

Συστήματα Πολυμέσων και Εικονική Πραγματικότητα

Εργασία 2016-2017

Σάββας Σαμπαζιώτης 7974

sampazio@ece.auth.gr

Εισαγωγή

Το κείμενο αυτό αποτελεί την αναφορά για την εργασία στα Συστήματα Πολυμέσων του ακαδημαϊκού έτους 2016-2017. Παρακάτω εμπεριέχονται συνοπτικές επεξηγήσεις πάνω στην λειτουργία του παραδοτέου κώδικα και σχόλια πάνω στα αποτελέσματα. Η εργασία υλοποιήθηκε πλήρως, και σε κάθε μία από τις παρακάτω ενότητες, αναλύεται ξεχωριστά η κάθε βαθμίδα του συστήματος και τα αποτελέσματά της.

Ο παραδοτέος κώδικας ακολουθά τις προδιαγραφές της εργασίας. Η κάθε βαθμίδα είναι στον φάκελο «level 1», «level 2» και «level 3» αντίστοιχα, με όλες σχετικές τι απαιτούμενες συναρτήσεις. Οι συναρτήσεις που υλοποιήθηκαν είναι αρκετά περισσότερες σε πλήθος από αυτές που περιγράφονται στην εκφώνηση της εργασίας, διευκολύνοντας την ανάπτυξη και την επαλήθευση του κώδικα.

Οι φάκελοι που εμπεριέχονται είναι οι εξής:

Φάκελος	Περιγραφή
level 1	Αρχεία κώδικα 1 ^{ου} επιπέδου
level 2	Αρχεία κώδικα 2 ^{ου} επιπέδου
level 3	Αρχεία κώδικα 3 ^{ου} επιπέδου
Exports	Αρχεία ήχου που παράχθηκαν κατά την ανάπτυξη της εργασίας
Songs	Αρχεία ήχου που χρησιμοποιήθηκαν κατά την ανάπτυξη της εργασίας
AACoder3 Saved data	Μερικά αποθηκευμένα αποτελέσματα του 3 ^{ου} επιπέδου, για την διευκόλυνση στην εξέταση των αποτελεσμάτων
generateFigures.m	Απλό m-script για την παραγωγή των περισσότερων από τα παρακάτω γραφήματα της αναφοράς αυτής

Αξίζει να σημειωθεί σε αυτό το σημείο πως η διεκπεραίωση της εργασίας, έγινε από έναν μόνο φοιτητή, και όχι από ομάδα των δύο, λόγω επιπλέον πανεπιστημιακών υποχρεώσεων που παρεμπόδισαν την τακτική παρακολούθηση του μαθήματος και την έγκαιρη εύρεση μέλους για συμμετοχή.

Επίπεδο 1^ο

Παραδοτέος Κώδικας και Υλοποίηση

Στο πρώτο επίπεδο υλοποιήθηκαν κανονικά οι συναρτήσεις σύμφωνα με τις προδιαγραφές της εκφώνησης

- SSC
- filterBank
- iFilterBank
- demoAAC1

Οι συναρτήσεις AACoder1 και iAACored1 όμως ως προς την εξής λεπτομέρεια:

```
[ AACSeq1, metadata ] = AACoder1( fNameIn )  
[ x ] = iAACoder1( AACSeq1, fNameOut, metadata )
```

Η παράμετρος metadata δεν είναι τίποτα παραπάνω, από μία δομή που εμπεριέχει το **zero padding** που εφαρμόστηκε στο κομμάτι ήχου και την συχνότητα δειγματοληψίας του –άσχετα από το γεγονός ότι ο κωδικοποιητής και αποκωδικοποιητής υποστηρίζουν μόνο σήματα στα 48kHz δειγματοληψία.

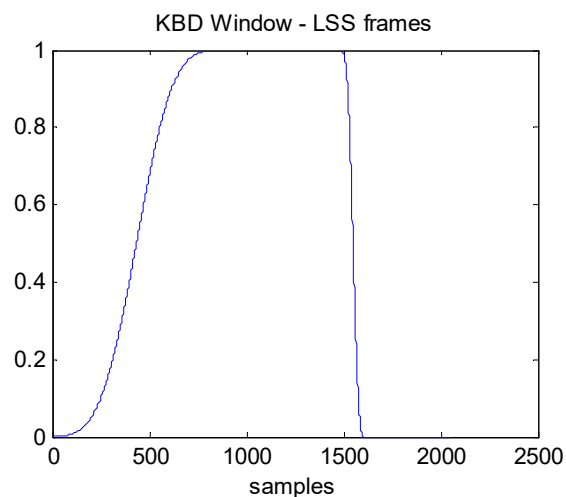
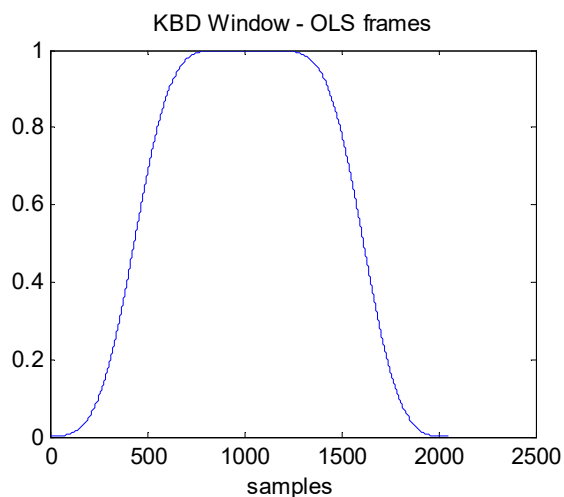
Ο αποκωδικοποιητής, αφαιρεί τα περίσσεια μηδενικά σύμφωνα με την metadata παράμετρο της συνάρτησης και το σήμα x που επιστρέφει έχει ακριβώς το ίδιο πλήθος και την ίδια σχετική θέση δειγμάτων με το αρχικό, επιτρέποντας απευθείας υπολογισμό του σήματος σφάλματος $e = \hat{x} - x$ και κατά επέκταση του SNR του τελικού αποτελέσματος.

