

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΜΜΕ

Υλοποίηση σταδίων κωδικοποίησης ηχητικών σημάτων στο προγραμματιστικό περιβάλλον Matlab

Χριστοπούλου Ζωή

Σακκά Ελισάβετ

Επιβλέπων Καθηγητής: Κούτρας Αθανάσιος

Πύργος, 21/12/2015

Contents

Εικόνες.....	4
Πίνακες.....	6
Ευχαριστίες.....	7
Περίληψη.....	8
Abstract	8
1 «Ήχος».....	9
1.1 Ορισμός και θεωρία του ήχου	9
1.2 Βασικές Αρχές Θεωρίας Ήχου.....	11
1.3 ΕνότηταΧαρακτηριστικά Ψηφιακού Ήχου	14
2 «Διαδικασία Ψηφιοποίησης Ήχου».....	16
2.1 Στάδια Ψηφιοποίησης Ηχητικών Αρχείων	16
2.2 Δειγματοληψία.....	19
2.3 Προβλήματα Δειγματοληψίας.....	22
2.4 Κβαντισμός.....	24
3 «Κωδικοποίηση Ψηφιακού Σήματος»	27
3.1 Μέθοδοι Κωδικοποίησης	27
3.2 PCM (PulseCodeModulation).....	27
3.3 Mu-Law PCM και A-Law PCM.	29
3.4 DPCM (Differential Pulse Code Modulation).....	30
3.5 ADPCM (Adaptive Differential Pulse Code Modulation).....	32
4. «Συμπύεση».....	34
4.1 Συμπύεση Ψηφιακού Ήχου	34
4.2 Πρότυπο MPEG.....	37
4.3MP3	40
4.4 Άλλα πρότυπα.....	46

4.4.1 WMA (Windows Media Audio)	47
4.4.2 MP3 Pro	47
4.4.3 MP3 Surround.....	48
4.4.4 AAC (Advance Audio Coding)	48
4.4.5 RA-Real Audio	49
4.4.6 OGG Vorbis	49
4.4.7 AC3 Dolby Digital	50
5. Περιβάλλον προγραμματισμού Matlab	51
5.1 Εισαγωγή στην Matlab	51
5.2 PCM κωδικοποίηση - Παλμοκωδική Διαμόρφωση.....	52
5.2.1 Ομοιόμορφος κβαντιστής.....	54
5.2.2 Μη ομοιόμορφο PCM.....	56
5.3 Πειραματική Αξιολόγηση	65
5.3.1 Με βάση το SQNR	66
5.3.2 Με βάση το ακουστικό αποτέλεσμα.....	69
5.3.3 Με βάση τις κυματομορφές εξόδου	70
5.4 Υλοποίηση σε Matlab	76
5.5 Κωδικοποίηση πηγής κατά Huffman.....	81
6. Παράρτημα.....	84
Αναφορές	91

Εικόνες

Εικόνα 1-1 Κυματομορφή σήματος με μεταβαλλόμενη ισχύ **Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.**

Εικόνα 1-1 Κυματομορφή σήματος με μεταβαλλόμενη ισχύ.....	11
Εικόνα 1-3 Κυματομορφή περιοδικού σήματος.....	13
Εικόνα 1-2 Κυματομορφή περιοδικού σήματος.....	12
Εικόνα 1-4 Σχέση τόνου και συχνότητας.....	13
Εικόνα 1-5 Διάφοροι κυματισμοί.....	14
Εικόνα 1-6 Παραδείγματα συχνοτικού εύρους.....	14
Εικόνα 2-1 ADC (Analog to digital converter)	18
Εικόνα 2-2 DAC (Digital to analog converter)	18
Εικόνα 2-3 Διαδικασία ψηφιοποίησης ηχητικού σήματος.....	19
Εικόνα 2-4 Α) Αναλογικό σήμα Β) Ψηφιακό με χαμηλή συχνότητα Γ) Ψηφιακό με υψηλή συχνότητα δειγματοληψίας.....	20
Εικόνα 2-5 Διαδικασία μετατροπής αναλογικού σε ψηφιακό και αντίστροφα.....	23
Εικόνα 2-6 Ψαλιδισμός.....	26
Εικόνα 3-1 Κωδικοποίηση PCM	28
Εικόνα 3-2 Σύγκριση των αλγορίθμων μ -law και A-law.....	30
Εικόνα 3-3 Κωδικοποιητής και αποκωδικοποιητής ενός συστήματος DPCM.....	31
Εικόνα 3-4 Κωδικοποιητής ADPCM.....	33
Εικόνα 3-5 Αποκωδικοποιητής ADPCM.....	33
Εικόνα 4-1 Ψυχοακουστικό Μοντέλο Ακοής.....	35
Εικόνα 4-2 Φαινόμενο Ηχητικής Σκίασης.....	36

Εικόνα 4-3 Φαινόμενο επικάλυψης. Ένας δυνατός ήχος, μεταβάλλεται τοπικά στην καμπύλη του κατωφλιού ακουστικότητας επικαλύπτοντας ένα γειτονικό αδύναμο ήχο.....	42
Εικόνα 4-4 Διάγραμμα βαθμίδων κωδικοποιητή MP3.....	43
Εικόνα 4-5 Μαθηματική περιγραφή του μετασχηματισμού MDCT	43
Εικόνα 4-6 Διαδικασία Ripping και Encoding	44
Εικόνα 4-7 Διάγραμμα βαθμίδων ενός MP3.....	46
Εικόνα 4-8 Βαθμίδες του κωδικοποιητή OggVorbis. Η κωδικοποίηση του φάσματος βάσης και του απομένοντος φάσματος.....	50
Εικόνα 5-1 Διάγραμμα βαθμίδων ενός συστήματος PCM.... Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.	
Εικόνα 5-2 Διάγραμμα βαθμίδων ενός συστήματος PCM.....	53
Εικόνα 5-3 Διάγραμμα βαθμίδων ενός συστήματος μη ομοιόμορφου PCM	57
Εικόνα 5-4 Γράφημα του μ-συμπίεστη και Α-συμπίεστη	57
Εικόνα 5-5 Μεταβολή SQNR για κβάντιση 2-bit.....	67
Εικόνα 5-6 Μεταβολή SQNR για κβάντιση 4-bit.....	67
Εικόνα 5-7 Κυματομορφή αρχικού σήματος.....	70
Εικόνα 5-8 Κυματομορφή εξόδου για ομοιόμορφη κβάντιση 2-bit	71
Εικόνα 5-9 Κυματομορφή εξόδου για ομοιόμορφη κβάντιση 6-bit	72
Εικόνα 5-10 Κυματομορφή εξόδου για μη ομοιόμορφο PCM 2-bit τύπου-Α	73
Εικόνα 5-11 Κυματομορφή εξόδου για μη ομοιόμορφο PCM 6-bit τύπου-Α	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 5-12 Κυματομορφή εξόδου για μη ομοιόμορφο PCM 4-bit τύπου-Α	74
Εικόνα 5-13 Έξοδος για μη ομοιόμορφο PCM 6-bit τύπου-Α.....	74

Εικόνα 5-14 Κυματομορφή εξόδου για μη ομοιόμορφο κβαντιστή 2-bit
..... 75

Εικόνα 5-15 Κυματομορφή εξόδου για μη ομοιόμορφο κβαντιστή 4-bit
..... 75

Εικόνα 5-16 Κυματομορφή εξόδου για μη ομοιόμορφο κβαντιστή 6-bit. 76

Πίνακες

Πίνακας 2.1 Αυτή είναι η λεζάντα του πίνακα ..**Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.**

Ευχαριστίες

Θα θέλαμε να εκφράσουμε τις ευχαριστίες μας στον επιβλέποντα Επίκουρο καθηγητή Αθανάσιο Κούτρα, για τη δυνατότητα που μας έδωσε να πραγματοποιήσουμε την πτυχιακή μας εργασία ,και για τον πολύτιμο χρόνο που διέθεσε για την περάτωση της παρούσας εργασίας. Οι σημαντικές υποδείξεις και συμβουλές του μας κατεύθυναν σε ένα σωστό τρόπο σκέψης πάνω από όλα και μας προσέφεραν σημαντικά εφόδια για την μετέπειτα ζωή μας.

Επίσης, θα θέλαμε να ευχαριστήσουμε ακόμα, όλους τους καθηγητές του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Δυτικής Ελλάδος, για τις πολύτιμες γνώσεις που μας προσέφεραν όλα αυτά τα χρόνια.

Τέλος, αισθανόμαστε την ανάγκη να επισημάνουμε την απέραντη ευγνωμοσύνη στην οικογένεια μας, για την εμπιστοσύνη που μας έδειξαν όλα αυτά τα χρόνια των σπουδών μας. Πέραν ,όμως, από την πολύτιμη αυτή στήριξη, μας έδωσαν όλα τα εφόδια ώστε να γίνουμε σωστοί άνθρωποι, και αυτό είναι κάτι που δεν μαθαίνεται αλλά διδάσκεται.

Περίληψη

Τα διαδεδομένα πρότυπα κωδικοποίησης ηχητικής πληροφορίας (π.χ. MPEG-1, MPEG-2, MPEG-4 κλπ) εξασφαλίζουν ικανοποιητική ηχητική ποιότητα σε σχετικά χαμηλούς ρυθμούς παροχής δεδομένων. Ο τρόπος λειτουργίας τους καθώς και ο μεγάλος βαθμός συμπίεσης, εξασφαλίζεται από την ψηφιακή επεξεργασία του σήματος σε προκαθορισμένα στάδια, την ενσωμάτωση βασικών ψυχοακουστικών τεχνικών καθώς και μεθόδων μη απωλεστικής κωδικοποίησης (π.χ. Huffman coding). Αντικείμενο της εργασίας είναι η υλοποίηση των παραπάνω σταδίων καθώς και η απεικόνιση των εξόδων από τα επιμέρους στάδια, σε κατάλληλη γραφική διεπαφή στο προγραμματιστικό περιβάλλον Matlab.

Abstract

The widespread sound information coding standards (MPEG-1, MPEG-2, MPEG-4 etc.) provide good sound quality at relatively low data delivery rates. The mode of operation and the high degree of compression is provided by the digital signal processing at predetermined stages, integrating key psychoacoustic techniques and methods of lossless coding (for example Huffman coding). Scope of this project is to implement the above steps and to display the outputs from the individual stages, at a suitable graphical interface to the programming environment Matlab.

1 <<Ήχος>>

1.1 Ορισμός και θεωρία του ήχου

Τι είναι ο ήχος;

Μια απλή και εύκολη απάντηση στο ερώτημα μας είναι η εξής: ήχος είναι κάθε τι που ακούμε. Ο ήχος, όπως τουλάχιστον τον αντιλαμβανόμαστε καθημερινά, παράγεται από μια πηγή και συλλαμβάνεται από το αυτί μας. Όμως για να φτάσει ο ήχος από την πηγή στο δέκτη, πρέπει να μεσολαβήσει ένα φέρον μέσο. Αυτό συνήθως είναι ο αέρας. Βέβαια μέσο διάδοσης του ήχου μπορεί είναι και το νερό ή κάποιο στερεό σώμα. Χωρίς μέσο διάδοσης δεν είναι δυνατή η μεταφορά του ήχου. Γι' αυτόν τον λόγο, για παράδειγμα, δεν είναι δυνατή η συνομιλία στο διάστημα. Επίσης, το μέσο διάδοσης καθορίζει και τις ιδιότητες του ήχου. Αλλιώς π.χ. ακούγονται οι ήχοι στο νερό, αλλιώς φιλτράρονται και με άλλες ταχύτητες μεταδίδονται.

Εμάς θα μας απασχολήσουν οι ήχοι που συναντάμε στην καθημερινή μας ζωή, αυτοί που διαδίδονται στον αέρα. Όταν ο αέρας πάλλεται, δημιουργούνται κύματα. Αυτές οι κυμάνσεις συλλαμβάνονται ως ήχος. Το πλάτος και η συχνότητα τους διαμορφώνουν το ποιόν του ήχου που ακούγεται.

Άρα ο ήχος είναι ενέργεια.

Αυτό που αντιλαμβανόμαστε ως «ήχο», δεν είναι παρά μικρές ή μεγάλες μεταβολές της πίεσης του αέρα, ικανότητα την οποία απέκτησε ο άνθρωπος ώστε να μπορεί να εντάσσεται καλύτερα στο περιβάλλον του. Ο ήχος, που παράγεται από μια πηγή και συλλαμβάνεται από το αυτί μας, όσο συνηθισμένο φαινόμενο κι αν είναι, αποτελεί ένα από τα πιο εντυπωσιακά στοιχεία των πολυμεσικών εφαρμογών, καθώς μπορεί να προσφέρει ακουστική απόλαυση, να εντυπωσιάσει με διαφορετικά ηχητικά εφέ και να ξεκουράζει σαν ηχητική υπόκρουση.

Από φυσική λοιπόν άποψη, ένας ήχος παράγεται από μεταβολές της πίεσης που μεταδίδονται σε ένα μέσο που μπορεί να συμπιεστεί, ενώ παρόλο που ο μηχανισμός διάδοσης του ήχου διαφέρει από περίπτωση σε περίπτωση, οι βασικές αρχές μετάδοσης και παραγωγής του είναι οι ίδιες. Πιο συγκεκριμένα, τα μόρια όλων των φυσικών σωμάτων οφείλουν να κρατούν σταθερές αποστάσεις από τα γειτονικά τους μόρια. Κατά συνέπεια, αν για κάποιο λόγο τα μόρια σε μια περιοχή συμπιεστούν θα επιδιώξουν να επανέλθουν στη αρχική τους μορφή συμπιέζοντας με την σειρά τους τα μόρια των γειτονικών τους περιοχών. Με αυτό τον τρόπο, δημιουργούνται μεταβολές πίεσης που μεταδίδονται με μια ορισμένη ταχύτητα σε ένα μέσο, ούτως ώστε αυτές οι μεταβολές να αποτελούν ένα ηχητικό κύμα.

Παράλληλα, προκειμένου να διαδοθεί ένα ηχητικό κύμα σε ένα μέσο, δεν πραγματοποιείται μεταφορά των μορίων του μέσου αλλά μεταφορά της διαταραχής από το ένα μόριο στο διπλανό του, με αποτέλεσμα ο ήχος να χαρακτηρίζεται ως κύμα με το οποίο μεταφέρεται ενέργεια και όχι ύλη. Για παράδειγμα, με το πέρασμα ενός αεροπλάνου μπορεί να σπάσουν τα τζάμια κτηρίου, τα οποία βρίσκονται σε κοντινή απόσταση λόγω του ισχυρού ηχητικού κύματος που διαμορφώνεται.

