συνάρτηση αυτή υπολογίζεται για κάθε μάσκα τόνου και θορύβου, οι θέσεις j των οποίων μπορούν να εντοπιστούν ως j:  $P_{TM}(j) > 0$  και j:  $P_{NM}(j) > 0$ . Εδώ θεωρούμε πως η κάλυψη περιορίζεται σε μια γειτονιά των 12 Bark δηλαδή στις θέσεις i για τις οποίες  $b(i) \in [b(j) - 3, b(j) + 8]$ . Το  $\Delta_b = b(i) - b(j)$  είναι η διαφορά των συχνοτήτων σε κλίμακα Bark μεταξύ της θέσης j της μάσκας και του κάθε σημείου i της γειτονιάς. Αντίστοιχη ακριβώς δουλειά γίνεται και για τον υπολογισμό των κατωφλίων  $T_{NM}$ .

## Βήμα 1.5: Υπολογισμός του συνολικού κατωφλίου κάλυψης (Global Masking Threshold)

Για τη δημιουργία του συνολικού κατωφλίου σε κάθε διακριτή συχνότητα συνδυάζουμε τα ξεχωριστά κατώφλια κάλυψης που υπολογίσαμε στο Βήμα 1.4. Το συνολικό κατώφλι δίνεται από τον παρακάτω τύπο:

$$T_g(i) = 10 \log_{10} \left( 10^{0.1 T_q(i)} + \sum_{l=1}^L 10^{0.1 T_{TM}(i,l)} + \sum_{m=1}^M 10^{0.1 T_{NM}(i,m)} \right) \text{ dB SPL}$$

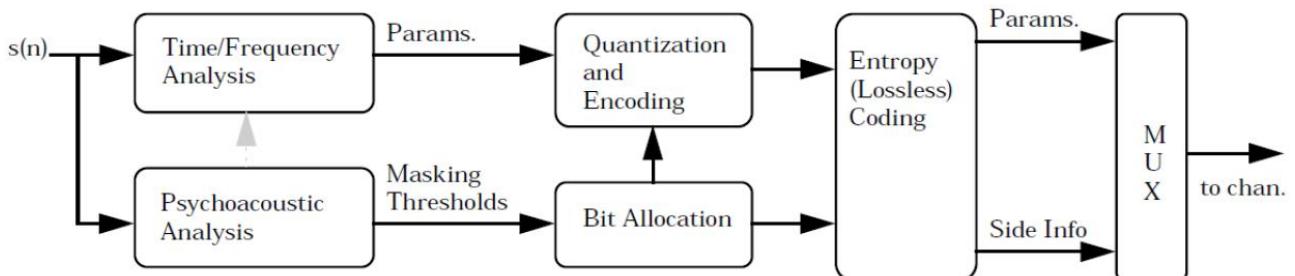
Όπου  $L, M$  ο αριθμός των μασκών τόνου και θορύβου αντίστοιχα.



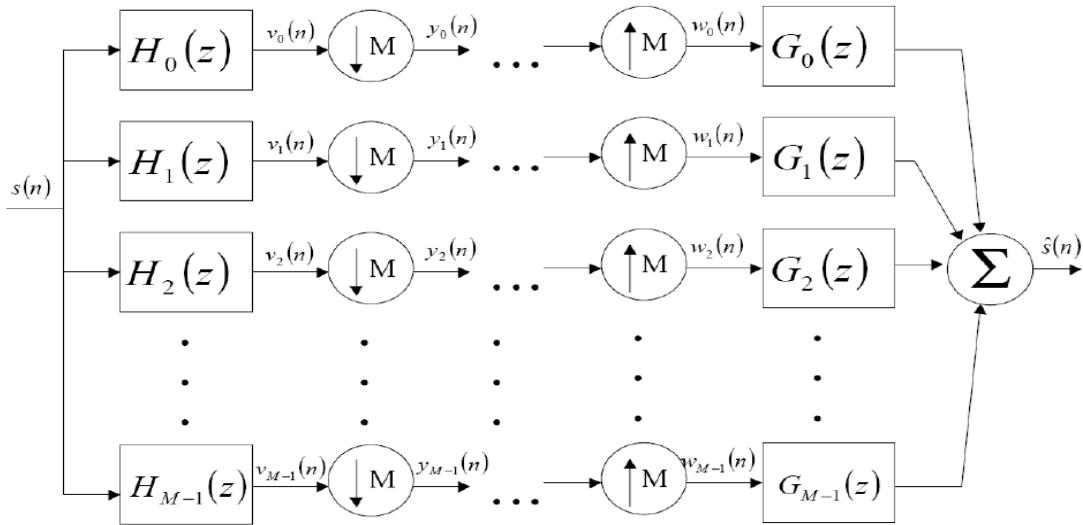
## ΜΕΡΟΣ 2: Χρονο-Συχνοτική Ανάλυση με συστοιχία Ζωνοπερατών Φίλτρων