Σχετικά με την επιλογή παραθύρου για τον MDCT μετασχηματισμό, στην συνάρτηση AACode1.m δηλώνεται ως σταθερά ο τύπος του παραθύρου που χρησιμοποιείται στον κωδικοποιητή. Ο αναγνώστης μπορεί με ευκολία να ανατρέξει στην πηγή του κώδικα και να επιλέξει αν θα γίνει χρήση του KBD ή του ημιτονοειδούς παραθύρου

Το 1^ο επίπεδο περιλαμβάνει τις παρακάτω επιπλέον συναρτήσεις

Συνάρτηση	Σύνοψη
$y = \text{mdct4}(x)$ $y = \text{imdct4}(x)$	Οι συναρτήσεις αυτές αποτελούν την γρήγορη υλοποίηση του μετασχηματισμού MDCT, της σελίδας http://www.ee.columbia.edu/~marios
$[W_L, W_R] = \text{SineWindow}(N)$	Η συνάρτηση αυτή επιστρέφει δεξί και αριστερό μέρος ημιτονοειδούς παραθύρου
$[WL, WR] = \text{KBDWindow}(N,a)$	Η συνάρτηση αυτή επιστρέφει δεξί και αριστερό μέρος Kaiser Bessel Derived παραθύρου

Οι συναρτήσεις των παραθύρων αυτών σχεδιάστηκαν έτσι ώστε να διευκολύνουν την υλοποίηση των σύνθετων παραθύρων κατά την εφαρμογή του MDCT και iMDCT στις συναρτήσεις filterbank και iFilterbank αντίστοιχα. Παρακάτω φαίνεται ένα δείγμα των παραθύρων αυτών (*demo.m*)



Επίσης, η συνάρτηση **SSC**, εμπεριέχει τοπικά πλήθος ιδιωτικών συναρτήσεων, καθιστώντας την υλοποίησή της πιο ευκολονόητη και συνοπτική.

Συνάρτηση	Σύνοψη
frameType = SSCsinglechannel(frameT, nextFrameT, prevFrameType)	Υλοποιεί το Segmentation σε ένα κανάλι μόνο, όπως ορίζει το πρότυπο. Ο τελικός τύπος του κάθε frame και των 2 καναλιών, προσδιορίζεται στην SSC
result = framelsESH(frame)	Προσδιορίζει αν το frame είναι τύπου ESH, βάσει τις ενεργειακής ανάλυσης που περιγράφει το πρότυπο
Index = S2Index(sequenceType)	Βοηθητική συνάρτηση για την χρήση switch-case στην SSCsingleChannel και SSC διαδικασία

Αποτελέσματα

Η λειτουργία του πρώτου επιπέδου δεν περιλαμβάνει καμίας μορφής απώλεια στο σήμα. Αντίθετα, όλοι οι μετασχηματισμοί είναι απόλυτα αντιστρέψιμοι, και οποιαδήποτε απώλεια προκύπτει, οφείλεται στην πεπερασμένη ακρίβεια του υπολογιστή στις πράξεις με πραγματικούς αριθμούς.

LicorDeCalandraca.wav (5 sec)	SIN	KBD
Coding Time	0.60753 sec	8.653 sec
Decoding Time	0.34721 sec	8.1328 sec
SNR channel 1	307.7758 dB	301.5971 dB
SNR channel 2	307.8630 dB	301.5680 dB

Η συνολική διαδικασία είναι εμφανώς πιο αργή στην περίπτωση των KBD παραθύρων. Το φαινόμενο αυτό ήταν αναμενόμενο, αλλά όχι σε τέτοιο βαθμό, καθώς η πολυπλοκότητα στην παραγωγή του KBD παραθύρου είναι σαφώς μεγαλύτερη σε σχέση με το ημιτονοειδές. Αυτό μπορεί να προφανώς να βελτιστοποιηθεί, ακόμα και σε ένα περιβάλλον όπως στο matlab (χρησιμοποιώντας προϋπολογισμένα παράθυρα ή συντελεστές Kaiser) , ωστόσο, δεν κρίθηκε αναγκαίο στα πλαίσια της εργασίας μία τέτοια εφαρμογή .

Όσο αφορά το SNR, η τάξη μεγέθους του αποτελέσματος, υποδεικνύει τέλεια ανακατασκευή. Το ακουστικό αποτέλεσμα όντως επαληθεύει την παραπάνω πρόταση.