Όπως διατυπώθηκε παραπάνω, ο ήχος μεταδίδεται εξαιτίας των μεταβολών της πίεσης του αέρα ενώ οι μεταβολές αυτές, δηλαδή τα πυκνώματα και τα αραιώματα του αέρα διαδίδονται σε όλες τις κατευθύνσεις στο γύρο χώρο. Για παράδειγμα, αν κτυπήσουμε την χορδή μιας κιθάρας, τότε η χορδή θα αρχίσει να ταλαντώνεται συμπιέζοντας τα μόρια του αέρα που βρίσκονται πάνω της, μέχρι που να συναντήσουν το τύμπανο του αυτιού μας διεγείροντας ένα ολόκληρο μηχανισμό με αποτέλεσμα την αίσθηση της ακοής. (Χρυσάνθου, 2006) Η ικανότητα του ανθρώπου να αντιλαμβάνεται τις μεταβολές αυτής της πίεσης και να αποκωδικοποιεί το περιεχόμενο τους, στηρίζεται στα τρία πιο κάτω στάδια: (α) η κατεύθυνση των κυμάτων του ήχου στο ακουστικό μέρος του αυτιού, (β) ο εντοπισμός της διακύμανσης του

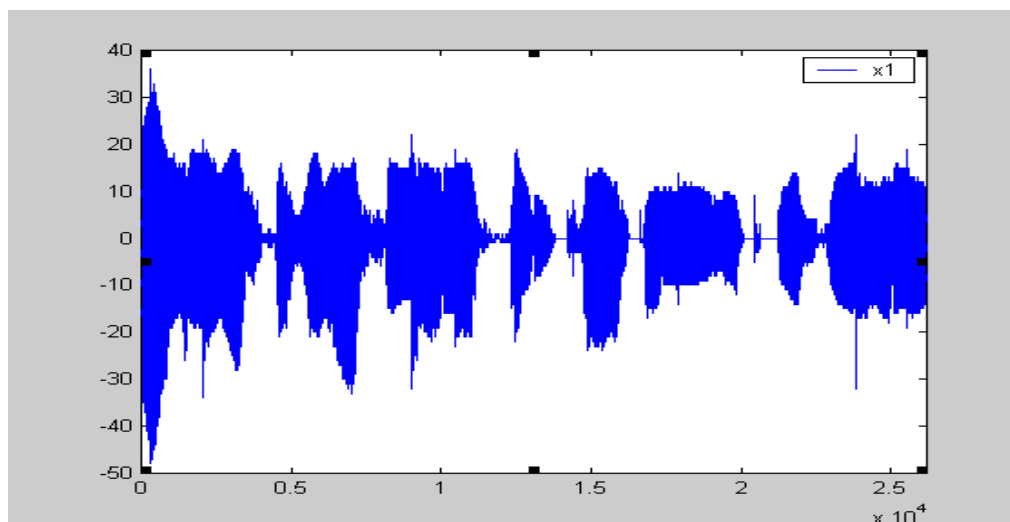
και (γ) η μετάφραση της διακύμανσης σε ηλεκτρικά σήματα όπου και γίνονται κατανοητά από τον εγκέφαλο. Παράλληλα, ο ανθρώπινος εγκέφαλος έχει την ικανότητα να συγκρίνει την χρονική διαφορά με την οποία ένας ήχος φτάνει στο αυτί, οπότε και συμπεράνει την απόσταση από την οποία προέρχεται.

1.2 Βασικές Αρχές Θεωρίας Ήχου

Αναμφισβήτητα ζούμε σε ένα "ωκεανό ήχων" που παρουσιάζουν μεγάλες διαφορές μεταξύ τους. Ήχοι δυνατοί, ήχοι ανεπαίσθητοι, ήχοι βαρείς, ήχοι λεπτοί, θόρυβοι, κρότοι. αρμονίες, ήχοι για όλες τις προτιμήσεις. (ΠΑΠΑΔΑΚΗΣ)

Τι είναι όμως εκείνο που διαφοροποιεί τους ήχους μεταξύ τους; Κάθε ήχος έχει μια ιδιαιτερότητα, η οποία οφείλεται σε ένα σύνολο από ηχητικά μεγέθη. Άλλα από τα μεγέθη αυτά είναι ανεξάρτητα από την προσωπική αντίληψη του ακροατή, λέγονται αντικειμενικά, ενώ άλλα που εξαρτώνται και από τον ακροατή, λέγονται υποκειμενικά γνωρίσματα ή χαρακτηριστικά του ήχου.

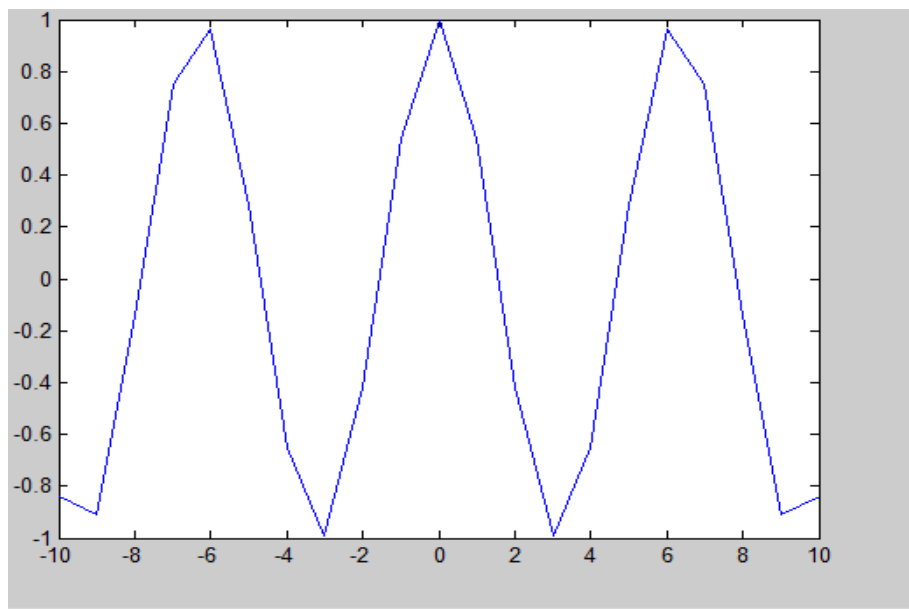
Όπως είδη έχει αναφερθεί, ο ήχος από την φύση του είναι μια



Εικόνα 1-1 Κυματομορφή σήματος με μεταβαλλόμενη ισχύ

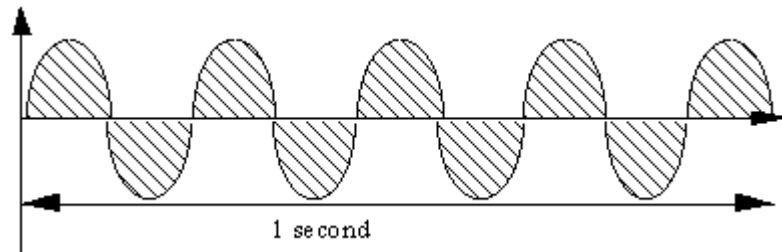
αναλογική ποσότητα και για να γίνει αντικείμενο επεξεργασίας θα πρέπει να ψηφιοποιηθεί ενώ για την κατανόηση της μεθόδου εισαγωγής, οφείλουμε να θέσουμε τις βασικές αρχές θεωρίας του. Ο ήχος, είναι η αίσθηση που αντιλαμβανόμαστε όταν τα ακουστικά νεύρα ενεργοποιούνται από μόρια που δονούνται στον αέρα, π.χ. όταν κτυπάει ένα κουδούνι αυτό δονείται. Η κίνηση αυτή διοχετεύεται στα περιβάλλοντα μόρια του αέρα, όπου και δονούν τα γειτονικά τους μόρια προκαλώντας με αυτή την κίνηση δόνηση στο τύμπανο. Εντούτοις, ο ήχος παρουσιάζεται και ως **κυματομορφή**, το μέγεθος της οποίας αντιπροσωπεύει την ευρύτητα του ήχου όπως φαίνεται στο παρακάτω σχήμα.

Επίσης, το **ύψος** αναδεικνύει την σχετική ισχύ ή δύναμη του ήχου. Στο προηγούμενο σχήμα, ο ήχος στα αριστερά είναι πιο δυνατός σε σχέση με αυτόν στα δεξιά. Εδώ παρατηρούμε πως οι δύο κυματομορφές έχουν διαφορετικό ύψος, άρα αναπαριστούν δύο ηχητικά σήματα με διαφορετική ένταση. Η απόσταση μεταξύ δύο συνεχόμενων κορυφών ονομάζεται **περίοδος** (period) και φαίνεται στο παρακάτω σχήμα.



Εικόνα 1-2 Κυματομορφή περιοδικού σήματος

Ο αριθμός των κορυφών που διαμορφώνονται σε χρονικό διάστημα ενός δευτερολέπτου λέγεται **συχνότητα**, η οποία μετράται σε Hertz (Hz) ή kilohertz (Khz) και έχει τιμή αντίστροφη της περιόδου.



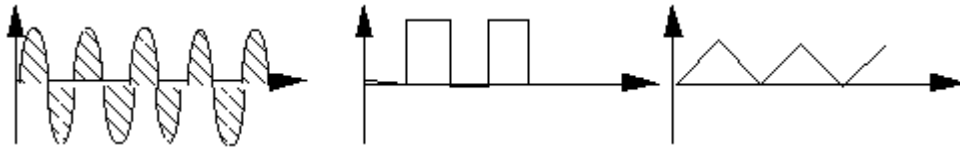
Εικόνα 1-3 Κυματομορφή περιοδικού σήματος

Ο **τόνος** (pitch) ενός ήχου χαρακτηρίζεται ως η ψυχολογική εντύπωση της υψηλότητας ή της χαμηλότητας του, με επακόλουθο όσο πιο υψηλός είναι ο τόπος τόσο πιο υψηλή είναι η συχνότητα του. Στο παρακάτω σχήμα αναδύεται η σχέση μεταξύ του τόνου και της συχνότητας.



Εικόνα 1-4 Σχέση τόνου και συχνότητας

Στο παρακάτω σχήμα τίθενται τρία βασικά χαρακτηριστικά δείγματα κυματομορφών ήχου, ημιτονοειδή, τετραγωνική και πριονωτή. Η ημιτονοειδής κυματομορφή έχει την ικανότητα να ακούγεται ευχάριστα, η τετραγωνική πιο σκληρά ενώ οι περίπλοκοι κυματισμοί ακούγονται φυσικοί. Συμπερασματικά, όσο πιο σταθερός και ομαλός είναι ο κυματισμός τόσο πιο σαφής και καθαρός είναι ο ήχος. Όσο πιο πολλές γωνίες διαμορφώνει ο κυματισμός τόσο πιο σκληρά ακούγεται ο ήχος.



Εικόνα 1-5 Διάφοροι κυματισμοί

1.3 Ενότητα Χαρακτηριστικά Ψηφιακού Ήχου

Η ιδιαιτερότητα κάθε ήχου οφείλεται σε ένα σύνολο από ηχητικά μεγέθη. Όταν τα μεγέθη αυτά είναι ανεξάρτητα από την προσωπική αντίληψη του ακροατή και αποκαλούνται αντικείμενα, διαφορετικά αν είναι άμεσα συνδεδεμένα με τον ακροατή και λέγονται υποκειμενικά χαρακτηριστικά του ήχου. Αντικειμενικά χαρακτηριστικά, είναι η συχνότητα και η ένταση και υποκειμενικά η ακουσιτικότητα, το ύψος και η χροιά. (Αμβράζης, 2009)

Συχνότητα: Αν θέλουμε να περιγράψουμε τον ήχο που παράγουν δύο πνευστά μουσικά όργανα όπως η τρομπέτα και η τούμπα, θα παρατηρήσουμε ότι παρόλο που και τα δύο είναι παρόμοια όργανα, η τρομπέτα παράγει πιο υψηλό ήχο από την τούμπα. (ΜΑΥΡΟΓΙΑΝΝΗΣ, 2008) Το ύψος του ήχου είναι ένα υποκειμενικό γνώρισμα που σχετίζεται αμοιβαία με ένα αντικειμενικό χαρακτηριστικό, τη συχνότητα. Η συχνότητα έχει να κάνει με το πόσο γρήγορα

όργανο	Συχνοτικό εύρος (σε Hz)
Ανθρώπινη φωνή	70-2000
Πιάνο	30-3500
Βιολί	200-3000
Φλάουτο	260-3000
Εκκλησιαστικό όργανο	16-4000

Εικόνα 1-6 Παραδείγματα συχνοτικού εύρους

ή αργά πάλλεται το σώμα που δημιουργεί τον ήχο, για παράδειγμα, όταν η χορδή μιας κιθάρας πάλλεται 100 φορές το δευτερόλεπτο, τότε προκαλείται ο αντίστοιχος αριθμός πυκνωμάτων ούτως ώστε, να γίνεται αντιληπτός ο συγκεκριμένος ήχος. Συμπερασματικά, η συχνότητα ορίζει τον αριθμό των κύκλων που εκτελεί η συνάρτηση ανά δευτερόλεπτο και μετριέται σε κύκλους ανά δευτερόλεπτο ή Hertz (Hz).

Κάθε ήχος για να γίνει αντιληπτός από τι ανθρώπινο αυτί, θα πρέπει να έχει συχνότητα μεταξύ 20 και 20.000Hz. Οι πιο χρήσιμες συχνότητες βρίσκονται κάτω από 10 kHz, (π.χ. ομιλία, μουσική, διάφοροι θόρυβοι). Εντούτοις, υπάρχουν ήχοι με συχνότητα μεγαλύτερη των 20 kHz οι οποίοι ονομάζονται και υπέρηχοι, ενώ αυτοί με μικρότερη συχνότητα των 20 Hz λέγονται υπόηχοι. Αν και οι δύο κατηγορίες δεν έχουν επίδραση από το ανθρώπινο αυτί, μπορεί να έχουν επίδραση στη υγεία του.

Ένταση: Το δεύτερο βασικό γνώρισμα του ήχου είναι η ένταση, στενά συνδεδεμένη με την ισχύ του ηχητικού σήματος που διεγείρει το αυτί μας. Η κύρια αιτία που οι ήχοι έχουν διαφορετικές εντάσεις είναι ότι πιέζουν με διαφορετική δύναμη το τύμπανο του αυτιού μας, δηλαδή το πόσο έντονες είναι οι αναταράξεις που προκαλεί το σώμα που παράγει τον ήχο. (Δρακόπουλος, 2008) Όσο πιο μεγάλη ισχύ διαμορφώνουν τα ηχητικά κύματα ολοένα και περισσότερη δύναμη εξασκούν στο μηχανισμό του αυτιού μας. Για παράδειγμα, κτυπώντας απλά την χορδή μιας κιθάρας τότε αυτή πάλλεται με μια συχνότητα, δημιουργώντας αναταράξεις που όμως δεν είναι έντονες. Αντίθετα αν κτυπήσουμε την ίδια χορδή με δύναμη παρατηρείται πως πάλλεται με την ίδια συχνότητα μεν, αλλά με πολύ εντονότερες διαταραχές. Άρα, η ένταση ορίζεται από το πλάτος της δόνησης με αποτέλεσμα όσο μεγαλύτερο είναι το πλάτος τόσο ισχυρότερος να είναι ο ήχος.