Για την κβαντοποίηση και την κωδικοποίηση ενός ηχητικού σήματος, χρησιμοποιούνται κάποιες παράμετροι τις οποίες όλοι οι κωδικοποιητές ήχου εξάγουν μέσω κάποιου είδους χρονο-συχνοτικής ανάλυσης. Συνήθως για αυτήν την ανάλυση χρησιμοποιούνται συστοιχίες ζωνοπερατών φίλτρων οι οποίες καλύπτουν όλο το φάσμα συχνοτήτων. Το φάσμα διαιρείται σε υποζώνες συχνοτήτων και με αυτόν τον τρόπο παίρνουμε πληροφορίες για τη συχνοτική κατανομή του σήματος οι οποίες βοηθούν στην ταυτοποίηση των αντιληπτικά περιττών σημείων του σήματος. Έτσι, λοιπόν, η συστοιχία των ζωνοπερατών φίλτρων διευκολύνει την ανάλυση με το ψυχοακουστικό μοντέλο και επίσης μειώνονται τα στατιστικά redundancies μέσω αυτής της αποσύνθεσης του σήματος στις διαφορετικές συχνοτικές περιοχές.

Δίνουμε σχηματικά ένα παράδειγμα κωδικοποιητή ήχου:



Στο μέρος αυτό, φιλτράρουμε τα παράθυρα  $x(n)$  του σήματος με τη συστοιχία φίλτρων ανάλυσης  $h_k(n)$  και σύνθεσης  $g_k(n)$ . Έχουμε ως στόχο να δημιουργήσουμε μια συνάρτηση που υλοποιεί τη διαδικασία του παρακάτω σχήματος, η οποία δέχεται σαν είσοδο το κάθε πλαίσιο ανάλυσης  $x(n)$ , τη συστοιχία φίλτρων και το συνολικό κατώφλι κάλυψης που υπολογίστηκε στο Μέρος 1. Η έξοδος θα είναι το ανακατασκευασμένο σήμα  $\hat{x}$  και ο αριθμός των bits που χρησιμοποιήθηκαν για τη δημιουργία του.



## Βήμα 2.0: Συστοιχία Ζωνοπερατών Φίλτρων (Filterbank)

Στην άσκηση αυτή, για την κωδικοποίηση και αποκωδικοποίηση του συστήματος συμπίεσης χρησιμοποιούνται  $M=32$  φίλτρα ανάλυσης και σύνθεσης, των οποίων οι κρουστικές αποκρίσεις είναι:

$$h_k(n) = \sin \left[ \left( n + \frac{1}{2} \right) \frac{\pi}{2M} \right] \sqrt{\frac{2}{M}} \cos \left[ \frac{(2n+M+1)(2k+1)\pi}{4M} \right]$$

Και

$$g_k(n) = h_k(2M - 1 - n)$$

με μήκος  $L=2M$ , όπου  $\sin[(n + \frac{1}{2}) \frac{\pi}{2M}]$  ( $0 \leq n \leq L-1$  και  $0 \leq k \leq M-1$ ) ένα βαθυπερατό φίλτρο γνωστό ως “ημιτονικό παράθυρο”.