Επίπεδο 2°

Παραδοτέος Κώδικας και Υλοποίηση

Στο δεύτερο επίπεδο υλοποιήθηκε η βαθμίδα TNS. Ο κώδικας που υλοποιήθηκε περιλαμβάνει όλες τις συναρτήσεις του πρώτου επιπέδου όπως ακριβώς περιγράφηκαν παραπάνω. Οι συναρτήσεις

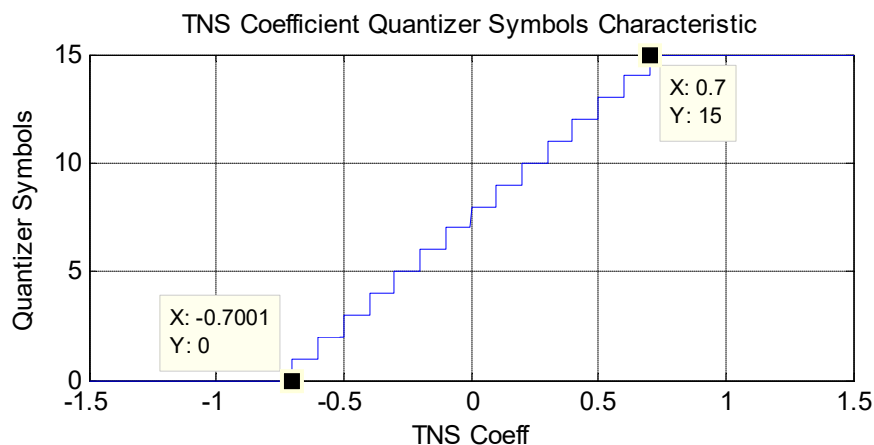
- TNS
- iTNS

υλοποιούνται όπως ορίζονται στην εκφώνηση. Ταυτόχρονα, υπάρχουν και οι επιπλέον συναρτήσεις

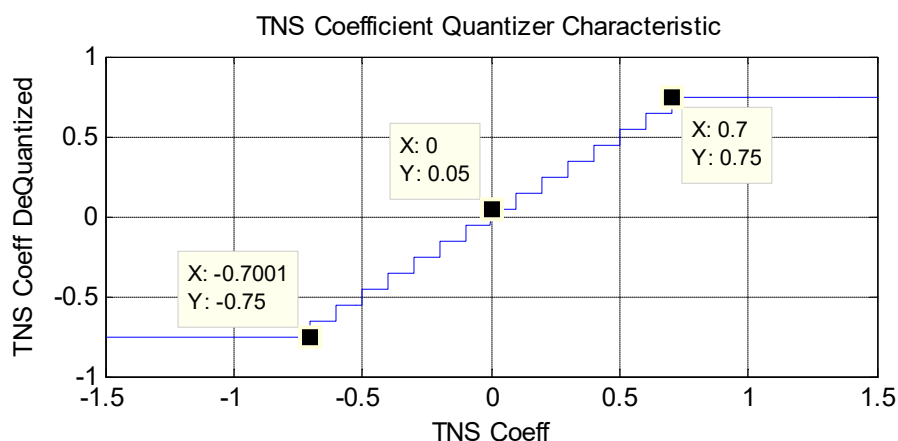
- `s = TNSQuantizer(a)`
- `a_bar = TNSDequantizer(s)`

Η TNSDequantizer υλοποιεί τον αποκβαντιστή, όπου αντιστοιχίζει τα σύμβολα σε πραγματικούς αριθμούς.

Η TNSQuantizer υλοποιεί τον ομοιόμορφο συμμετρικό κβαντιστή-4bit των συντελεστών του TNS φίλτρου. Η έξοδος της είναι σύνολο συμβόλων αποτελούμενο από τους ακέραιους το 0 έως 15. Εφόσον το βήμα κβαντισμού είναι 0.1, το εύρος του αποκβαντιστή θα είναι $R = 16 * 0.1 = 1.6$. Επίσης, ο κβαντιστής είναι ομοιόμορφος και συμμετρικός, με αποτέλεσμα οι τιμές να «μοιράζονται» ομοιόμορφα γύρω από το 0. Συγκεκριμένα, τα σύμβολα, αντιστοιχίζονται στις στάθμες του κβαντιστή ως εξής:



Ενώ οι τιμές του αποκβαντισμένου σήματος σε σχέση με αρχικό φαίνονται στην παρακάτω χαρακτηριστική



Κατά την διάρκεια εκτέλεσης του κωδικοποιητή, γίνεται έλεγχος για την ευστάθεια του $H_{TNS}^{-1}(z)$ φίλτρου, χωρίς να βρεθεί κάποιος ασταθής συνδυασμός συντελεστών. Η λειτουργία αυτή είναι ενεργοποιημένη και όπως φαίνεται παρακάτω επηρεάζει σε σημαντικό βαθμό το χρόνο εκτέλεσης του κώδικα.

ΠΡΟΒΛΗΜΑ: Σε περίπτωση που η διαδικασία χρειαστεί να αντιμετωπίσει κάποιο παράθυρο με όλα τα δείγματα να είναι μηδενικά (αναφερόμενοι συνήθως στην αρχή και στο τέλος του κομματιού ήχου), η επίλυση της Wiener-Hopf εξίσωσης για τον προσδιορισμό των συντελεστών των TNS θα έχει ως αποτέλεσμα μη-πραγματικό αριθμό. Η περίπτωση αυτή **δεν** έχει αντιμετωπιστεί από τον κώδικα, και μονολότι η βαθμίδα του TNS δεν εμφανίζει πρόβλημα με αυτά τα ιδιάζων σημεία, η κωδικοποίηση Huffman θα εμφανίσει πρόβλημα. Η συνάρτηση του TNS προβάλλει σχετικό μήνυμα ώστε να αντιμετωπιστεί η περίπτωση αυτή

Αποτελέσματα

Η λειτουργία του δευτέρου επιπέδου δεν περιλαμβάνει ακόμα απώλεια πληροφορίας στο σήμα. Όλοι οι μετασχηματισμοί είναι απόλυτα αντιστρέψιμοι, και η βαθμίδα TNS-iTNS απλά αποτελεί την εφαρμογή ενός FIR φίλτρου και του αντιστρόφου του.