Η σχετική τιμή μεταξύ δύο ηχητικών κυμάτων μετριέται σε decibels (dB=μονάδα μέτρηση της έντασης ενός ήχου και 1dB αντιστοιχεί σε 0,1 bel). Για να συγκρίνουμε την ισχύ δύο ηχητικών κυμάτων υπολογίζουμε

τον λόγο της ισχύος τους. Η λογαριθμική κλίμακα dB παρουσιάζει τη σχέση της ηχητικής ισχύος σε μια συγκεκριμένη πηγή, σε σχέση με την χαμηλότερη ηχητική ισχύ που μπορεί να γίνει αντιληπτή από το ανθρώπινο αυτί.

$$\text{Ένταση ήχου} = 10 \log(P/P_0) \text{ dB}$$

Η τιμή P = τιμή μέτρησης

Η τιμή P_0 = τιμή αναφοράς σε μονάδες Watt/m^2

Παράδειγμα: Για $P = 10^{-12} \text{Watt/m}^2$ έχουμε

$$\text{Ένταση} = 10 \log(10^{-12}/10^{-12}) = 10 \log 1 = 0 \text{ dB}$$

Δηλαδή, ο ήχος έντασης 0 dB αντιστοιχεί στο κατώφλι ακουστικότητας ενώ ο ήχος έντασης 140dB αντιστοιχεί στο όριο πόνου. Συμπερασματικά, μικρότερη ένταση σε dB αντιπροσωπεύει μεγάλη αύξηση ηχητικής ισχύος ενώ θεωρητικά, η αύξηση κατά 3 dB διαμορφώνει διπλασιασμός της ηχητικής ισχύος. Έτσι, όταν ένας ήχος είναι 10 φορές πιο ισχυρός από την σχεδόν απόλυτη ησυχία έχει ένταση 10dB, 100 φορές πιο ισχυρός έχει ένταση 20dB και 1000 φορές έχει ένταση 30dB. (ΧΡΙΣΤΟΔΟΥΛΟΣ, 2011)

2 «Διαδικασία Ψηφιοποίησης Ήχου»

2.1 Στάδια Ψηφιοποίησης Ηχητικών Αρχείων

Η ραγδαία ανάπτυξη της επιστήμης των υπολογιστών και της ψηφιακής τεχνολογίας τα προηγούμενα χρόνια, είχε σαν αποτέλεσμα να επεκταθούν στην κυριολεξία όλοι οι φραγμοί που επί δεκαετίες υπήρχαν στο χώρο της μουσικής απόλαυσης. Μπήκε πια ένα τέρμα στους διαχωρισμούς μεταξύ επαγγελματικών και ερασιτεχνικών στούντιο και αυτό πρόκειται ούτε λίγο ούτε πολύ για ένα θαύμα, είτε μας αρέσει είτε όχι η ψηφιοποίηση του κόσμου αποτελεί κατόρθωμα του πνεύματος.

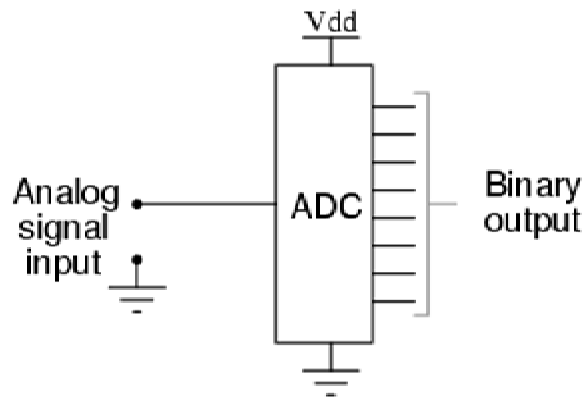
Ο ήχος είναι μια διακύμανση πιέσεων και το μικρόφωνο αποτελεί το όργανο εκείνο που μετατρέπει ένα ηχητικό σήμα σε διακύμανση ηλεκτρονικής τάσης, αυτή η ηλεκτρονική τάση είναι το αναλογικό

ηλεκτρονικό σήμα. Για να γίνει δυνατή η επεξεργασία του από τον Η/Υ απαραίτητη προϋπόθεση είναι η ψηφιοποίηση του, μετατρέποντας το αναλογικό (analogy) σε ψηφιακό (digital) με την χρήση των μετατροπέων, ADC (Analog-to-Digital-Converters) και αντίστροφα DAC (Digital-to-Analog-Converters/DACs). Έτσι, η διαδικασία αυτή συμπεριλαμβάνει τρεις βασικές λειτουργίες: την «δειγματοληψία» του αρχικού σήματος, τον «κβαντισμό» των τιμών του σήματος διακριτού χρόνου που προκύπτει από την δειγματοληψία και τέλος η «κωδικοποίηση».

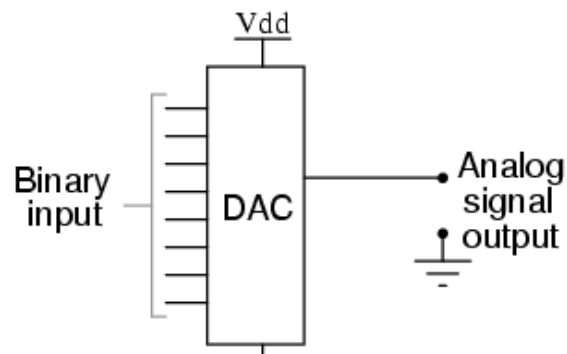
Πιο συγκεκριμένα, κατά την **δειγματοληψία** παίρνονται δείγματα του πλάτους του αναλογικού σήματος σε τακτά χρονικά διαστήματα με συνεπακόλουθο να δημιουργείται ένα στιγμιότυπο του σήματος της συγκεκριμένης τιμής, σύνολο το οποίου τα δείγματα διαμορφώνουν το ψηφιοποιημένο σήμα. Όσο περισσότερα είναι τα δείγματα, τόσο καλύτερη είναι η ποιότητα του σήματος που προκύπτει. Αντίθετα, στον **κβαντισμό** οι διαδοχικές τιμές της στάθμης (πλάτους) του σήματος διακριτού χρόνου $s_d(nT_s)$ μετατρέπονται σε διακριτές (ψηφιακές) τιμές, πραγματοποιείται δηλαδή μια απεικόνιση της μορφής $s(nT_s) = Q\{s_d(nT_s)\}$, όπου $Q\{s_d(nT_s)\}$ είναι η κβαντισμένη τιμή. Κατά την παραπάνω μετατροπή το τελικό σήμα, που είναι διακριτό τόσο ως προς το χρόνο όσο και ως προς το πλάτος, καλείται ψηφιακό. (Φλώρος, 2013)

Όπως έχει ήδη διατυπωθεί, το αναλογικό σήμα ακουστικής πίεσης μετατρέπεται σε ψηφιακό με την χρήση υποσυστημάτων εφαρμογής ADC (Analog-to-Digital-Converters) και το αντίστροφο, υλοποιείται με την χρήση DAC μετατροπών (Digital-to-Analog- Converters/DACs) όπου οι κβαντισμένες τιμές μετατρέπονται σε αναλογικές. Σε τέτοια υποσυστήματα οι διαδικασίες κβαντισμού και δειγματοληψίας πραγματοποιούνται ταυτόχρονα, ενώ η υλοποίηση των μετατροπέων και στις δύο φάσεις, παρουσιάζουν τόσο θεωρητικά όσο και πρακτικά προβλήματα, ενώ για την σταθερή μετατροπή

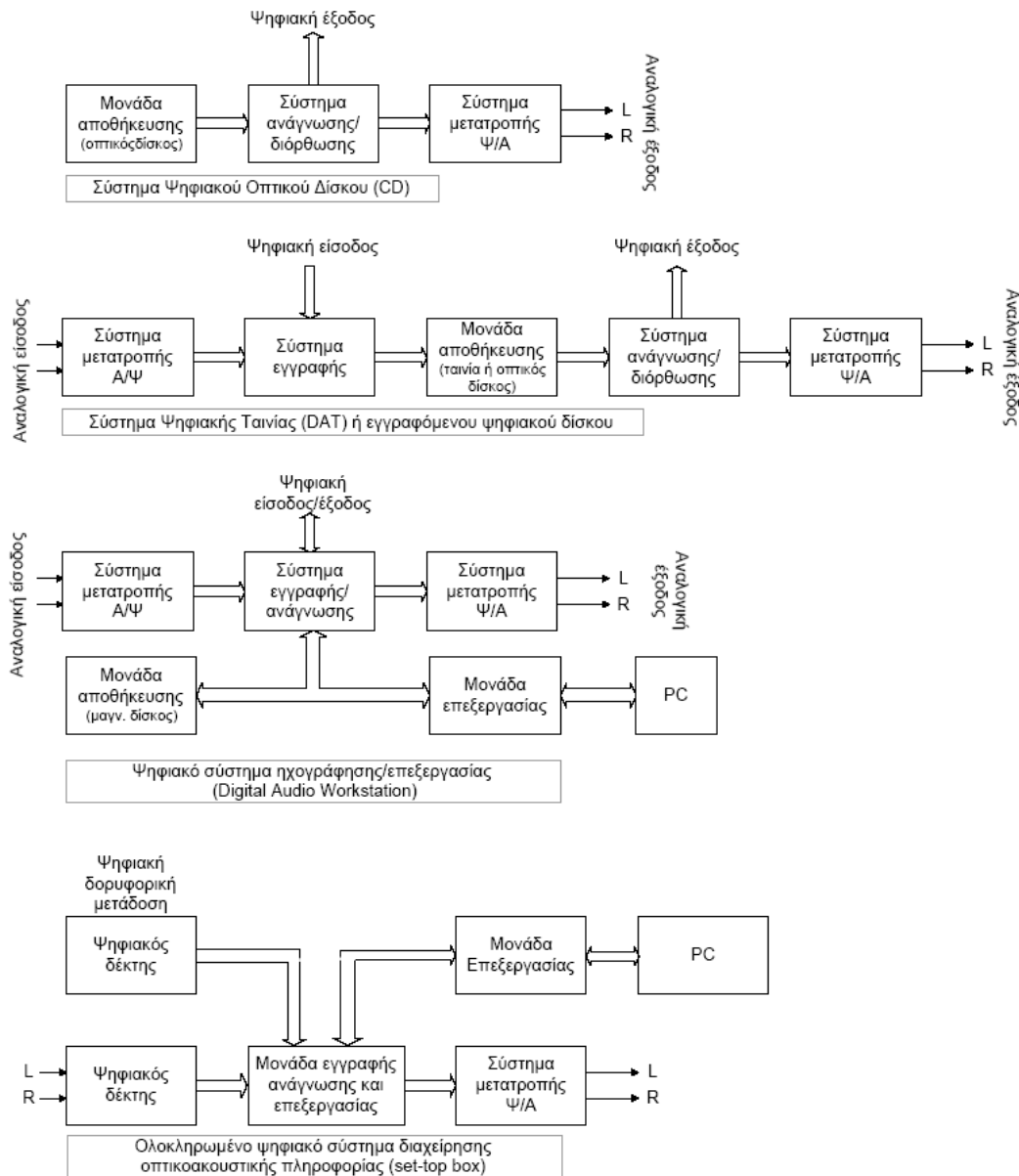
καλό θα είναι να ικανοποιούνται συγκεκριμένες συνθήκες δειγματοληψίας και κβαντισμού στις οποίες αναφορά γίνεται πιο κάτω.



Εικόνα2-1ADC (Analog to digital converter)



Εικόνα2-2 DAC (Digital to analog converter)



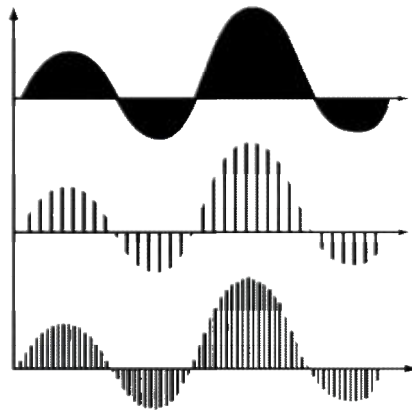
Εικόνα 2-3 Διαδικασία ψηφιοποίησης ηχητικού σήματος

2.2 Δειγματοληψία

Όπως προαναφέρθηκε, η δειγματοληψία είναι η διαδικασία κατά την οποία παράγεται η σειρά των δειγμάτων, από τα οποία τελικά δημιουργείται η ψηφιακή πληροφορία. Πρόκειται δηλαδή η διαδικασία κατά την οποία ένα αναλογικό σήμα $sc(t)$ (σήμα του οποίου η ανεξάρτητη χρονική μεταβλητή t , παίρνει συνεχείς τιμές) μετατρέπεται σε σήμα διακριτού χρόνου (σήμα το

οποίο ορίζεται σε διακριτές χρονικές στιγμές), πραγματοποιείται δηλαδή μια απεικόνιση της μορφής $s_c(t) \rightarrow sd(nTs)$ όπου $n=1,2,\dots$ είναι ο αριθμός του δείγματος και $Ts(\text{sec})$ η περίοδος δειγματοληψίας.

Η επιλογή της περιόδου δειγματοληψίας, καθορίζει και το μέτρο της ποιότητας του ψηφιακού ήχου με επακόλουθο να αποτελεί καθοριστικό παράγοντα ορθής μετατροπής ενός σήματος συνεχούς χρόνου σε διακριτού χρόνου δείγματος. Η συχνότητα δειγματοληψίας (sampling frequency), δηλώνει τον αριθμό των δειγμάτων πλάτους του αναλογικού σήματος καθώς και το πόσες φορές το δευτερόλεπτο παίρνουμε δείγματα από το αναλογικό σήμα, το οποίο μετριέται σε χιλιάδες κύκλους ανά δευτερόλεπτο η kHz. Όσο μεγαλύτερη είναι η συχνότητα ψηφιοποίησης τόσο καλύτερη ποιότητα αναπαραγωγής έχουμε άλλα ταυτόχρονα, με ριζική αύξηση του χώρου αποθήκευσης.