## Βήμα 2.1: Ανάλυση με Συστοιχία Φίλτρων

Στο βήμα αυτό, υπολογίζουμε τις αποκρίσεις  $u_k(n)$  οι οποίες βρίσκονται από τη συνέλιξη του κάθε παραθύρου του σήματος  $sig\_final(n)$  μήκους  $N=512$  δειγμάτων με τις αποκρίσεις  $h_k(n)$  των φίλτρων σύνθεσης. Η συνέλιξη επιτυγχάνεται με τη ρουτίνα `conv` στο περιβάλλον του MATLAB. Στη συνέχεια υπολογίζουμε τις αποκρίσεις  $y_k(n)$  οι οποίες αποτελούν τον αποδεκατισμό των αποκρίσεων  $u_k(n)$  με βάση τη σχέση:

$$y_k(n) = v_k(Mn).$$

όπου  $M = 32$ . Το decimation υλοποιείται με την ρουτίνα `downsample` του MATLAB.

## Βήμα 2.2: Κβαντοποίηση

Ο κβαντιστής αποτελεί ένα σημαντικό στοιχείο κάθε ψηφιακού κωδικοποιητή μέσω του οποίου οι τιμές των δειγμάτων μιας διακριτής ακολουθίας αντιστοιχίζονται σε αριθμημένα επίπεδα κβάντισης. Υλοποιούμε έναν προσαρμοζόμενο ομοιόμορφο κβαντιστή  $2^{B_k}$  επιπέδων, όπου  $B_k$  ο αριθμός των bits κωδικοποίησης ανά δείγμα της ακολουθίας  $y_k(n)$  στο τρέχον πλαίσιο ανάλυσης  $x(n)$  του σήματος. Το βήμα Δ αυτού του κβαντιστή προσαρμόζεται σε κάθε πλαίσιο ανάλυσης. Ο αριθμός των επιπέδων του κβαντιστή είναι τέτοιος ώστε το σφάλμα κβαντισμού να μη γίνεται αντιληπτό από τον άνθρωπο μετά τη συμπίεση του σήματος μουσικής. Το μέγιστο ανεκτό σφάλμα συνδέεται με το συνολικό κατώφλι κάλυψης  $T_g(i)$  και η σχέση μέσω της οποίας υπολογίζεται είναι η παρακάτω:

$$B_k = \left\lceil \log_2 \left( \frac{2^{16}}{T_g} \right) - 1 \right\rceil$$

όπου  $R$  το πλήθος των βαθμίδων έντασης του αρχικού σήματος  $sig\_final(n)$ . Το  $T_g(i)$  ορίζεται στο διάστημα ορισμού του κάθε φίλτρου ανάλυσης, δηλαδή στα  $i$  για τα οποία ισχύει  $f(i)$  (ανηκει)  $[f_k - \frac{F_s \pi}{2M}, f_k + \frac{F_s \pi}{2M}]$ ,  $k=1,2,\dots,M$  όπου  $f_k$  η κεντρική συχνότητα κάθε φίλτρου. Με στόχο ένα μικρότερο σφάλμα κβαντισμού, στον προσαρμοζόμενο κβαντιστή η θέση του πρώτου επιπέδου ταυτίζεται με την ελάχιστη τιμή του σήματος σε κάθε πλαίσιο ανάλυσης. Το βήμα Δ ρυθμίζεται βάσει του  $B_k$  και του εκάστοτε  $[x_{min}, x_{max}]$  στο τρέχον πλαίσιο ανάλυσης. Αξίζει να σημειωθεί ότι οι διαφορετικοί μέθοδοι κβάντισης ακολούθησαν και διαφορετική μέθοδο υπολογισμού. Πιο συγκεκριμένα, για τον προσαρμοζόμενο κβαντιστή χρησιμοποιήθηκε η σχέση:

$$y_{new}(x) = \Delta \cdot \left\lceil \frac{y(x)}{\Delta} + \frac{1}{2} \right\rceil$$