Η βαθμίδα δοκιμάστηκε με τα εξής κομμάτια ήχου

LicorDeCalandraca.wav (5 sec)	SIN	KBD
Coding Time	9.0077 sec	8.7182 sec
Decoding Time	6.3044 sec	6.2711 sec
SNR channel 1	301.6353 dB	301.6353 dB
SNR channel 2	301.5913 dB	301.5913 dB

DoctoWho.wav (35 sec)	SIN	KBD
Coding Time	17.1215 sec	54.4138 sec
Decoding Time	1.7522 sec	38.9608 sec
SNR channel 1	307.6117 dB	301.1952 dB
SNR channel 2	307.6343 dB	301.1943 dB

Στην περίπτωση των SIN παραθύρων, το SNR έχει πέσει ελάχιστα, ίσως λόγω του πλήθους των επιπλέον υπολογισμών που εφαρμόζονται πάνω στα δεδομένα. Παρόλα αυτά, τα KBD παράθυρα έχουν εμφανίσει μεγάλη ευρωστία σε σχέση με τα SIN παράθυρα σε ότι αφορά αυτό το φαινόμενο. Σε κάθε περίπτωση το σήμα ανακατασκευάζεται σχεδόν τέλεια, χωρίς καμία απώλεια. Τα παραπάνω κομμάτια εμπεριέχονται στον φάκελο **songs**, σε περίπτωση που ο αναγνώστης επιθυμεί να επιθεωρήσει την λειτουργία του κώδικα και τα αποτελέσματά του.

Επίπεδο 3^ο

Παραδοτέος Κώδικας και Υλοποίηση

Στην Τρίτη βαθμίδα ζητήθηκε να υλοποιηθεί το ψυχοακουστικό μοντέλο βάσει του οποίου ρυθμίζεται ο κβαντιστής του κωδικοποιητή. Υλοποιήθηκαν όλες οι συναρτήσεις της εκφώνησης όπως σύμφωνα με τις προδιαγραφές που δίνονται. Βασικά σημεία που χρειάζονται προσοχή για την εκτέλεση του κώδικα είναι τα εξής:

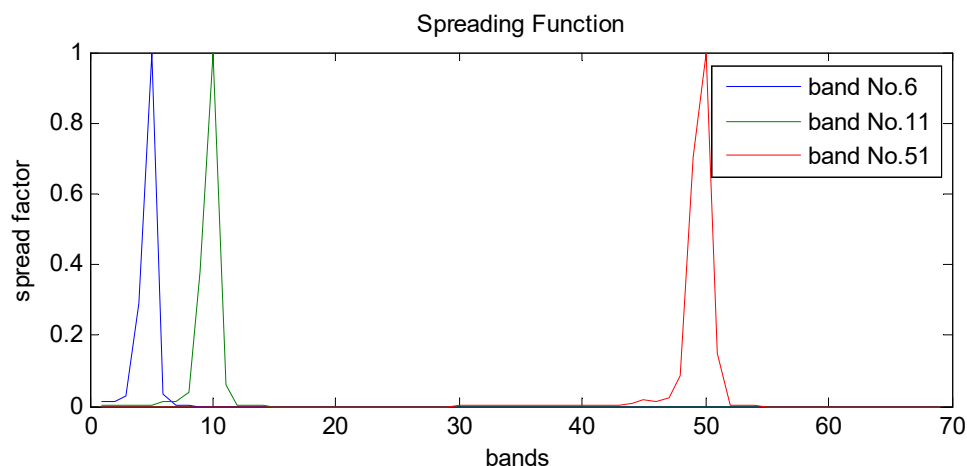
- Οι *AACoder3* και *iAACoder3*, έχουν ωστόσο την τροποποίηση με την παράμετρο *metadata* όπως περιγράφεται στο 1^ο επίπεδο.
- Το script *generateSpreadMatrix* καλεί την *spreadingfun* και κατασκευάζει πίνακα με προϋπολογισμένα αποτελέσματα της συνάρτησης διασποράς για κάθε ζεύγος μπάντας. Ο πίνακας αυτός αποθηκεύεται ως *TableSpread.mat* και εμπεριέχεται στον φάκελο *level3*
- Οι δοσμένες συναρτήσεις Huffman εμπεριέχονται στον φάκελο *huffman* εντός του φακέλου *level3*. Για την σωστή εκτέλεση των αρχείων χρειάζεται να προστεθεί ο φάκελος αυτός στη λίστα με τις διαδρομές αρχείων του matlab.

Οι επιπλέον συναρτήσεις που υλοποιήθηκαν είναι οι

Συνάρτηση	Σύνοψη
[x] = spreadingfun(i, j, B219)	Υλοποιεί την συνάρτηση διασποράς κάθε μπάντας σύμφωνα με τις τιμές.
generateSpreadMatrix	Παράγει τιμές τις συνάρτησης διασποράς και τις αποθηκεύει σε πίνακα, οποίος αποθηκεύεται στο <i>TableSpread.mat</i>
S = quant(X, a, B219)	Ο κβαντιστής που χρησιμοποιείται από τον κωδικοποιητή σύμφωνα με τις περιγραφές του προτύπου. Η παράμετρος <i>a</i> είναι το <i>scale factor gain</i>
X_ = iQuant(S, a, B219)	Ο αποκβαντιστής σύμφωνα με τις περιγραφές του προτύπου.