Εικόνα 2-4 Α) Αναλογικό σήμα Β) Ψηφιακό με χαμηλή συχνότητα Γ) Ψηφιακό με υψηλή συχνότητα δειγματοληψίας

Απαραίτητη διαδικασία για την μετατροπή ενός αναλογικού ηχητικού σήματος σε ψηφιακό είναι η δειγματοληψία, όπως αναδύεται στο **θεώρημα του Nyquist** από το οποίο απορρέει ότι συχνότητα δειγματοληψίας θα πρέπει να είναι μεγαλύτερη ή ίση του διπλάσιου της μέγιστης συχνότητας του αρχικού αναλογικού σήματος, $f_s > 2f$ ή $T_s \leq T/2$. Για να ισχύει το παραπάνω πρέπει το αρχικό σήμα να αναζωπυρωθεί στο μισό της συχνότητας δειγματοληψίας περνώντας από ένα ιδανικό βαθυπερατού φίλτρου. (Tsapatsoulis, 2006)

Οι συχνότητες δειγματοληψίας που χρησιμοποιούνται σήμερα για την ψηφιακή εγγραφή μουσικής, κυμαίνονται στα 22 με 44kHz επιτρέποντας να περάσει ένα φάσμα ακουστικών συχνοτήτων από 11 έως 22kHz. Κατά συνέπεια, ο ήχος που γίνεται αντιληπτός από το ανθρώπινο αυτί δεν ξεπερνά τα 18 με 22kHz και γι' αυτό θα πρέπει να δειγματολογούμε με συχνότητα περίπου 44kHz(CD) με επακόλουθο ένα πιστό ψηφιακό αντίγραφο του αναλογικού σήματος. Ο A/D παρέχει σε κάθε δείγμα μια τιμή ακέραιου αριθμού που ισοδυναμεί με το μέγεθος του δείγματος . Η ανάλυση τιμών εξαρτάται από την ανάλυση των δειγμάτων (sampling resolution), όπου εάν ο A/D είναι 8 bits, οι τιμές μπορεί να κυμαίνονται από 0 έως 255. Συνήθως κατά την διαδικασία της ηχογράφηση επιλέγουμε την μέγιστη ανάλυση δειγμάτων που υποστηρίζεται από το hardware. Ενώ μια καλή επιλογή είναι τα 8 bits και τα οποία χρησιμοποιούνται ευρέως . Αντίθετα, για ποιότητα CD ο ψηφιακός ήχος κωδικοποιείται ως 44,1kHz στηριζόμενος στο 16bit σύστημα δηλ. το αρχικό κύμα τεμαχίζεται 44.100 φορές το δευτερόλεπτο και ένα μέσο εύρος επίπεδο εφαρμόζεται σε κάθε δείγμα (16 bit σημαίνει ότι 65.536 διαφορετικές τιμές μπορούν να οριστούν ή να κβαντοποιηθούν σε κάθε δείγμα).

Πίνακας 1 Ρυθμός δειγματοληψίας και δείγματα ανά sec

Κανάλια	Δείγμα	Δειγματοληψία	Χρόνος Δείγματος	Μέγεθος Αρχείου
Mono	8bits	11kHz	10 sec	110000 bytes
Stereo	8bits	11kHz	10 sec	220000 bytes
Mono	8bits	22kHz	10 sec	220000 bytes
Stereo	8bits	22kHz	10 sec	440000 bytes
Mono	16bits	44,1kHz	10 sec	882000 bytes
Stereo	16bits	44,1kHz	10 sec	1764000 bytes

Στον παραπάνω πίνακα παρατηρείται ο ρυθμός δειγματοληψίας (samplingrate) και ο αριθμός των δειγμάτων που συλλέγονται ανά sec. Όπως είναι φυσικό, το μέγεθος του αρχείου στο οποίο αποθηκεύεται το ψηφιοποιημένο σήμα ολοένα και αυξάνεται με το μέγεθος του δείγματος

(*sampling size*) και το ρυθμό δειγματοληψίας (*sampling rate*) , με αποτέλεσμα η καλύτερη απόδοση να επιταχύνεται με δειγματοληψία 44,1 kHz με μέγεθος δείγματος 16bits σε 2 κανάλια ήχου (ποιότητα μουσικών CD).

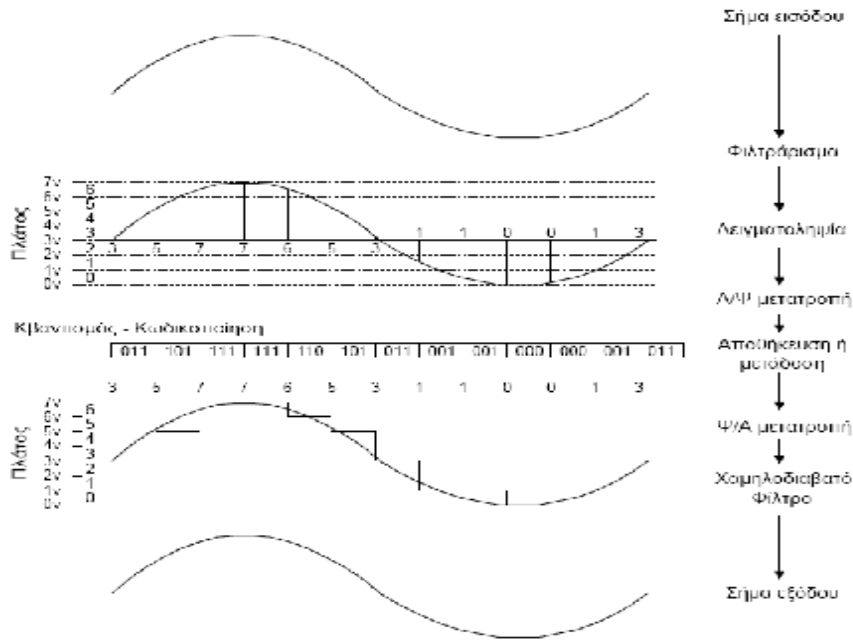
Συμπερασματικά, για να υπολογίσουμε το μέγεθος ενός ψηφιακού αρχείου το οποίο εξαρτάται από την χρονική διάρκεια και την ποιότητα του ήχου, πολλαπλασιάζουμε την συχνότητα δειγματοληψίας σε Hz με το μέγεθος του δείγματος σε bit και με την διάρκεια του ήχου σε δευτερόλεπτα :

Χωρητικότητα (bits) = Συχνότητα (Hz) . Μέγεθος δείγματος (bits) . Διάρκεια (sec)

2.3 Προβλήματα Δειγματοληψίας

Ο H.Nyquist, πρώτος ανακάλυψε ότι η μέγιστη συχνότητα αναλογικού σήματος που μπορεί να αποδοθεί χωρίς αλλοίωση είναι το μισό της συχνότητας δειγματοληψίας. Συχνότητες μεγαλύτερες από τα μισά της συχνότητας ψηφιοποίησης εμφανίζονται λανθασμένες σαν χαμηλές συχνότητες, με επακόλουθο το φαινόμενο αυτού του ελλιπούς ρυθμού δειγματοληψίας να ονομάζεται αναδίπλωση (*aliasing*) .Έτσι για να μην εμφανιστούν συχνοτικά ειδώλια που θα παραμορφώσουν το σήμα μετά την δειγματοληψία οι ψηφιοποιητές φιλτράρουν όλες τις συχνότητες που είναι μεγαλύτερες από το μισό της συχνότητας δειγματοληψίας.(Βελώνη)

Κατά συνέπεια, το ηλεκτρονικό σήμα μετά το στάδιο της προενίσχυσης τροφοδοτείται σε ένα ιδανικό βαθυπερατό φίλτρο (*low-pass*) , με στόχο την αποκοπή των συχνοτήτων μεγαλύτερων από την μέγιστη συχνότητα δειγματοληψίας για αποφυγή του φαινομένου της αναδίπλωσης. Στη συνέχεια, κατά το στάδιο της δειγματοληψίας τροποποιείται το αναλογικό ηλεκτρικό σήμα της εισόδου σε δυαδικούς αριθμούς και αποθηκεύεται στη μνήμη του Η/Υ, με αποτέλεσμα να ολοκληρώνεται ο μετασχηματισμός από το αναλογικό (*analog*) στο ψηφιακό (*digital*).



Εικόνα 2-5 Διαδικασία μετατροπής αναλογικού σε ψηφιακό και αντίστροφα

Η απόδοση ενός ψηφιακού ηχητικού σήματος ακολουθεί μια αντίστροφη διαδρομή βημάτων από αυτή της καταγραφής, δηλαδή η τροφοδότηση του σήματος υλοποιείται από τη μνήμη του Η/Υ σε ένα μετατροπέα ψηφιακού σε αναλογικού σήματος. Με βάση τον τρόπο λειτουργίας και την συχνότητα δειγματοληψίας, ο μετατροπέας παράγει ένα ηλεκτρικό σήμα που αποτελεί μια προσεγγιστική μορφή του ηλεκτρικού σήματος το οποίο ενισχύεται και στη συνέχεια τροφοδοτείται στη είσοδο του ηχείου, αποδίδοντας το σήμα.

Στο σημείο αυτό, ανακύπτει το ερώτημα αν η εισαγωγή τέτοιων φίλτρων (filter), διαμορφώνουν νέα προβλήματα κατά την ψηφιοποίηση του ήχου. Απάντηση είναι, πώς δεν υπάρχει ένα τέλειο βαθυπερατό φίλτρο που να μπορεί να αποκόψει τέλεια συχνότητες πάνω από ένα επιθυμητό όριο, ενώ η χρήση τέτοιων φίλτρων διαμορφώνει μικρές παραμορφώσεις στο σήμα στο οποίο και εκφράζεται. Ομοίως, όπως έχει ήδη τυπωθεί κατά την διαδικασία της Ψ/Α μετατροπής εξαιτίας των κβαντισμένων τιμών πλάτος του αναπαραγόμενου σήματος, διαμορφώνονται συχνότητες υψηλότερες της

μέγιστης συχνότητας όπου και φιλτράρονται με ένα παρόμοιο φίλτρο εξομάλυνσης (smoothingfilter).

2.4 Κβαντισμός

Το επόμενο πολύ σημαντικό βήμα είναι αυτό του **κβαντισμού**, προκειμένου να αναπαρασταθεί κάθε δείγμα με την μορφή μιας δυαδικής σειράς από bits, καθώς πρέπει στο συνεχώς μεταβαλλόμενο πλάτος της τάσης του αναλογικού σήματος να τεθεί μια διακριτή τιμή. Σημαντικό είναι να τονίσουμε ότι η δειγματοληψία και η κβαντοποίηση είναι συμπληρωματικές διαδικασίες και ο συνδυασμός αυτών των δύο διαδικασιών, καλείται ψηφιοποίηση. Ο κβαντισμός παίζει καθοριστικό ρόλο στο σχεδιασμό του συστήματος του ψηφιακού ήχου, ενώ ανάλογα με τις στάθμες κβάντισης διαμορφώνεται και το format του ψηφιακού ήχου.

Βασική παράμετρος της φάσης αυτής είναι το μέγεθος του δείγματος (samplingsize), με επακόλουθο όσα περισσότερα bits χρησιμοποιούνται με μεγαλύτερη ακρίβεια περιγραφής. Για παράδειγμα, αν έχουμε 8 bit, τότε μπορούν χρησιμοποιηθούν $2^8=256$ διαφορετικές τιμές, ενώ για 16bit μπορούν να χρησιμοποιηθούν $2^{16}=65.536$ διαφορετικές τιμές, άρα μεγαλύτερη πιστότητα και ακρίβεια.

Κατά την πεπερασμένη διακριτικότητα της ψηφιακής αναπαράστασης, εισάγεται ο θόρυβος ένα σημαντικό πρόβλημα τόσο κατά την ηχογράφηση όσο και κατά την αναπαραγωγή, γνωστός και ως **θόρυβος κβαντισμού**. Το φαινόμενο αυτό προέρχεται από το γεγονός ότι τα δείγματα του αναλογικού σήματος που θα χρησιμοποιηθούν δεν είναι κατά ανάγκη ακέραιοι και συνεπώς η μετατροπή τους σε ψηφιακά δείγματα συνεπάγεται στην στρογγυλοποίηση του δείγματος. Πρέπει, όμως να τονισθεί ότι στη περίπτωση αυτή το ψηφιοποιημένο σήμα απαρτίζεται από το άθροισμα του

αναλογικού σήματος καθώς και από ένα σήμα θορύβου το οποίο και μετριέται σε αρνητικά decibel (db= μονάδα μέτρησης έντασης του ήχου) .Το σφάλμα αυτό εκφράζεται από το λόγο σήματος προς θόρυβο, (SignaltoNoiseRatio, SNR) $S/E \text{ (dB)}=6.02n+176$ όπου το n είναι ο αριθμός των bits της λέξης στο format του ψηφιακού ήχου. Όσο μεγαλύτερος είναι ο αριθμός των db τόσο μικρότερος είναι ο θόρυβος, ενώ τα περισσότερα συστήματα σήμερα κυμαίνονται στα 90dbSNR, όπου η παραμόρφωση μετριέται επί τις εκατό του αρχικού σήματος Η ευκρίνεια του κβαντισμού και του λόγου σήματος προς θόρυβο παρουσιάζεται στον παρακάτω πίνακα, όπου ενδεικτικές τιμές SNR αντιστοιχούν σε διαφορετικές τάξεις μετατροπής.

Πίνακας 2 - Ευκρίνεια Κβαντισμού και SNR

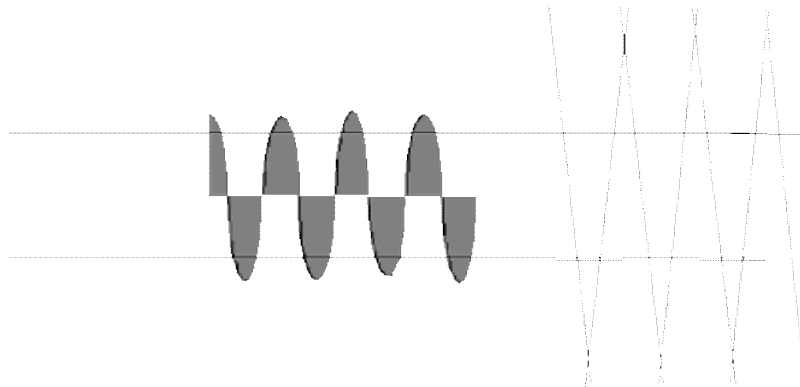
Τάξη N(bits)	SNR (db)
8	49.8
12	73.8
16	97.8
18	109.8

Συνοψίζοντας τα πιο πάνω, ανάγεται το συμπέρασμα ότι ο θόρυβος που παράγουν τα ψηφιακά ηχητικά συστήματα , εμφανίζεται κατά την διαδικασία της αναπαραγωγής του κβαντισμένου ηχητικού σήματος, έτσι ώστε να διαμορφώνεται ένας ανεπιθύμητος χαρακτήρας.

Στην περίπτωση των 16bit συστημάτων, επιβάλλεται η χρήση του dither κατά την Α/Ψ, μέθοδος κατά την οποία ενσωματώνεται μικρή ποσότητα αναλογικού θορύβου πριν από το στάδιο της δειγματοληψίας, εξασθενώντας πολλά από τα προβλήματα που συζητήθηκαν παραπάνω ούτως ώστε να εμφανίζεται ένα ικανοποιητικό αποτέλεσμα. Σε γενικές γραμμές, η προσθήκη του dither τροποποιεί την μορφή του σφάλματος και το καθιστά ανεξάρτητο από το σήμα εισόδου, αποκόπτει κάθε είδους αρμονικής παραμόρφωσης η οποία εμφανίζεται κατά την μετατροπή σήματος μικρού

πλάτους και τέλος βελτιώνει την διακριτή ικανότητα του κβαντιστή αυξάνοντας κατά υποκειμενικό τρόπο την δυναμική του περιοχή.