Όπου  $y_{new}(x)$  η έξοδος του κβαντιστή και  $y$  η είσοδος(downsampled σήμα). Το ακέραιο μέρος υλοποιήθηκε στο MATLAB με τη ρουτίνα `floor()`. Αντίθετα, στο μη-προσαρμοζόμενο ομοιόμορφο κβαντιστή, αρχικά βρέθηκαν οι στάθμες κβάντισης χρησιμοποιώντας 8 bits και σημείο εκκίνηση τοι -1 και, στη συνέχεια, κωδικοποιήθηκε το κάθε δείγμα βρίσκοντας την ελάχιστη απόσταση από τα εν λόγω επίπεδα κβάντισης.

## Βήμα 2.3: Σύνθεση

Στη συνέχεια, υπολογίζουμε τα  $w_k(n)$ , τα οποία αποτελούν μια υπερδειγματολειψία των κβαντισμένων  $\hat{y}_k$ , σύμφωνα με τη σχέση:

$$w_k(n) = \begin{cases} \hat{y}_k(n/M), & n = 0, M, 2M, 3M, \dots \\ 0, & \text{αλλιώς.} \end{cases}$$

Έστερα, τα  $w_k(n)$  συνελίσσονται με τα φίλτρα σύνθεσης και οι προκύπτουσες αποκρίσεις στέλνονται σε ένα αθροιστή προκειμένου να προκύψει το σήμα εξόδου  $\widehat{s(n)}$ . Αξίζει να

αναφέρουμε πως η έξοδος  $\widehat{s(n)}$  θα είναι μια μετατόπιση του σήματος εισόδου  $x(n-n_0)$ , αν το σφάλμα κβαντισμού είναι μηδενικό.

Η ανασύνθεση του σήματος γίνεται με τη μέθοδο Overlap Add. Χρησιμοποιείται επικάλυψη ίση με  $N-M_{\text{τελικό}} = 639-512=127$ , όπου  $N=512$  το μήκος του παραθύρου που χρησιμοποιείται ως είσοδος στη διάταξη των φίλτρων και  $M_{\text{τελικό}} = 639$  το μήκος του σήματος εξόδου από την λόγω διάταξη. Αναλυτικότερα, το μήκος  $M$  προκύπτει ως εξής.

1. Συνέλιξη σήματος μήκους  $N=512$  δείγματα με φίλτρο μήκους  $L=64$  δείγματα και προκύπτει σήμα μήκους  $M_1=512+64-1=575$ ,
2. Υποδειγματοληψία με συντελεστή  $M=32$ , και προκύπτει σήμα μήκους  $M_2=18$  δείγματα,
3. Επεξεργασία και κβάντιση,
4. Υπερδειγματοληψία με συντελεστή  $M=32$  και προκύπτει μήκος σήματος  $M_3=576$  δείγματα.
5. Συνέλιξη σήματος μήκους  $N=512$  δείγματα με φίλτρο μήκους  $L=64$  δείγματα και προκύπτει σήμα μήκους  $M_{\text{τελικό}}=576+64-1=639$ .

# ΠΑΡΑΔΟΤΕΑ

## Compression Ratio

Το αρχικό σήμα κωδικοποιείται με 16 bits ανά δείγμα. Επομένως, για τον ομοιόμορφο, μη προσαρμοζόμενο κβαντιστή που χρησιμοποιεί 8 bits ανά δείγμα, το compression ratio είναι:

$$\text{compression ratio}_{\text{non-adaptive}, \text{uniform}} = 2.$$

Για τον προσαρμοζόμενο κβαντιστή, ωστόσο, το ποσοστό συμπίεσης προκύπτει διαφορετικά. Αναλυτικότερα, λόγω της προσαρμοζόμενης κβάντισης κάθε παράθυρο κωδικοποιείται με μεταβλητό αριθμό bits ίσο με το  $B_k$ . Απαιτούνται, λοιπόν, για κάθε παράθυρο:

- 16 bits για το  $\Delta$ ,
- 16 bits για το minimum του εκάστοτε παραθύρου, αφού προκύπτει από το αρχικό 16-bit-ο σήμα,
- $B_k \cdot L$ , όπου  $L=18$  το μήκος του downsampled σήματος.