Συνάρτηση διασποράς

Οι προϋπολογισμένες τιμές της *spreadingFun* αποθηκεύονται στους πίνακες *SFa* και *SFb* του αρχείου *TableSpread.mat*. Οι δύο αυτοί πίνακες έρχονται σε αντιστοιχία με τους πίνακες *B219a* και *B219b* από του πρότυπο. Μία απεικόνιση της διασποράς της κάθε μπάντας είναι η παρακάτω:

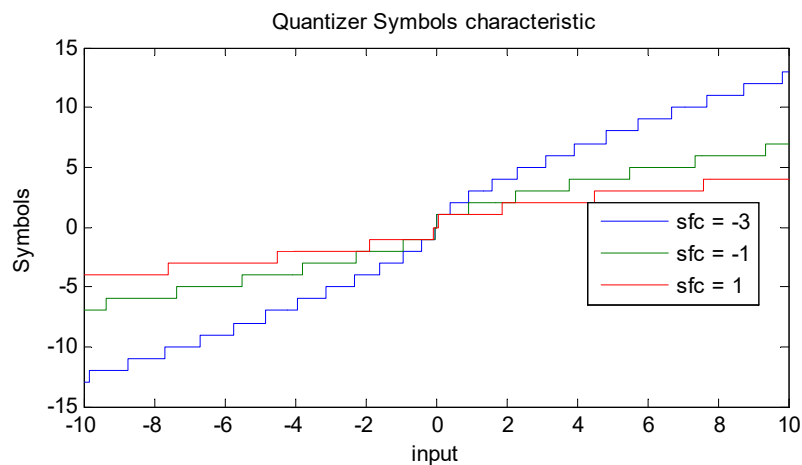


Κβαντιστής

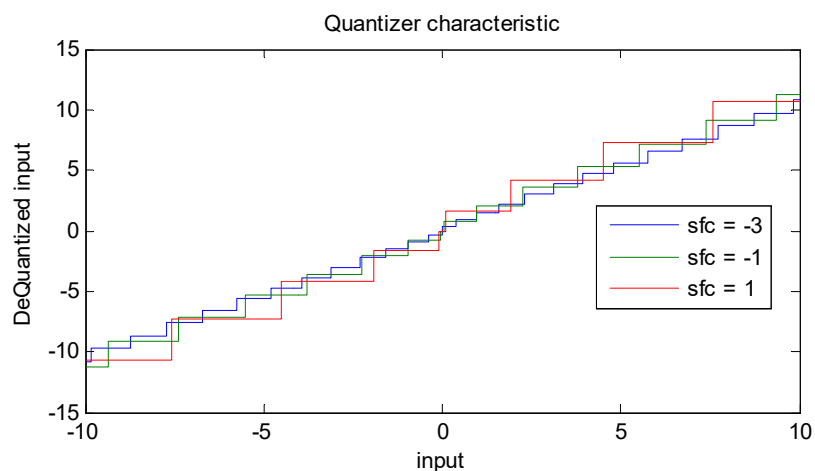
Η διαδικασία κβαντισμού υλοποιείται στην συνάρτηση AACQuantizer. Η συνάρτηση αυτή εμπεριέχει εσωτερικά τις εξής συναρτήσεις:

Συνάρτηση	Σύνοψη
<code>[S, sfc, G, T] = AACQuantizerSingleFrame(frameF, SMR, B219)</code>	Κβαντίζει ένα παράθυρο σήματος. Στην περίπτωση των LONG_WINDOWS καλείται μία φορά, και στην περίπτωση των EIGHT_SHORT καλείται 8 φορές
<code>Pe = quantError(X, a, b B219)</code>	Βάσει του scale factor gain, κβαντίζει και αποκβαντίζει το σήμα, και επιστρέφει την ισχύ του σφάλματος για την ακουστική μπάντα με δείκτη b
<code>sfc = optimizeSFC(frameF, a, T, B219)</code>	Μεταβάλλει τα scale factor gains, με κατάλληλο τρόπο όπως περιγράφεται στην εργασία και στο πρότυπο. Περισσότερες λεπτομέρειες παρακάτω

Η κατανομή των συμβόλων του κβαντιστή φαίνεται στο παρακάτω διάγραμμα, για μερικές ενδεικτικές τιμές του scale factor gain. Η μη-γραμμικότητα είναι εμφανής, όπως και η μεταβολή του βήματος κβαντισμού



Η τελική χαρακτηριστική της συνολικής διαδικασίας κβαντισμού παρουσιάζεται παρακάτω. Παρατηρούμε ως η μη γραμμικότητα αντισταθμίζεται, και διατηρείται απλώς η ανομοιομορφία του βήματος κβάντισης.