Ένας άλλος σημαντικός παράγοντας που επηρεάζει την ποιότητα του ήχου, είναι ο **ψαλιδισμός** (clipping) του σήματος, όπου κατά την εγγραφή του σήματος η μέγιστη στάθμη του σήματος εισόδου υπερβαίνει την μέγιστη στάθμη κβαντισμού διαμορφώνοντας κάποια διαστρέβλωση. Στο παρακάτω σχήμα 2.4, απεικονίζεται ένα ψαλιδισμένο σήμα με κοψίματα στην κορυφή και στη βάση με επακόλουθο ο ψαλιδισμός να μειώνεται αφού μειωθεί το μέγεθος του σήματος.



Εικόνα 2-6 Ψαλιδισμός

3 «Κωδικοποίηση Ψηφιακού Σήματος»

3.1 Μέθοδοι Κωδικοποίησης

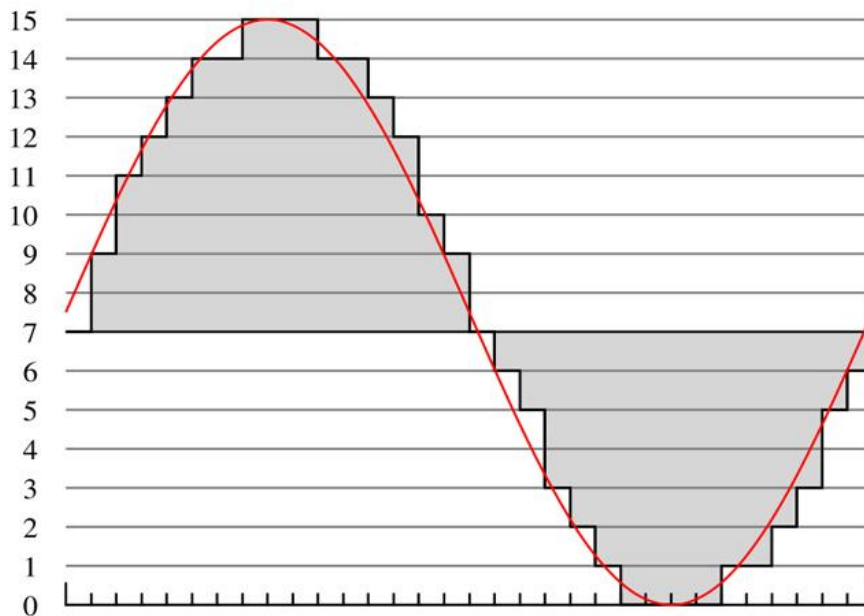
Ο συνδυασμός της δειγματοληψίας και της κβαντοποίησης ως συμπληρωματικές διαδικασίες αποκαλείται ψηφιοποίηση, ενώ για την εκμετάλλευση των πλεονεκτημάτων τους απαραίτητη προϋπόθεση η χρησιμοποίηση μιας διαδικασίας **κωδικοποίησης** (encoding process) για την μετατροπή του διακριτού συνόλου των τιμών των δειγμάτων σε μια πιο κατάλληλη μορφή. Κατά το στάδιο αυτό, η ένταση του κάθε δείγματος αντιστοιχεί σε ένα σταθερό δυαδικό αριθμό 0,1 και διατίθεται για την αποθήκευση κάθε δείγματος και καλείται εύρος δείγματος (τιμές που μπορεί να πάρει το κάθε δείγμα).

Για το ψηφιακό ήχο οι δυνατότητες ποικίλουν από 8bits ($2^8=256$ στάθμες) έως και 16bits ($2^{16}=65536$ στάθμες), με επακόλουθο η χρήση ενός δυαδικού συστήματος να αναδύει καλύτερα αποτελέσματα σε συνάρτηση με την επίδραση του θορύβου σε ένα μέσο μετάδοσης και αυτό γιατί ένα δυαδικό σύμβολο, αντέχει σε μια σχετική υψηλή στάθμη θορύβου καθώς εύκολα μπορεί να αναγεννηθεί.

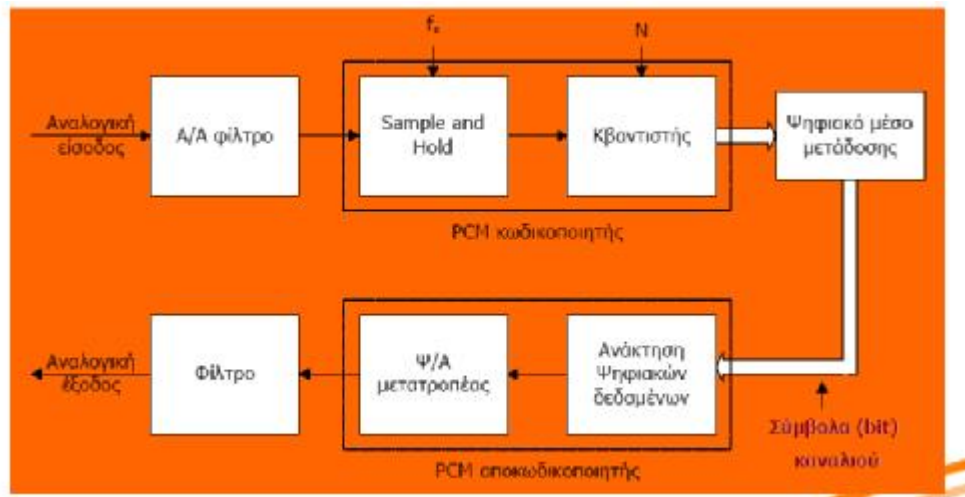
3.2 PCM (PulseCodeModulation)

Μια από τις πιο απλές και ευρέως μεθόδους κωδικοποίησης ψηφιακού ήχου είναι η παλμοκωδική κωδικοποίηση PCM. Στην μέθοδο αυτή κάθε δείγμα αναπαριστάται με ένα σύνολο παλμών που αντιστοιχούν στον δυαδικό κώδικα και στην τιμή του δείγματος με επακόλουθο να αποθηκεύεται ένα προς ένα τα δείγματα σε ψηφιακή μορφή χρησιμοποιώντας γραμμική κωδικοποίηση. Όπως είναι αναμενόμενο, η πιστότητα του σήματος που προκύπτει είναι συνάρτηση του δυαδικού κώδικα, π.χ. ένα δυναμικό εύρος 128 τιμών θα έχει ως αποτέλεσμα την

απάλειψη ήχων στο ψηφιοποιημένο σήμα με ένταση ίση ή μικρότερη από το $1/128$ της έντασης του δυνατού ήχου που μπορεί να αναπαρασταθεί από το σήμα. Στο σημείο αυτό, ανακύπτει το ερώτημα αν η πιστότητα σε ένταση σε ασθενή ηχητικά σήματα εισάγει πρόβλημα. Προφανώς εισάγει και για αυτό το λόγο έχουν αναπτυχθεί άλλες τεχνικές οι οποίες χρησιμοποιούν λογαριθμική κωδικοποίηση PCM όπου και αναλύονται στις επόμενες παραγράφους. Τέτοιες τεχνικές, χρησιμοποιούν περισσότερα δυαδικά ψηφία για την αναπαράσταση ασθενών ήχων και λιγότερα για τους ήχους μεγάλης έντασης στους οποίους η ευαισθησία της ανθρώπινης ακοής σε σχετικές διαφορές έντασης είναι μειωμένη (βλ. προβλήματα δειγματοληψίας).



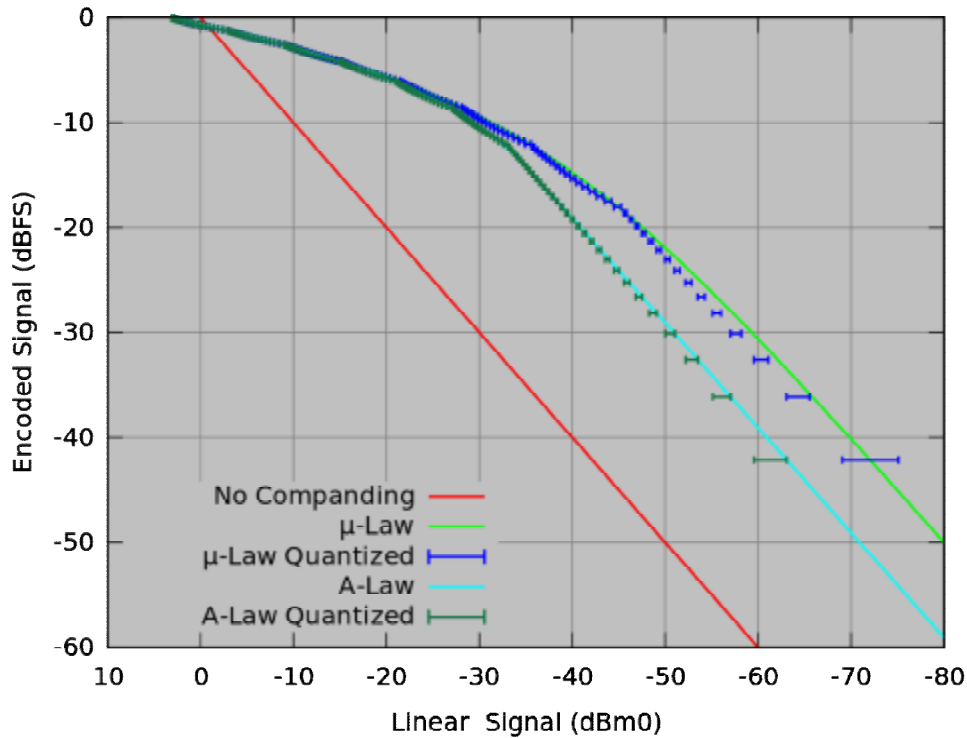
Εικόνα 3-1 Κωδικοποίηση PCM



Εικόνα 3-2PCM κωδικοποίηση

3.3 Mu-Law PCM και A-Law PCM.

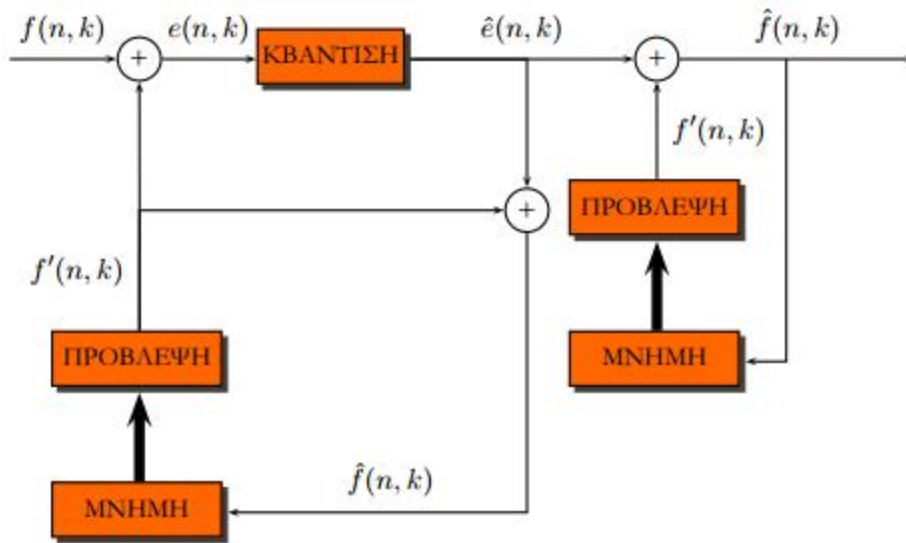
Η διαφορά της λογαριθμικής κωδικοποίησης ήχου σε σχέση με την γραμμική έγκειται στο γεγονός ότι κατά την λογαριθμική αντιστοίχιση, αντιστοιχούνται ολοένα και περισσότερες στάθμες σε χαμηλές συχνότητες και λιγότερες στις υψηλές, ενώ ταυτόχρονα πραγματοποιείται τόσο καλύτερη αναπαράσταση όσο και καλύτερη συμπίεση του σήματος με αυτό της παλμοκωδικής, η οποία δεν πραγματοποιεί καμία συμπίεση ούτως ώστε να προκύπτει ένα αρχείο ήχου κωδικοποιημένο χωρίς καμία απώλεια. Πιο συγκεκριμένα, 8bits σε συνδυασμό με λογαριθμική κωδικοποίηση καλύπτει το ίδιο εύρο τιμών 14bits και παλμοκωδική κωδικοποίηση PCM, επομένως πρόκειται για μια συμπίεση της τάξης του 1,75 προς 1. Κατά συνέπεια, στην κατηγορία αυτή της λογαριθμικής κωδικοποίησης ήχου εντάσσονται οι Mu-LawPCM και A-LawPCM, δύο μέθοδοι που έχουν τυποποιηθεί από το ITU-T (International Telecommunication Union Telecommunication Standardization Sector) στο πρότυπο G711, Pulse Code Modulation of noise Frequencies το οποίο και περιγράφει λεπτομερές τα δύο είδη κωδικοποίησης. Συγκεκριμένα, η Mu-Law υλοποιείται σε ISDN (Integrated Services Digital Network) δίκτυα της Ιαπωνίας και Β. Αμερικής ενώ η A-Law σε δίκτυα των υπόλοιπων χωρών.

Εικόνα 3-3 Σύγκριση των αλγορίθμων μ -law και A-law

3.4 DPCM (Differential Pulse Code Modulation)

Σε σχέση με την PCM κωδικοποίηση ψηφιακού σήματος, η διαφορική παλμοκωδική κωδικοποίηση δεν κωδικοποιεί το κάθε δείγμα ανεξάρτητα από τα υπόλοιπα αλλά σε συνάρτηση με τα γειτονικά δείγματα, αποθηκεύοντας τις διαφορές μεταξύ των διαδοχικών τιμών και όχι τις απόλυτες τιμές των δειγμάτων. (Μπούρας) Δηλαδή για την χρονική στιγμή t θα κωδικοποιηθεί η διαφορά των δειγμάτων $\delta_t - \delta_{t-1}$ όπου ως προβλεπόμενη τιμή έχει χρησιμοποιηθεί η τιμή του δείγματος δ_{t-1} την χρονική στιγμή $t-1$. Η βασική έννοια DPCM που κωδικοποιεί μια διαφορά, στηρίζεται στο γεγονός ότι τα περισσότερα σήματα πηγής παρουσιάζουν σημαντικό συσχετισμό μεταξύ των διαδοχικών δειγμάτων που κωδικοποιούν, ούτως ώστε οι τιμές των δειγμάτων να διαμορφώνουν χαμηλό ποσοστό δυαδικών ψηφίων. Για την υλοποίηση της βασική έννοιας θα πρέπει να προβλέψουμε την τρέχουσα αξία

των δειγμάτων η οποία βασίζεται στα προηγούμενα δείγματα καθώς και να κωδικοποιήσουμε την αξία του δείγματος και την προσληφθείσα αξία.