Συνολικά λοιπόν απαιτούνται

$$\text{bits}_{/\text{κβαντιστή}} = B_k \cdot L + 32 \text{ bits}$$

Πειραματικά, προκύπτει ότι

$$\text{compression ratio}_{\text{adaptive}, \text{non-uniform}} = 1.3343$$

Όπως είναι αναμενόμενο, ο προσαρμοζόμενος κβαντιστής χρησιμοποιεί περισσότερα bits για την κωδικοποίηση του αρχικού σήματος. Αυτό είναι ξεκάθαρο και από τις παραπάνω εξισώσεις μιας και παρατηρούμε ότι  $B_k \geq 8$  για όλο το σήμα.

## Μέσο Τετραγωνικό Λάθος

Για την εύρεση του μέσου τετραγωνικού λάθους βρίσκουμε το άθροισμα των τετραγώνων των αποκλίσεων του ανακατασκευασμένου σήματος με το αρχικό (κανονικοποιημένο) και διαιρούμε με το μήκους του αρχικού σήματος. Αυτό συμβαίνει, διότι το αρχικό σήμα εισόδου και το ανακατασκευασμένο σήμα εξόδου έχουν ελαφρώς διαφορετικό αριθμό δειγμάτων (λόγω ανασύνθεσης και συνελίξεων) και, προφανώς, υπολογίζουμε με βάση το μήκος του αρχικού.

Βρίσκουμε ότι:

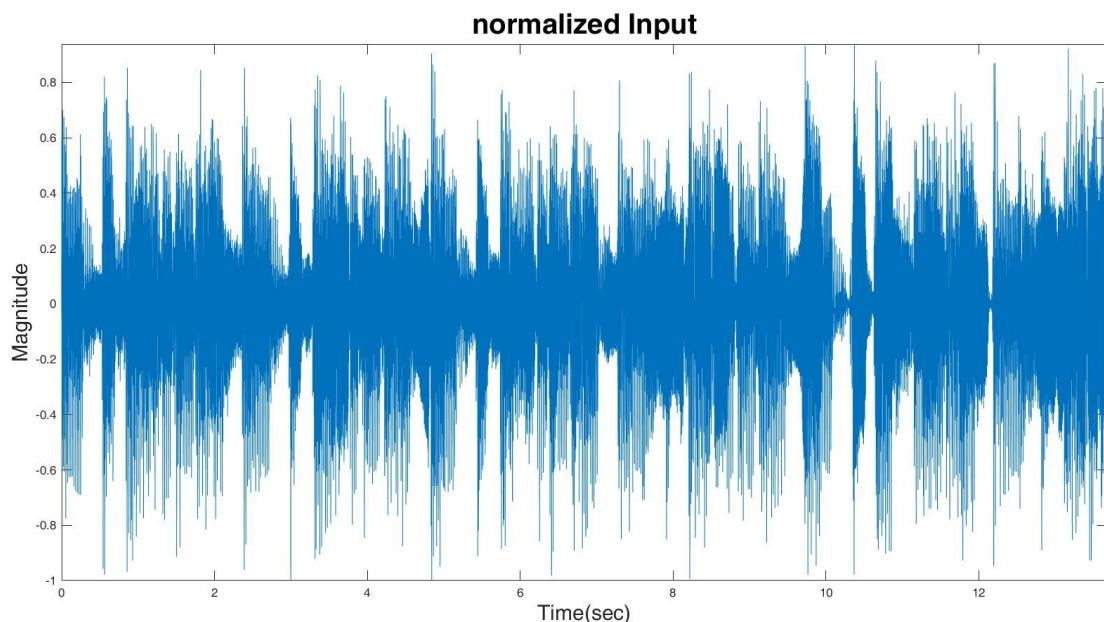
$$mse_{\text{non-adaptive}, \text{uniform}} = 0.0171$$