Βελτιστοποίηση των Scale Factor Gains

Σε ότι αφορά την βελτιστοποίηση του κβαντιστή, τα βασικά βήματα του αλγορίθμου επεξηγούνται παρακάτω σε φυσική γλώσσα:

1. Ο αλγόριθμος ξεκινάει με μία αρχική τιμή για κάθε έναν από τους συντελεστές sfc , η οποία αντιστοιχεί σε ένα επαρκώς μικρό βήμα κβάντισης.
2. Υπολογίζεται η ισχύς του σφάλματος κβαντισμού P_e
3. Για κάθε δείκτης μπάντας b , ελέγχεται αν ισχύουν οι παρακάτω συνθήκες:
 - a. $P_e(b) < T(b)$ η οποία εξασφαλίζει πως το σφάλμα κβαντισμού δεν θα είναι **ακουστό**
 - b. $|sfc(b) - sfc(b + 1)| < 60$ που εξασφαλίζει το πλήθος των bit για κάθε sfc μετά την **DPCM κωδικοποίηση**
4. Αν οι παραπάνω συνθήκες ισχύουν, τότε ο συντελεστής μπορεί να αυξηθεί παραπάνω οδηγώντας σε μεγαλύτερο σφάλμα κβαντισμού, άρα και σε καλύτερη συμπίεση του σήματος, διατηρώντας τις απώλειες μη-αντιληπτές από το ανθρώπινο αυτί
5. Η διαδικασία επαναλαμβάνεται μέχρις ότου να ικανοποιείται οριακά οποιαδήποτε από τις δύο συνθήκες

Σχετικά με την DPCM κωδικοποίηση, η συνθήκη εξασφαλίζει τα εξής: Κάθε $d(n) = sfc(n + 1) - sfc(n)$ δείγμα θα έχει εύρος $R = 60 - (-60) = 120$. Ως αποτέλεσμα, **7 bit** θα είναι επαρκή για την ασυμπίεστη κωδικοποίηση του κάθε συντελεστή, επιτρέποντας την ομαλή λειτουργία της κωδικοποίησης Huffman, όπως ορίζεται στο πρότυπο. Η ρύθμιση των sfc , χωρίς περιορισμούς, θα οδηγήσει σε ακραίες διαφορές, τόσο σε μέγεθος όσο και σε «ποικιλία», αυξάνοντας την εντροπία του συστήματος και μειώνοντας την απόδοση της κωδικοποίησης Huffman.

Ομολογουμένως, η υλοποίηση του αλγορίθμου δεν ήταν εύκολο να υλοποιηθεί απολύτως σωστά, ούτε να βρεθεί κάποια απόδειξη πως η λύση της ήταν όντως η βέλτιστη σύμφωνα με τις παραπάνω συνθήκες. Η υλοποίηση της συνάρτησης *optimizeSFC* περιλαμβάνει τα εξής βήματα:

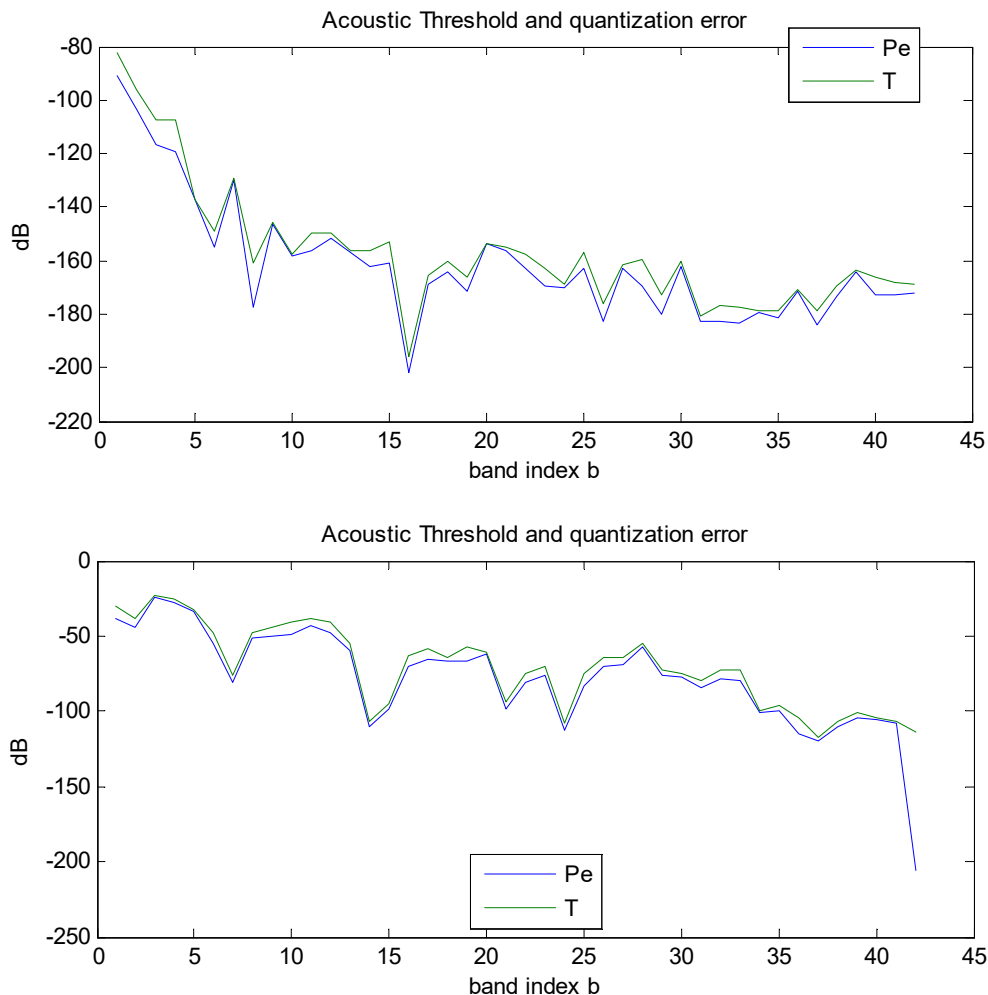
Αλγόριθμος *optimizeSFC*(X, sfc)