Εικόνα 3-4Κωδικοποιητής και αποκωδικοποιητής ενός συστήματος DPCM

Μια ειδική μορφή της διαφορικής παλμοκωδικής κωδικοποίησης είναι η DM (Delta Modulation), ως απλουστευμένη DPCM κατά την οποία η διαφορά της προβλεπόμενης και της τρέχουσας τιμής του δείγματος κωδικοποιείται με ένα μόνο bit παίρνοντας τιμές σύμφωνα με την **αρχή DM**: Η παραγωγή DM είναι 0 εάν το τρέχον δείγμα έχει μικρότερο εύρος από το προηγούμενο και 1 εάν το τρέχον δείγμα έχει εύρος μεγαλύτερο από το αμέσως προηγούμενο, με επακόλουθο το DM να κωδικοποιεί την κατεύθυνση των διαφορών στο εύρος σήματος αντί την αξία της διαφοράς DPCM. Αυτό σημαίνει ότι κάθε δείγμα μπορεί να είναι είτε μεγαλύτερο είτε μικρότερο κατά ένα κβάντο από το προηγούμενο του με αποτέλεσμα ο περιορισμός να οδηγεί σε μεγάλη οικονομία αλλά αν το σήμα αλλάζει γρήγορα θα υπάρχει μεγάλη απώλεια πληροφορίας.

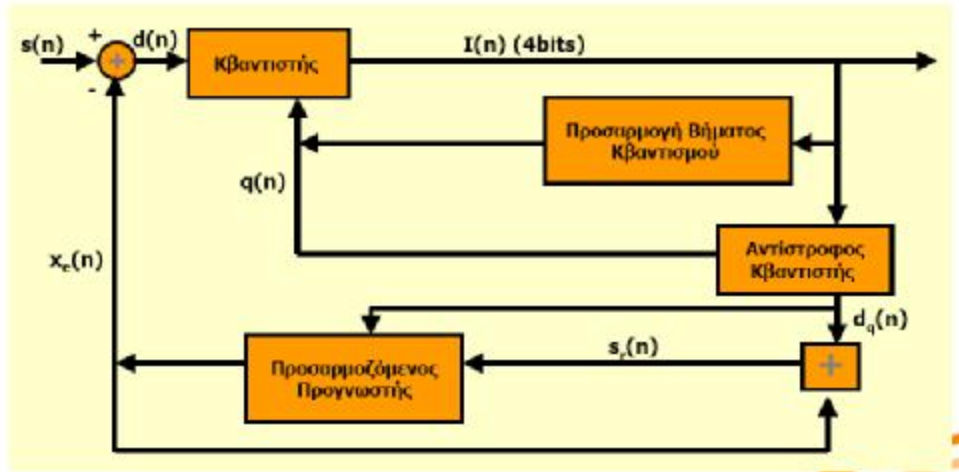
3.5 ADPCM (Adaptive Differential Pulse Code Modulation)

Η διαφορική παλμοκωδική κωδικοποίηση, αποτελεί μια επέκταση της DPCM μεθόδου κατά την οποία προβλέπεται η τιμή ενός δείγματος με βάση την τιμή του γειτονικού δείγματος και αυτό επειδή τα γειτονικά δείγματα πολύ πιθανόν να είναι όμοια αν όχι και ίδια. Συνεπώς, η ADPCM κωδικοποίησης, υπολογίζει την διαφορά μεταξύ της πραγματικής τιμής του δείγματος και της τιμής που είχε προβλεφθεί και κωδικοποιείται με συνεπακόλουθο, η διαφορά της τιμής να είναι σχετικά ελάχιστη (4bits) απαιτώντας λιγότερα bits. Στο σημείο αυτό, ένα σημαντικό μειονέκτημα αναδύεται ανάμεσα στις δύο μεθόδους κωδικοποίησης ADPCM και DPCM και αυτό εξαιτίας της εξάρτησης του μεγέθους των διαφορών μεταξύ διαδοχικών δειγμάτων ενός ηχητικού σήματος σε σχέση με την συχνότητα.

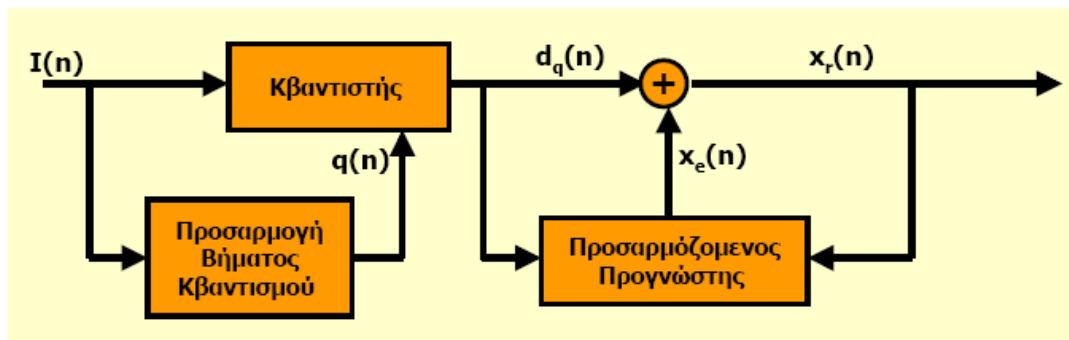
Όπως είναι αντιληπτό, οι διαφορές μεταξύ διαδοχικών δειγμάτων σε σχέση με ήχους χαμηλών συχνοτήτων, είναι σημαντικά μικρότερες από τις αντίστοιχες διαφορές σε ήχους υψηλών συχνοτήτων, με αποτέλεσμα η κωδικοποίηση αυτών των διαφορών να εντάσσεται στο συχνοτικό περιεχόμενο του σήματος της κάθε χρονικής στιγμής, γεγονός που δεν πραγματοποιείται στις παραπάνω μεθόδους. Αντίθετα, οι μέθοδοι *κωδικοποίηση υποζώνης (subbandcoding)*, κατηγοριοποιούν το ηχητικό σήμα σε δύο ή περισσότερες ζώνες συχνοτήτων και συμπιέζουν κάθε μια από αυτές ξεχωριστά με σκοπό μια τέτοια κατηγοριοποίηση να εκμεταλλεύεται τα χαρακτηριστικά της ανθρώπινης ακοής που παρουσιάζει την μέγιστη ευαισθησία στο εύρος συχνοτήτων μεταξύ 2700-3200Hz με την ευαισθησία να ελαττώνεται όσο απομακρυνόμαστε από την ζώνη αυτή.

Συνοψίζοντας τα πιο πάνω, ανάγεται το συμπέρασμα, πως σε μια τέτοια μέθοδο κωδικοποίησης υποζώνης, τα αποτελέσματα της δεν γίνονται αντιληπτά από την ανθρώπινη ακοή και αυτό επιτυγχάνεται, γιατί έχει την ικανότητα τόσο να συμπιέζει ήχους με συχνοτικό περιεχόμενο που απέχουν

από την παραπάνω ζώνη όσο και να εφαρμόζει την ελάχιστη συμπίεση που ανήκει στην ζώνη αυτή. Σε μια τέτοια κατηγορία, ανήκουν τα πρότυπα MPEGaudio, DolbyAC-2 και AC-3 και το RealAudio για τα οποία εκτεταμένη αναφορά γίνεται στο παρακάτω κεφάλαιο.



Εικόνα 3-5 Κωδικοποιητής ADPCM



Εικόνα 3-6 Αποκωδικοποιητής ADPCM

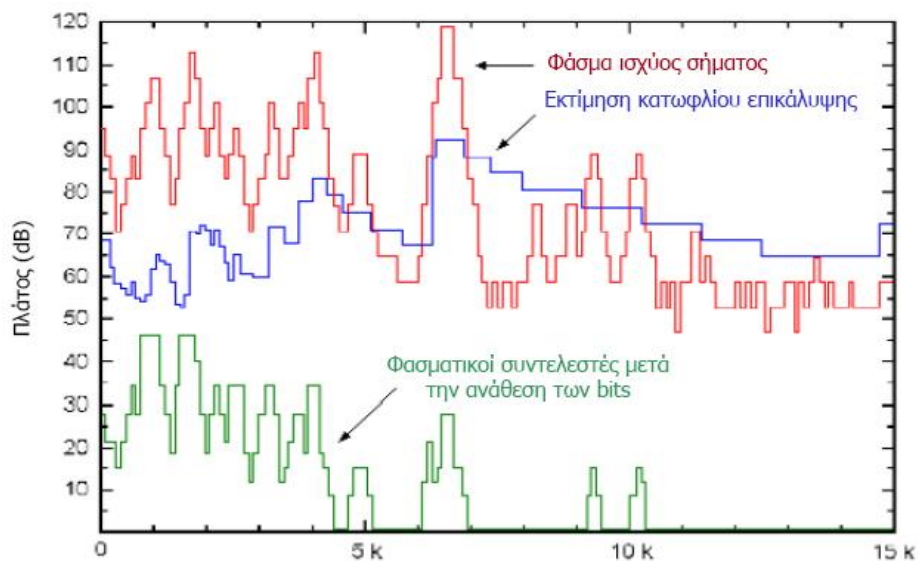
4. «Συμπίεση»

4.1 Συμπίεση Ψηφιακού Ήχου

Με δεδομένη την συνεχώς αυξανόμενη χρήση της ψηφιακής τεχνολογίας η οποία αντικαθιστά την αναλογική που παραδοσιακά χρησιμοποιείται, εδώ και δεκαετίες στις τηλεπικοινωνίες, την ηχογράφηση και αναπαραγωγή της μουσικής αλλά και σε άλλους τομείς έχει αρχίσει παράλληλα να γίνεται έρευνα προς την κατεύθυνση της συμπίεσης ψηφιακής πληροφορίας και στην περίπτωση μας του ψηφιακού ηχητικού σήματος με στόχο τη οικονομία εύρους φάσματος. Σύμφωνα με αυτά που έχουν αναφερθεί στα παραπάνω κεφάλαια, παρατηρείται ότι κατά την διαδικασία της ψηφιοποίησης ηχητικών δεδομένων, παράγονται αρχεία μεγάλων μεγεθών τα οποία δύσκολο να μεταφερθούν ή να υποστούν επεξεργασία αφού προϋποθέτουν πολύ χρόνο, μεγάλους αποθηκευτικούς χώρους και πολύ μνήμη.

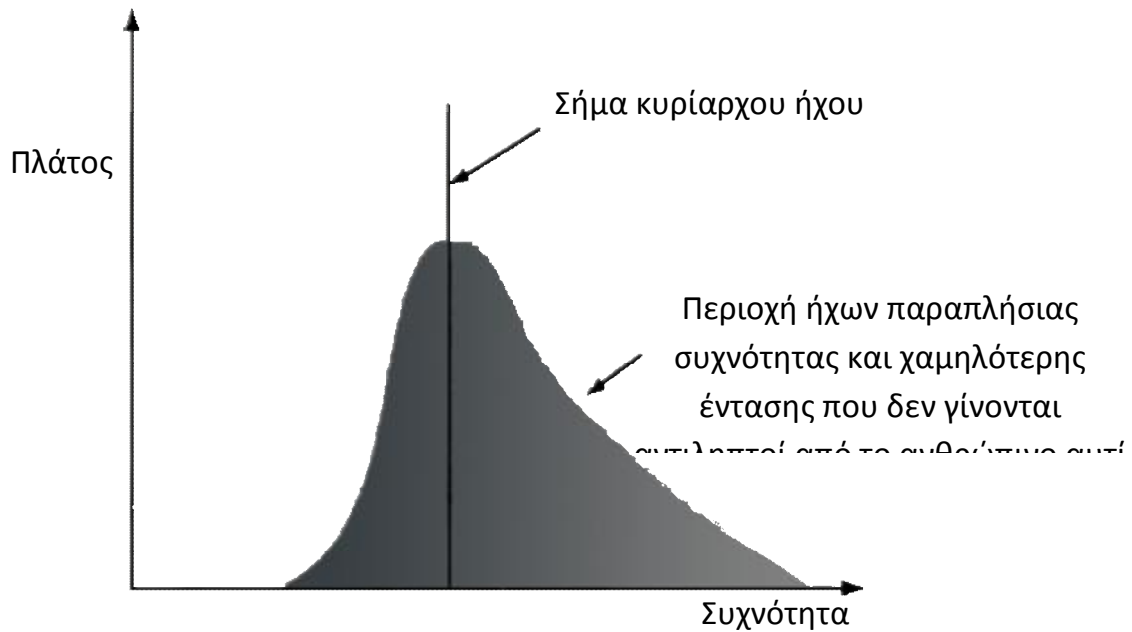
Το τεχνικό πρόβλημα του χώρου και της ανεπάρκειας στην ταχύτητα μεταφοράς έρχονται να λύσουν οι τεχνολογίες συμπίεσης του ηχητικού σήματος. Οι τεχνολογίες αυτές, χρησιμοποιούν διάφορες τεχνικές μείωσης του όγκο και της ροής των δεδομένων που απαιτούνται για την κωδικοποίηση των ηχητικών σημάτων και διακρίνονται σε δύο κατηγορίες, τις **μη απωλεστικές (lossless)** και τις **απωλεστικές (lossy)**. Οι μη απωλεστικές, έχουν το ιδιαίτερο χαρακτηριστικό ότι η διαδικασία συμπίεσης δεν αλλοιώνει καθόλου την πληροφορία δηλαδή μετά την αποσυμπίεση η πληροφορία επανέρχεται στην ακριβώς στην αρχική της μορφή. Συνήθως, οι μη απωλεστικοί αλγόριθμοι συμπίεσης ψηφιακού ήχου, εφαρμόζονται σε περιπτώσεις που δεν υπάρχει κανένα περιθώριο απωλειών. Αντίθετα, οι απωλεστικές τεχνικές αλλοιώνουν τα δεδομένα, απορρίπτοντας τα μέρη εκείνα που δεν είναι χρήσιμα με βάση κάποιο συγκεκριμένο κριτήριο (irrelevancy) . Είναι φανερό, ότι σε τέτοιες περιπτώσεις το σημασιολογικό περιεχόμενο ουσιαστικά δεν μεταβάλλεται αλλά υπεισέρχεται η έννοια της

μείωσης της ποιότητας καθώς οι απωλεστικές τεχνικές είναι περισσότερο αποτελεσματικές επιτυγχάνοντας συμπίεση που φθάνει το 24:1 με καλή ποιότητα ενώ οι μη απωλεστικές σπανίως ξεπερνούν το 4:1. Εντούτοις, στις σύγχρονες μεθόδους συμπίεσης εφαρμόζονται διάφορα ψυχοακουστικά μοντέλα αντίληψης βάση των οποίων απορρίπτονται οι ήχοι που έτσι και αλλιώς δεν μπορούν να γίνουν κατανοητοί από το ανθρώπινο αυτί.



Εικόνα 4-0-1 Ψυχοακουστικό Μοντέλο Ακοής

Οι σημαντικότεροι μέθοδοι συμπίεσης ψηφιακών ηχητικών αρχείων στηρίζονται στο φαινόμενο της ηχητικής σκίασης (auditory masking), που αποσκοπεί στην ακουστικότητα των ήχων. Πιο συγκεκριμένα, όταν υπάρχει ένας ήχος συγκεκριμένης συχνότητας και έντασης, άλλοι ήχοι σε κοντινές συχνότητες δεν γίνονται αντιληπτοί από το ανθρώπινο αυτί. Παράλληλα, κατά την συμπίεση των ψηφιακών ηχητικών δεδομένων απορρίπτονται οι συχνότητες που το ανθρώπινο αυτί δεν μπορεί να ξεχωρίσει μέσα σε ένα καθορισμένο διάστημα συχνοτήτων. Αν και χάνονται ορισμένες ηχητικές πληροφορίες εντούτοις η ποιότητα του ήχου παραμένει σε υψηλά επίπεδα.