$$mse_{\text{adaptive}, \text{non-uniform}} = 2.418 \cdot 10^{-7}$$

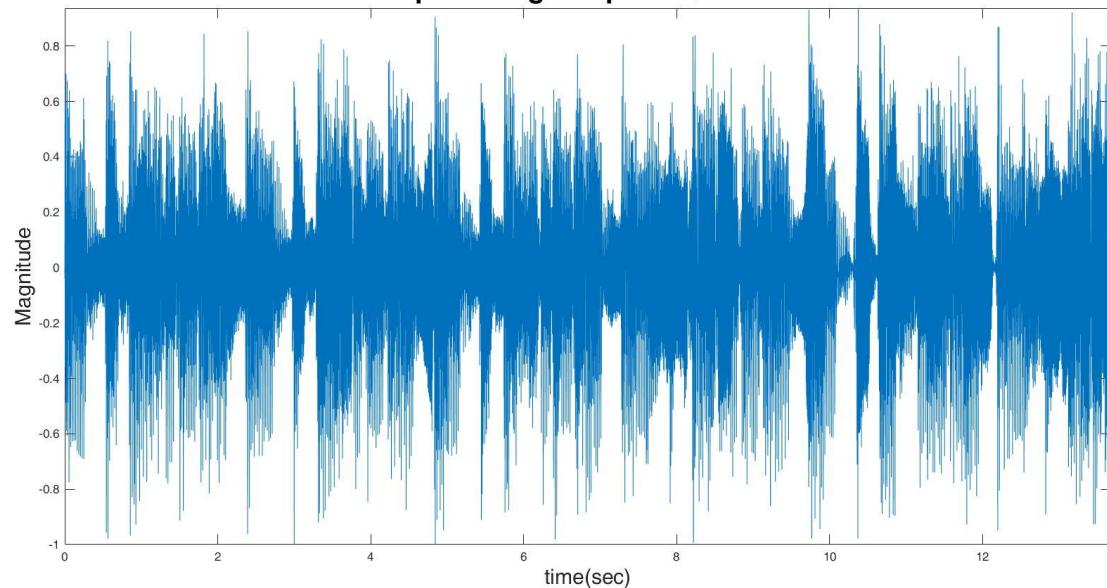
Είναι προφανές ότι προσαρμοζόμενος κβαντιστής πετυχαίνει μία πολύ χαμηλότερη απόκλιση σε σχέση με το αρχικό σήμα, αφενός επειδή κβαντίζει στο διάστημα τιμών του κάθε παραθύρου και όχι αόριστα στο διάστημα τιμών όλου του σήματος και, αφετέρου, επειδή προσαρμόζεται στο ψυχοακουστικό μοντέλο, μειώνοντας το θόρυβο και διατηρώντας τη σημαντική πληροφορία. Αυτό είναι ξεκάθαρο και από το ακόλουθα σχήματα των κυματομορφών, όπου είναι εμφανές ότι το κωδικοποιημένο σήμα με προσαρμοζόμενο, ομοιόμορφο κβαντιστή πετυχαίνει πολύ καλύτερη απεικόνιση σε σχέση το κωδικοποιημένο σήμα με μη-προσαρμοζόμενο, ομοιόμορφο κβαντιστή όσον αφορά το αρχικό σήμα (κανονικοποιημένο).

Τη διαφορά στην ποιότητα υπογραμμίζει και μαρτυρά ιδιαίτερα η απεικόνιση του λάθους, όπου η διαφορά στην κλίμακα του οριζόντιου άξονα είναι πολλών τάξεων μεγέθους.