1. Υπολογίζονται οι βέλτιστοι συντελεστές SFC σύμφωνα με την συνθήκη 3.a της παραπάνω περιγραφής του αλγορίθμου
2. **ΓΙΑ ΚΑΘΕ** sfc
 - a. **ΑΝ** $sfc(b + 1) - sfc(b) > 60$ **TOTE** Μείωσε τον $sfc(b+1)$
 - b. **ΑΛΛΙΩΣ ΑΝ** $sfc(b + 1) - sfc(b) < -60$ **TOTE** Μείωσε τον $sfc(b)$
3. **ΑΝ ΓΙΑ ΚΑΘΕ** sfc ισχύει $|sfc(b + 1) - sfc(b)| < 60$ **TOTE** ΤΕΛΟΣ
4. **ΑΛΛΙΩΣ** GOTO 2

Η βασική ιδέα του αλγορίθμου αυτού είναι η σταδιακή και ομοιόμορφη μείωση των συντελεστών sfc , ώσπου να ικανοποιηθεί η συνθήκη 3.b. Η διαδικασία δεν είναι βέλτιστη από άποψη πολυπλοκότητας, αλλά μπορεί να εξασφαλίσει την ικανοποίηση και των δύο συνθηκών. Η μείωση των κερδών sfc , ελαττώνει το σφάλμα κβαντισμού, με αποτέλεσμα να εξασφαλίζεται η συνθήκη 3.a

Μερικές βελτιώσεις που μπορούν να εφαρμοστούν, είναι η **δυναμική αναζήτηση**, κατά τον υπολογισμό των βέλτιστων συντελεστών SFC στο βήμα 1 του *optimizeSFC*. Ωστόσο, η εφαρμογή αυτή κρίθηκε περιττή στα πλαίσια της εργασίας, και δεν υλοποιήθηκε.

Σχόλια σχετικά με την απόδοση της διαδικασίας ως προς το SNR και το compression που επιτυγχάνεται, εμπεριέχονται στην παρακάτω υποενότητα με τα αποτελέσματα του αλγορίθμου. Μία γραφική απεικόνιση του τελικού σφάλματος κβαντισμού σε σχέση με το κατώφλι ακουστότητας παρουσιάζεται στα παρακάτω γραφήματα σε dB.



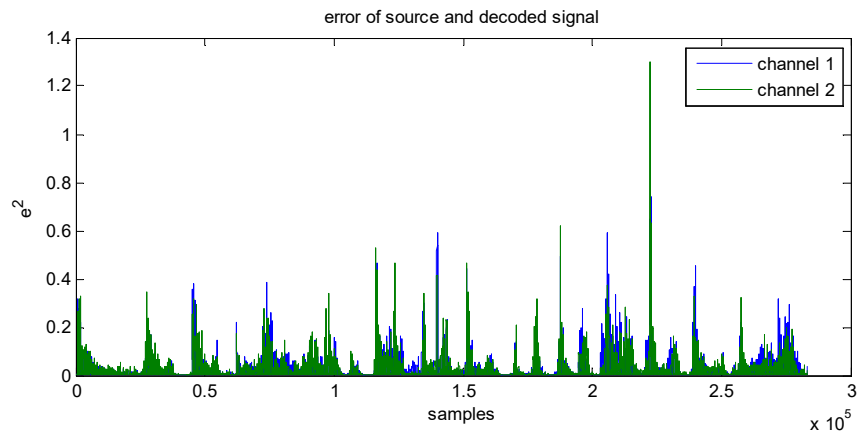
Τα παραπάνω γραφήματα αποτελούν κάποια τυχαία frames του σήματος. Το κάθε γράφημα παράγεται από στην **συνάρτηση AACQuantizer.m στην γραμμή 52, η οποία έχει σχολιαστεί**, για την ομαλή εκτέλεση του κώδικα.

Αποτελέσματα

Σε αυτό το στάδιο υλοποίησης, υπάρχει πλέον απώλεια πληροφορίας στο σήμα. Παρακάτω φαίνεται η συνολική απόδοση του συστήματος.

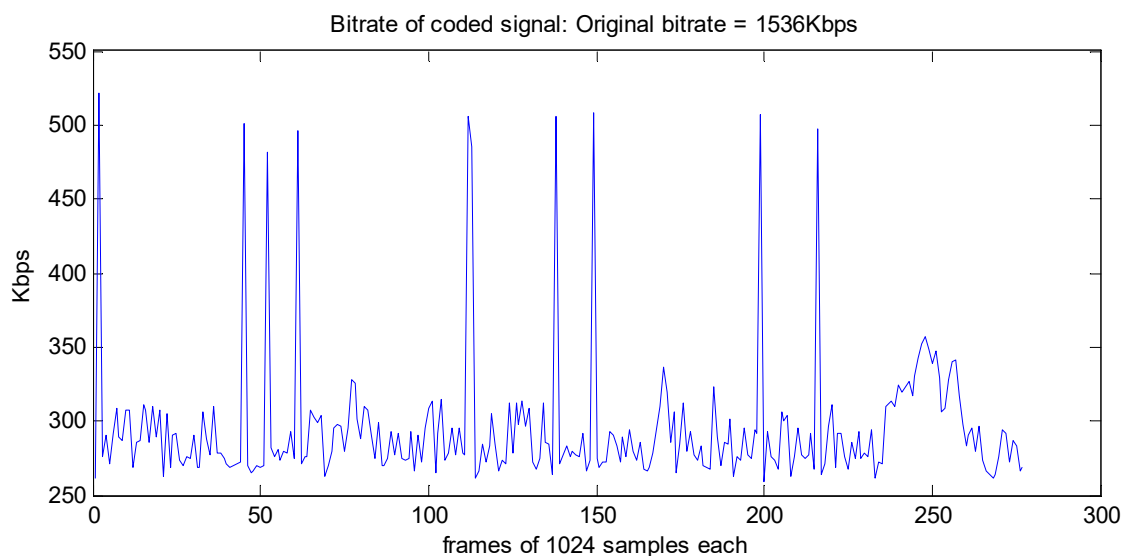
LicorDeCalandraca.wav (5 sec)	SIN	KBD
Coding Time	652.7693 sec	730.7016 sec
Decoding Time	235.8093 sec	277.3157 sec
SNR channel 1	6.5267 dB	7.7916 dB
SNR channel 2	6.5166 dB	7.7798 dB
Mean(Bitrate)	295.18 Kbps	294.25 Kbps
Compression Ratio	x5.2036	x5.2200