Εικόνα 4-0-2 Φαινόμενο Ηχητικής Σκίασης

Το φαινόμενο της ηχητική σκίασης, είναι άμεσα συνδεδεμένο τόσο με το πρότυπο AC-3 DolbyDigital (Η.Π.Α) όσο και με το Mpeg των οποίων οι διαφορές τους στηρίζονται στο τρόπο υλοποίησής τους. Εντούτοις και τα δύο πρότυπα συμπίεσης ηχητικών δεδομένων, δουλεύουν χωρίζοντας το φάσμα των ακουστικών συχνοτήτων σε υπομπάντες χρησιμοποιώντας περίπλοκους αλγόριθμους και ψυχοακουστικά μέσα για να απορρίψουν τις μη ακουστικές από το ανθρώπινο αυτί συχνοτήτες. Ο ανταγωνισμός μεταξύ των προτύπων AC-3 και Mpeg, φαίνεται να ευνοεί το μοντέλο AC-3 το οποίο στην πορεία θα αναδειχθεί ως το διεθνές πρότυπο για την συμπίεσης ψηφιακών ηχητικών δεδομένων.

Συμπερασματικά, όλες οι διαδικασίες συμπίεσης και αποσυμπίεσης υλοποιούνται μέσω κατάλληλων προγραμμάτων, όπου ο ρυθμός μετάδοσης των ψηφιακών δεδομένων είναι πολύ σημαντικός καθορίζοντας την ποιότητα του ήχου καθώς εξαρτάται τόσο από την συχνότητα δειγματοληψίας όσο και το μέγεθος του δείγματος. Για παράδειγμα, ο ήχος του μουσικού CD χρησιμοποιεί συχνότητες δειγματοληψίας 44.1 kHz με μέγεθος δείγματος 16bits παράγοντας ένα ρυθμό δεδομένων 1.4 Mbit/sec, ενώ αν μειωθεί η

συχνότητα δειγματοληψίας τότε θα χαθούν οι υψηλές συχνότητες του ήχου. Επομένως, οι περισσότεροι σύγχρονοι αλγόριθμοι συμπίεσης που χρησιμοποιούν ψυχοακουστικό μοντέλο, στηρίζονται στη μεταβολή των bit για να επιτύχουν την μεγαλύτερη συμπίεση διατηρώντας υψηλή την ποιότητα του αναπαραγόμενου ήχου.

4.2 Πρότυπο MPEG

Οι προσπάθειες για μετάδοση ψηφιακού DAB (Digital Audio Broadcasting), ξεκίνησε στην Ευρώπη από το 1987 με το πρόγραμμα Eureka, στο οποίο συμμετείχαν ερευνητές τόσο από το Ινστιτούτο Fraunhofer με επικεφαλής τους Brandenburg όσο και τα μέλη της ομάδας Moving Picture Expert Group (MPEG). Η MPEG, ως μια συνεργασία Πανεπιστημίων, ερευνητών ινστιτούτων και εταιριών λειτουργεί στα πλαίσια του Διεθνούς Οργανισμού Τυποποίησης γνωστή σαν ISO/IEC με στόχο την ανάπτυξη διεθνών πρότυπων για την συμπίεση και αποσυμπίεση, την επεξεργασία και την κωδικοποιημένη αντιπροσώπευση της κίνησης των εικόνων, του ήχου και τους συνδυασμούς τους.

Το όνομα MPEG, έχει επικρατήσει όμως, να αναφέρεται και στη οικογένεια των τυποποιήσεων (standards) που δημιουργήθηκαν από την ομάδα Mpeg και χρησιμοποιούνται για την μετάδοση οπτικών και ηχητικών δεδομένων σε ψηφιακή συμπιεσμένη μορφή. Στην οικογένεια Mpeg, εντάσσονται τα standards Mpeg-1 που αφορά την συμπίεση ήχου και εικόνας, το Mpeg-2 για την εφαρμογή στην ψηφιακή τηλεόραση και το Mpeg-4 ως standard για εφαρμογές επικοινωνίας πολυμέσων. Επίσης, υπάρχει στα σχέδια τους και το Mpeg-7, με στόχο την αναπαράσταση περιεχομένου (content representation) για την αναζήτηση πληροφοριών σε εφαρμογές. Στο σημείο αυτό, οφείλουμε να επισημάνουμε ότι τα δύο τελευταία στάδια του Mpeg βρίσκονται σε υπανάπτυξη και δεν έχουν γίνει ακόμη στάνταρ ενώ το

Mpeg 1 και 2 έχουν τεθεί ήδη σε εκτεταμένη εφαρμογή. Ενδιάμεσα, για αρκετό χρονικό διάστημα σε εξέλιξη υπήρξε το Mpeg 3 αλλά εγκαταλείφθηκε και ενσωματώθηκε ένα μέρος του στο Mpeg 2.

Πιο συγκεκριμένα, το πρότυπο που έγκειται στον ψηφιακό ήχο είναι το Mpeg-1 Audio, ως το πρώτο διεθνές πρότυπο για την ψηφιακή συμπίεση ήχου υψηλής πιστότητας που δεν αποτελεί ένα αλγόριθμο συμπίεσης αλλά μια οικογένεια τριών διαφορετικών τεχνικών κωδικοποίησης και συμπίεσης. Και τα τρία αυτά στάδια στηρίζονται στην ίδια αρχή, δηλαδή η συμπίεση ολοκληρώνεται με το συνδυασμό ενός είδος κωδικοποίησης μετασχηματισμού και sub-band division ενώ οι διαφορές του αναδύονται στο τελικό στάδιο της κβαντοποίησης. Παράλληλα, το πρότυπο Mpeg-1 Audio προβλέπει ένα ή δύο ηχητικά κανάλια χρησιμοποιώντας 16bits για την κωδικοποίηση των δειγμάτων, ενώ η συχνότητα δειγματοληψίας του ήχου μπορεί να είναι 32kHz, 44kHz ή 48 kHz.

Όπως προαναφέρθηκε, το πρότυπο Mpeg-1 Audio στηριζόμενο στην κωδικοποίηση ψηφιακού ήχου διακρίνεται σε Mpeg-1 Audio Layer I, II και III (ή MP3). Συγκεκριμένα, το Mpeg-1 Audio Layer I χρησιμοποιήθηκε στο σύστημα συμπίεσης ψηφιακής κασέτας DCC της Philips προσφέροντας συμπίεση 4:1. Ως αποτέλεσμα, η ηχητική ποιότητα είναι μέτρια ενώ το bandwidth που απαιτείται είναι αυξημένο 192 ή 256 kbps ανά κανάλι. Το Mpeg-1 Audio Layer II (ή mp2), χρησιμοποιήθηκε στο ψηφιακό ραδιόφωνο όπου ο αλγόριθμος αυτή της κατηγορίας έχει βελτιστοποιηθεί για ένα εύρος ζώνης 96 ή 128 kbps ανά μονοφωνικό κανάλι, ενώ ως αποτέλεσμα η ποιότητα είναι εφάμιλλη του CD (6:1...8:1 με 256...142 kbps για στερεοφωνικό ήχο) . Αντίθετα, το Mpeg-1 Audio Layer III ή αλλιώς mp3, έχοντας καλύτερη απόδοση από τα παραπάνω παρουσιάζει συμπίεση περίπου 12:1 με ποιότητα που πλησιάζει αυτή των CD και ρυθμό μετάδοσης δεδομένων στα 64 kbps. Σήμερα, με βάση αυτό το πρότυπο είναι κωδικοποιημένα τα αρχεία ήχου mp3 και τα οποία παρουσιάζονται τόσο για την μεταφορά όσο και για την

φόρτωση μέσω διαδικτύου όσο και για την αναπαραγωγή ή ανάκληση από το σκληρό δίσκο. Στον πιο κάτω πίνακα, παρουσιάζεται ο λόγος συμπίεσης σε συνάρτηση με το ρυθμό μετάδοσης των δεδομένων που υποστηρίζει το κάθε στρώμα του προτύπου Mpeg-1 Audio.

Πίνακας 3 - Λόγος συμπίεσης

Στρώμα	Λόγος Συμπίεσης
Mpeg-1 Audio I	4:1 με 384 kbps για στερεοφωνικό ήχο
Mpeg-1 Audio II	6:1 με 256-192 kbps για στερεοφωνικό ήχο
Mpeg-1 Audio III	12:1 με 128-112 kbps για στερεοφωνικό ήχο

Όπως έχει αναφερθεί παραπάνω, το πρότυπο Mpeg 1 μπορεί να κωδικοποιήσει μόνο δυο κανάλια ήχου ενώ για το δίκτυο που δεν διαθέτει μεγάλο εύρος ζώνης χρησιμοποιούνται κυρίως τεχνολογίες Mpeg 2. Το Mpeg 2 για τον ήχο, επεκτείνει την κωδικοποίηση μονοφωνικού και στερεοφωνικού ήχου του Mpeg 1, χωρίζεται σε τρία επίπεδα ανάλογα με το bitrate (ρυθμό δειγματοληψίας) που πρόκειται να υπάρξει ενώ κάθε επίπεδο δεν είναι καλύτερο από το άλλο, απλώς πιο περίπλοκο. Όσο προχωράμε στα επίπεδα τόσο πιο περίπλοκος γίνεται ο κωδικοποιητής και τόσο καλύτερη εκμετάλλευση του ρυθμού δειγματοληψίας πραγματοποιείται. Ο παρακάτω πίνακας, αναδύει αποτελέσματα τεστ σύγκρισης των τριών επιπέδων με κλίμακα από το 1 ως το 5.

Πίνακας 4 - Συμπίεση σε κάθε επίπεδο

Επίπεδο	Bitrate	Συμπίεση	Ελάχιστη καθυστέρηση	Ποιότητα 64 kbit
I	192 kbit	4:1	19ms	---
II	128 kbit	6:1	35ms	2.1 ως 2.6
III	64 kbit	12:1	59ms	3.6 ως 3.8

Αυτή την στιγμή το αγαπημένο της βιομηχανίας είναι το επίπεδο II αφού όταν σχεδίαζαν τους εξοπλισμούς τους δεν είχαν ακόμη οριστικοποιηθεί το επίπεδο III. Όμως με ολοένα και αυξανόμενους ρυθμούς το επίπεδο III κερδίζει θέση με πρωταρχικό χώρο το διαδίκτυο. Για δομημένη ποιότητα ήχου Mpeg LayerIII απαιτεί μικρότερο bitrate ή αλλιώς για δοσμένο bitrate πετυχαίνει υψηλότερη ποιότητα ήχου. Συμπερασματικά, το πρότυπο προσφέρει δειγματοληψία ελαττωμένη κατά το ήμισυ (16 kHz, 22kHz και 24kHz), παρέχοντας βελτιωμένη ποιότητα για ρυθμούς μετάδοσης σε 64 Kbit/sec.

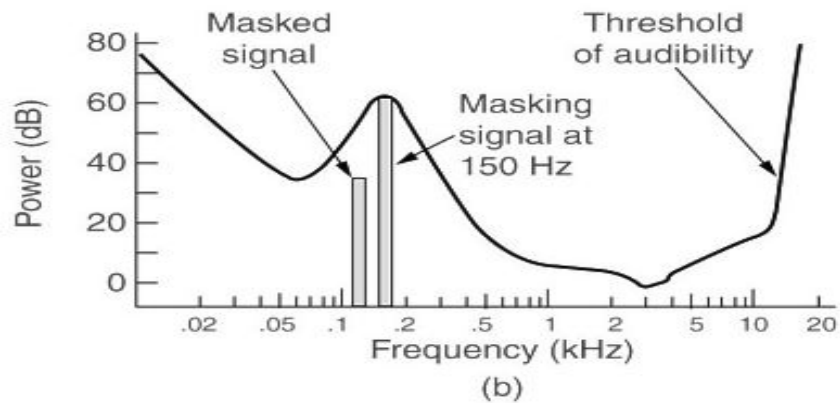
4.3 MP3

Το MP3 καταγράφηκε στην Ιστορία, ως μια από τις πιο θορυβώδεις καινοτομίες της ψηφιακής μουσικής. Προκάλεσε χαρά στους μουσικόφιλους και πανικό στις δισκογραφικές εταιρίες, ταρακούνησε θεσμούς και αμφισβήτησε τα δεδομένα της εποχής τα οποία τελικά άλλαξαν άρδην. Η ψηφιακή μουσική είναι άρρηκτα συνδεδεμένη με το MP3, πράγμα που διαπιστώνει κανείς από το γεγονός ότι οι νέες συσκευές κατακλύζουν ριζικά και ραγδαία την αγορά.

Όπως αναφέρθηκε, το Mpeg LayerIII ή MP3 όπως είναι παγκόσμια γνωστό έχει αναπτυχθεί από τους ερευνητές του Ινστιτούτου FraunhoferIIS με επικεφαλής, τους Karlheinz Branderburg και Dieter ως το πιο διαδεδομένο πρότυπο συμπίεσης ψηφιακού ήχου με στόχο την μεγαλύτερη συμπίεση και καλύτερη ποιότητα ήχου. Συγκεκριμένα, η συμπίεση των αρχείων ήχου μπορεί να φτάσει έως και 85%, χωρίς την ύπαρξη αισθητής διαφοράς στην ποιότητα απόδοσης ενώ αν η συμπίεση γίνει με ρυθμό μετάδοσης άνω των 128 Kbit/sec η διάφορά από το πρωτότυπο CD δεν είναι αντιληπτή. Αποτέλεσμα, η ύπαρξη μουσικών αρχείων 4-6 λεπτών που κανονικά θα καταλάμβαναν 40-70MB να καταλαμβάνουν χώρο μόνο 3-7MB.