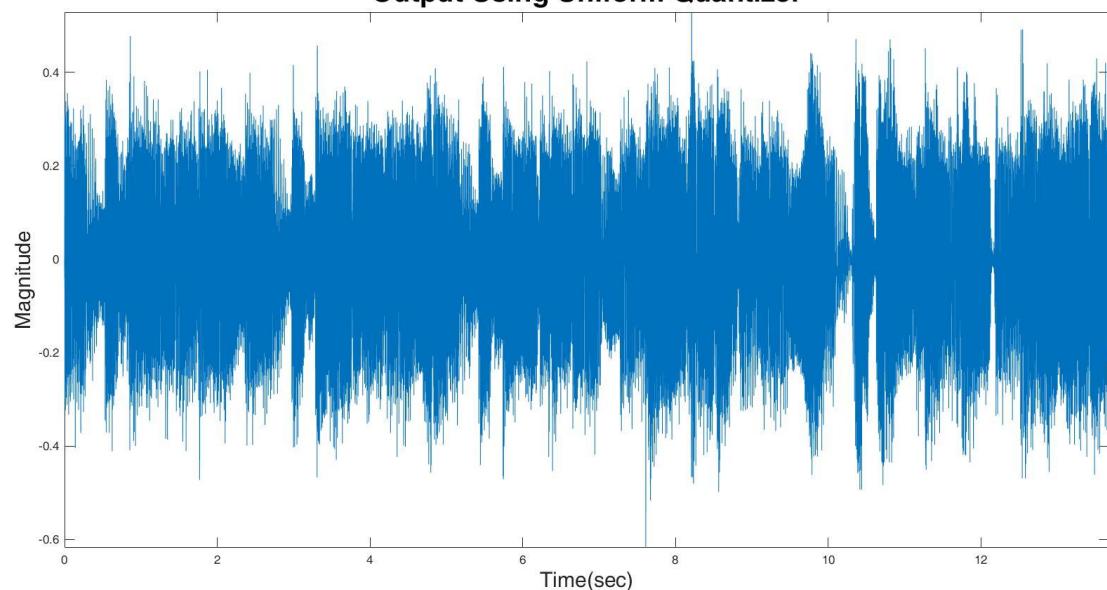
## Σχήματα κυματομορφών

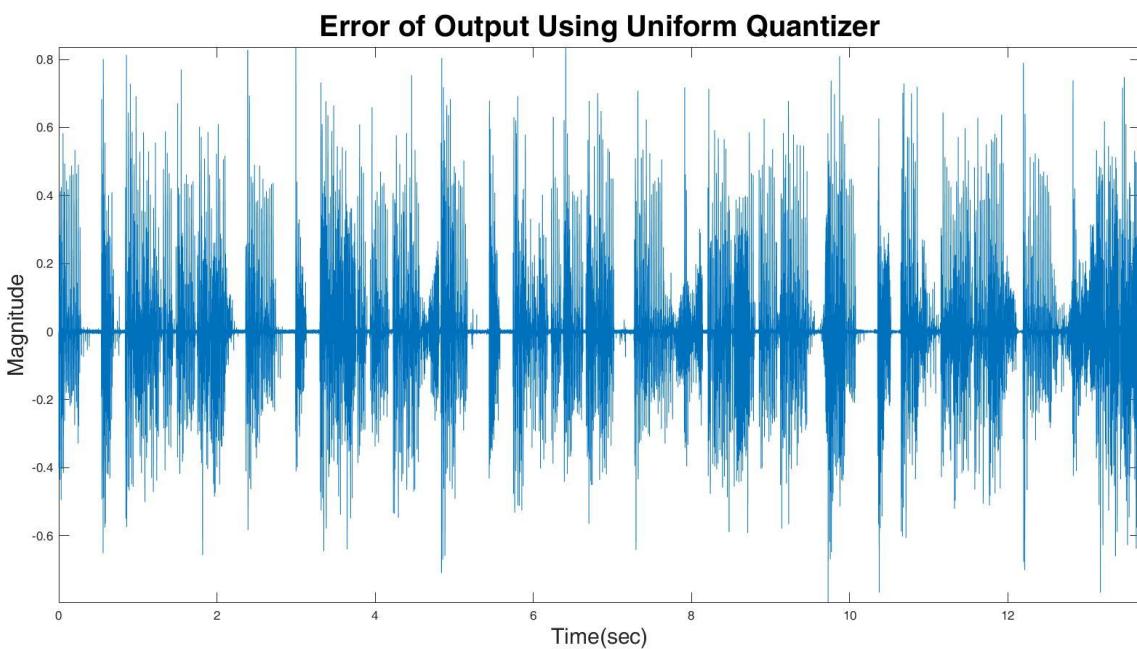
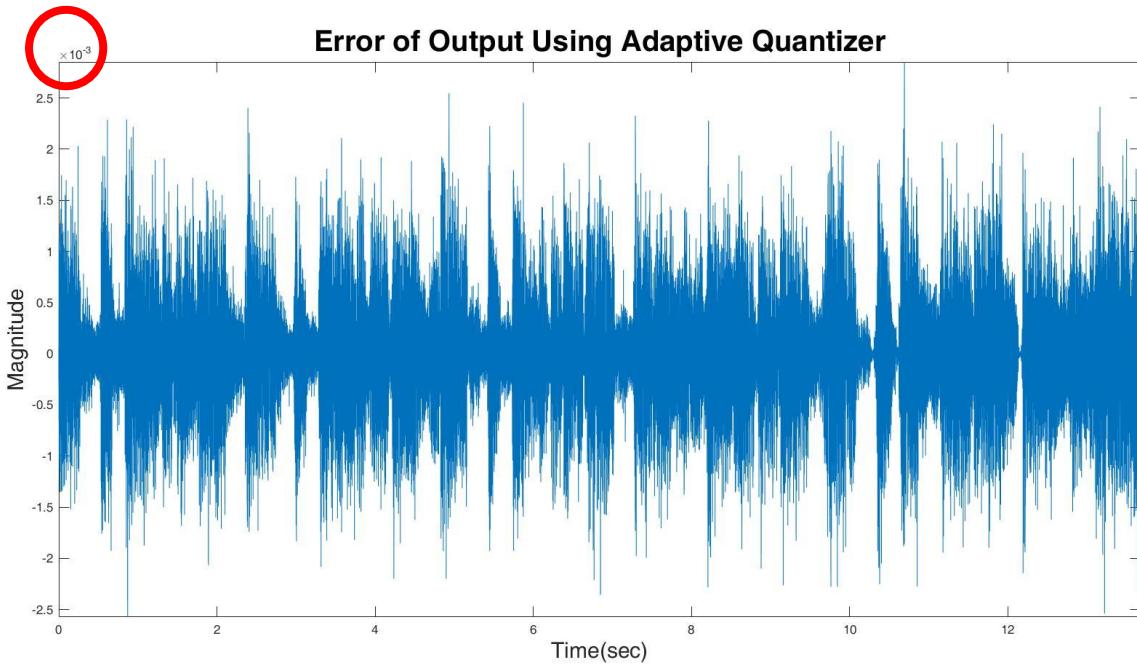


**Output Using Adaptive Quantizer**



**Output Using Uniform Quantizer**





# Βιβλιογραφία

1. T. Painter, A. Spanias, “Perceptual Coding of Digital Audio”, IEEE Proceedings 2000
2. Αρχείο FAQ από mycourses.ntua.gr
3. [https://en.wikipedia.org/wiki/Data\\_compression\\_ratio](https://en.wikipedia.org/wiki/Data_compression_ratio)
4. [https://en.wikipedia.org/wiki/Quantization\\_\(signal\\_processing\)](https://en.wikipedia.org/wiki/Quantization_(signal_processing))