Παρατηρούμε πως τα KBD παράθυρα, επιτύγχανε σχετικά καλύτερο SNR από τα SIN παράθυρα, με ελαφρώς καλύτερη συμπίεση. Σε κάθε περίπτωση, η τάξη μεγέθους του SNR υποδηλώνει πως το σήμα διαφέρει δραματικά από το αρχικό. Με μία γρήγορη επισκόπηση του σφάλματος μεταξύ των δύο σημάτων παρατηρούμε το εξής:



Εμφανίζονται διάφορες «κορυφές» στο τετραγωνικό σφάλμα η οποίες αναμένονται να μειώνουν το SNR αρκετά, παρόλα αυτά η τάξη μεγέθους των υπόλοιπων δειγμάτων δεν φαίνεται ιδιαίτερη αισιόδοξη. Έπειτα, όμως από αναπαραγωγή των αποκωδικοποιημένου κομματιού ήχου, η ποιότητα του ήχου ήταν αρκετά ικανοποιητική. Ένας ευαίσθητος ακροατής αναμένεται να προσέξει μερικές μικρές διαφορές, η οποίες περισσότερο αποδίδονται στην απλοποίηση του κωδικοποιητή ως προς την επιλογή παραθύρων και το segmentation του σήματος. Υπενθυμίζεται πως ο αλγόριθμος βελτιστοποίησης των συντελεστών κβάντισης, είναι κάθε άλλο παρά «επιθετικός», διατηρώντας κάποια περιττή ακρίβεια στην κβάντιση σε αντάλλαγμα με μικρότερο λόγο συμπίεσης.

Σε ότι αφορά το ρυθμό δεδομένων του κωδικοποιημένου σήματος, υπενθυμίζεται πως το μέσο bitrate που καταγράφεται στον παραπάνω πίνακα, απέχει από το πραγματικό bitrate του σήματος. Η παρακάτω απεικόνιση αποκαλύπτει το εξής:



Το μέσο bitrate μπορεί να 5 φορές μικρότερο του πρωτότυπου bitrate, ωστόσο, παρατηρούμε διάφορες περιστασιακές κορυφώσεις στο bitrate του κάθε frame του σήματος που φτάνουν έως και τα 500Kbps. Σε αυτό το σημείο θα αναφερθούμε σε ένα στοιχείο του κωδικοποιητή που δεν έχει αναφερθεί μέχρι τώρα: τα

ESH frames. Οι παραπάνω κορυφές αντιστοιχούν στην ουσία στα 9 ESH frames που εντοπίζονται στο συγκεκριμένο σήμα. Σε περίπτωση που η κωδικοποίηση και η μετάδοση του σήματος γίνεται σε πραγματικό χρόνο και απαιτείται σταθερό bitrate (CBR), τα ESH frames θα χρειαστούν ίσως ειδικό χειρισμό.

Αποτελέσματα του ενός ακόμα κομματιού ήχου είναι το παρακάτω:

DoctorWho.wav (35 sec)	SIN	KBD
Coding Time	100 min	-
Decoding Time	25 min	-
SNR channel 1	9.1110 dB	-
SNR channel 2	8.8679 dB	-
Mean(Bitrate)	288.80 Kbps	-
Compression Ratio	x5.3186	-

Το τραγούδι εμφάνιζε κάποια ιδιαιτερότητα στον φασματικό του περιεχόμενο, λόγω ίσως της επιλογής των μουσικών οργάνων και της μεγάλης ποικιλομορφίας του. Ως αποτέλεσμα, επιτεύχθηκε αρκετά υψηλό SNR με μεγαλύτερη συμπίεση, σε σχέση με το κομμάτι LicorDeCalandraca.wav

Τα παραχθέντα αρχεία ήχου βρίσκονται στον φάκελο *exports*, με το όνομα

- LicorDeCalandraca_3_SIN.wav
- LicorDeCalandraca_3_KBD.wav
- DoctorWho_3_SIN.wav

Σε περίπτωση που αναγνώστης επιθυμεί να εξετάσει ο ίδιος τα αποτελέσματα. Επίσης, στον φάκελο *AACoder3 Saved Data*, εμπεριέχονται τα αρχεία

- AACSeq3_SIN.mat
- AACSeq3_KBD.mat
- AACSeq3_DOCTORWHO_SIN.mat

Τα οποία εμπεριέχουν τις όλες μεταβλητές της συνάρτησης demoAAC3.m σε περίπτωση που ο αναγνώστης επιθυμεί να εξετάσει το συμπίεσμένο αποτέλεσμα, το bitrate και τις λοιπές μεταβλητές του συστήματος.