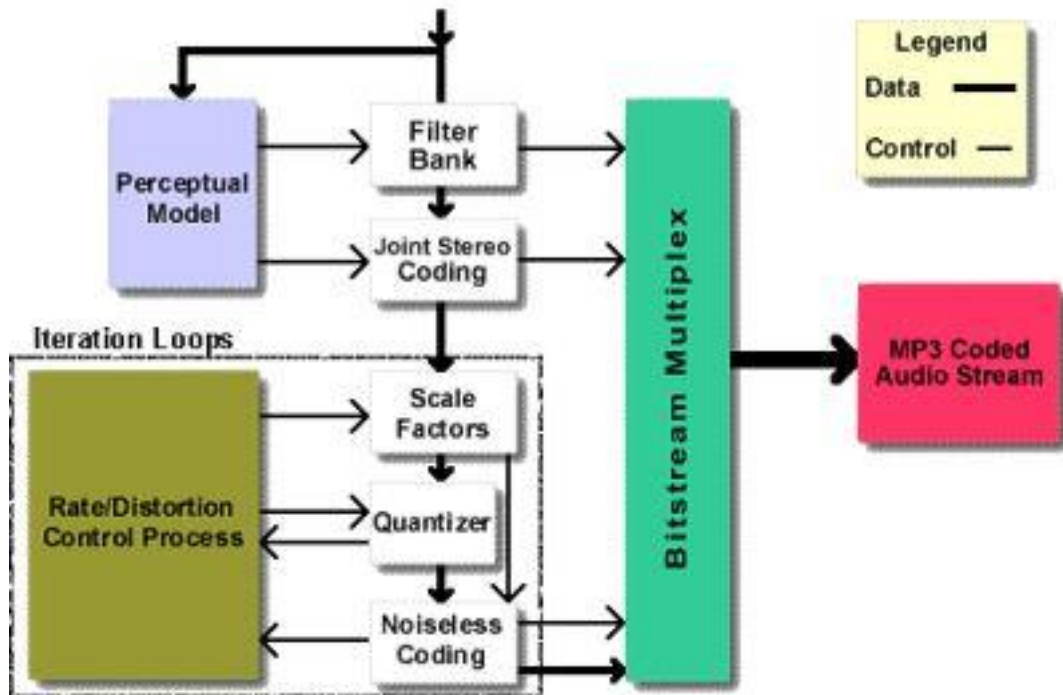
Οι codes MP3, χρησιμοποιούν ως βασικό μοντέλο ακοής αυτό που βασίζεται στις **καμπύλες κατώφλιού ακουστικότητας** (Minimal Audition Threshold) , δηλαδή την ελάχιστη ένταση που πρέπει να έχει ένας ήχος δια να τον ακούμε. Όσο μικρότερη είναι η ένταση, τόσο μικρότερη η ευαισθησία της ακοής και τόσο μεγαλύτερη στάθμη απαιτείται για να γίνει ακουστή μία συγκεκριμένη συχνότητα. Εμπειρικά αποτελέσματα, έχουν δείξει ότι το ανθρώπινο αυτί έχει μια περιορισμένη και εξαρτώμενη από την συχνότητα διακριτικότητα με επακόλουθο το κατώφλι ακουστικότητας να εξαρτάται από την ένταση του σήματος μέσα σε ένα περιορισμένο εύρος ζώνης γειτονικό αυτό της συχνότητας. Για κάθε συχνότητα του ακουστικού φάσματος το κατώφλι ακουστικότητας είναι διαφορετικό.

Ωστόσο, πολύ μεγαλύτερο ενδιαφέρον παρουσιάζει και το **φαινόμενο την επικάλυψης** (masking), με δεδομένους δυο γειτονικούς ήχους ο ισχυρότερος αλλοιώνει τοπικά την καμπύλη ακουστικότητας επικαλύπτοντας τον ασθενέστερο ο οποίος δεν γίνεται αντιληπτός από το ανθρώπινο αυτί. Έτσι το μοντέλο που χρησιμοποιείται στον codec MP3 υλοποιεί αυτό, δηλαδή υπολογίζει σε κάθε στιγμή το φασματικό περιεχόμενο του σήματος αποφασίζοντας ποιοι ήχοι επικαλύπτονται από ισχυρότερους με αποτέλεσμα να μην κωδικοποιηθούν. Σε κάθε χρονική στιγμή, ο codec MP3 έχει στη διάθεση του ένα αριθμό ψηφίων ο οποίος εξαρτάται από το βαθμό συμπίεσης που του έχει ζητηθεί, ενώ όσο εξελίσσεται η συμπίεση δεν χρησιμοποιούνται όλα τα ψηφία, ιδιαίτερα αν η στιγμιαία μορφή του σήματος είναι εύκολα συμπιεσμένη με συνεπακόλουθο την συντήρηση μιας δεξαμενής ψηφίων που απαιτούν μεγαλύτερη ακρίβεια στην κωδικοποίηση.



Εικόνα 4-3 Φαινόμενο επικάλυψης. Ένας δυνατός ήχος, μεταβάλλεται τοπικά στην καμπύλη του κατωφλιού ακουστικότητας επικαλύπτοντας ένα γειτονικό αδύναμο ήχο.

Στο σημείο αυτό, οφείλουμε να αναφέρουμε ότι για καλύτερη συμπίεση δεδομένων το MP3 για την κωδικοποίηση των κβαντισμένων δειγμάτων χρησιμοποιεί την **εντροπική κωδικοποίηση** (entropy encoding), η οποία κωδικοποιεί τα ψηφία που προκύπτουν από τα προηγούμενα στάδια. Η εντροπική κωδικοποίηση αποκαλείται και ως Huffman Coding. Συμπερασματικά, το ψυχοακουστικό μοντέλο για την συμπίεση ψηφιακού ήχου στην περίπτωση του MP3 τρέχει από το πεδίο του χρόνου σε αυτό της συχνότητας. Για υλοποίηση του περάσματος αυτού, χρησιμοποιείται μια πολύπλοκη μαθηματική διαδικασία κατά την οποία το υπό κωδικοποίηση μέρος του σήματος, υπόκεινται σε επεξεργασία από μια τράπεζα φίλτρων (filter bank) όπου και χωρίζει το φάσμα σε 32 περιοχές και ακολουθεί ο μετασχηματισμός MDCT (Modified Discrete Cosine Transform), με στόχο την καλύτερη φασματική διακριτότητα. Η διαδικασία αυτή, στηρίζεται στην αρχή του μετασχηματισμού Fourier όπου και αναλύει ένα σήμα εξελισσόμενο στο χρόνο σε μια σειρά συνιστωσών στο πεδίο της συχνότητας.



Εικόνα 4-4 Διάγραμμα βαθμίδων κωδικοποιητή MP3

Εικόνα 4-5 Μαθηματική περιγραφή του μετασχηματισμού MDCT

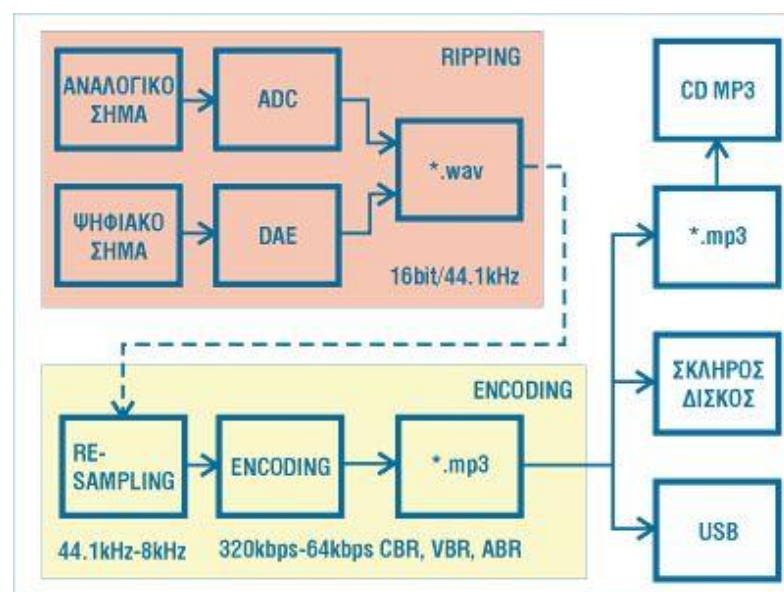
$$f = \sum_{k=0}^{2n-1} x \left(\cos \left(j + \frac{1}{2} \right) \left(k + \frac{1}{2} + n/2 \right) \right)$$

Δημιουργία και Αναπαραγωγή Αρχείου MP3 :

Προτού εστιάσουμε το ενδιαφέρον μας στην δημιουργίαMP3, οφείλουμε να αναφέρουμε ότι σημαντικό χαρακτηριστικό τους είναι πως δεν έχουν όλα την ίδια συμπίεση, με αποτέλεσμα το ποσοστό συμπίεσης του κάθε αρχείου να καθορίζεται από εμάς. Όσο μεγαλύτερο είναι το ποσοστό συμπίεσης τόσο μικρότερο θα είναι το αρχείο, ενώ όσο περισσότερο συμπιέζεται ο ήχος τόσο περισσότερη πληροφορία χάνεται, γεγονός που επηρεάζει την ποιότητα του ήχου. Το μέγεθος και η ποιότητα του αρχείου ήχου ρυθμίζεται από το ροή μετάδοσης των δεδομένων.

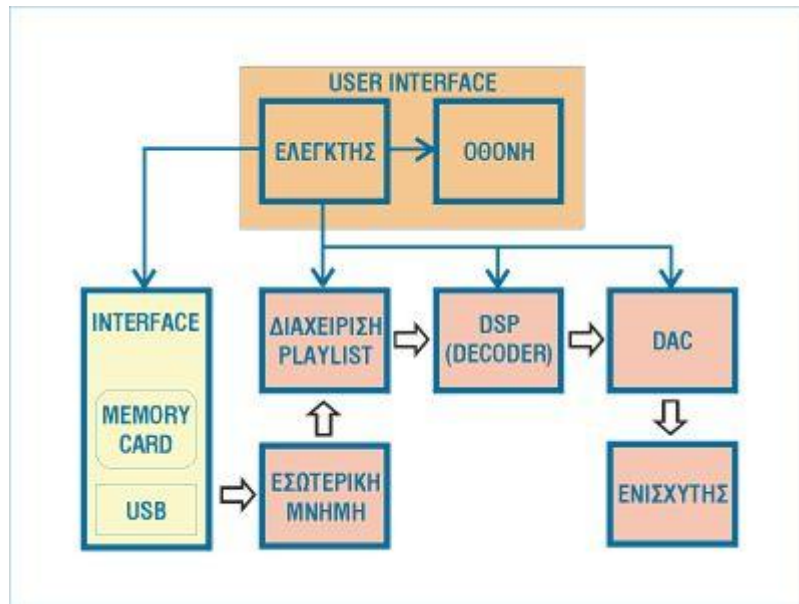
Συγκεκριμένα, την **δημιουργία ενός αρχείου MP3** υποδηλώνει η διαδικασία **ripping**, επιτρέποντας την δημιουργία ενός αρχείου wav που αποθηκεύεται στον υπολογιστή, από το πρωτογενές υλικό. Αν το υλικό αυτό

είναι αναλογικό, θα πρέπει πρώτα να περάσει από έναν μετατροπέα A/D, αντίθετα αν το υλικό είναι αποθηκευμένο σε CD, τότε μπορούμε να χρησιμοποιήσουμε την διαδικασία Digital Audio Extraction (DAE) καθώς και να μεταφέρουμε το ψηφιακό περιεχόμενο του δίσκου απευθείας σε αρχείο wav. Το αρχείο wav που προκύπτει από τον ripper έχει προδιαγραφές που εξαρτώνται από την διαδικασία CD Quality, δηλαδή συχνότητα δειγματοληψίας 44.1kHz, και μήκος λέξης 16bit. (χωρίς να αποκλείονται και άλλες εκδοχές, όπως τα 48kHz ή τα 24bit). Αυτό το αρχείο, είναι η πηγή των δεδομένων που τοποθετούνται στην είσοδο του encoder το οποίο πραγματοποιεί resampling (με βάση τις οδηγίες που του δίνουμε), κωδικοποιώντας το σήμα με βάση το μοντέλο της απωλεστικής συμπίεσης με αποτέλεσμα την δημιουργία ενός αρχείο MP3. Το αρχείο αυτό, μπορεί να αποθηκευτεί τοπικά και να αναπαράγεται μέσω του σχετικού player, της κάρτας ήχου και των ηχείων του υπολογιστή, να μεταφερθεί σε κάποια εξωτερική συσκευή μέσω USB, να μετατραπεί και πάλι σε wav με απώτερο στόχο την εγγραφή του σε CD που είναι συμβατό με απλά CDplayers ή να εγγραφεί απ' ευθείας σε CD με στόχο να χρησιμοποιηθεί από συσκευές που είναι συμβατές με CDMP3.



Εικόνα 4-6 Διαδικασία Ripping και Encoding

Η διαδικασία *αναπαραγωγή αρχείων MP3* πραγματοποιείται όταν αυτά βρίσκονται αποθηκευμένα στον υπολογιστή. Καταρχήν τα αρχεία με κατάληξη mp3 μέσω των "File Associations", αναδύουν με διπλό κλικ το media player ούτως ώστε να μην χρειάζεται άλλη προσπάθεια από την πλευρά του χρήστη. Στην περίπτωση που τα αρχεία βρίσκονται στον υπολογιστή και που πρέπει να φορτωθούν σε κάποια εξωτερική συσκευή player, αυτό γίνεται χωρίς ειδικό λογισμικό αφού οι μνήμες και όλο και πιο συχνά τα players φαίνονται από την πλευρά του υπολογιστή ως "removable storage device" όταν συνδέονται στο USB. Αντίθετα, από την πλευρά του player τα πράγματα δεν είναι σύνθετα, ένα user interface παρέχει πρόσβαση τόσο στην δημιουργία όσο και διαχείριση ενός καταλόγου με τα περιεχόμενα του player είτε αυτά βρίσκονται σε εξωτερική είτε σε εσωτερική μνήμη. Στο σημείο αυτό, οφείλουμε να αναφέρουμε ότι η διαδικασία την διαχείριση είναι απλή ή σύνθετη ανάλογα με τις δυνατότητες της συσκευής, ενώ από την στιγμή που ο χρήστης επιλέξει το τρακ και στην συνέχεια το play, τα δεδομένα του αντίστοιχου αρχείου mp3 οδηγούνται, στον αποκωδικοποιητή, στον μετατροπέα d/a και στον ενισχυτή ακουστικών για περισσότερη επεξεργασία. Εντούτοις, πολλά players διαθέτουν και αναλογικές εισόδους έτσι ώστε ο ψηφιακός επεξεργαστής να διαθέτει μετατροπέα A/D και MP3 encoder.



Εικόνα 4-7 Διάγραμμα βαθμίδων ενός MP3

Επίσης, καλό θα ήταν να τυπωθεί ότι, το πιο γνωστό και διαδεδομένο πρόγραμμα αναπαραγωγής μουσικών αρχείων MP3 είναι το Winamp της εταιρίας Nullsoft, το οποίο και διανέμεται δωρεά και είναι εγκατεστημένο σε εκατομμύρια υπολογιστές του κόσμου. Εκτός από το Winamp, μπορεί να βρει κανείς στο διαδίκτυο και άλλα προγράμματα αναπαραγωγής MP3 όπως τα MediaJukeBox, MusicMatch, Sonique, Ejay κ.α. Ο μεγαλύτερος όμως συναγωνιστής του Winamp είναι ο MediaPlayer της Microsoft, ο οποίος διαχειρίζεται αποκωδικοποιητές MP3 του Ινστιτούτου Fraunhofer με καλύτερη ποιότητα ήχου, ενώ οι περισσότεροι MP3 player μπορούν να αναπαράγουν και άλλα πρότυπα ψηφιακής μουσικής όπως Wav, CD Audio, WMA κ.α., των οποίων ανάλυση γίνεται παρακάτω.

4.4 Άλλα πρότυπα

Παρά το γεγονός της η ριζική και ραγδαία εξέλιξη του MP3 το οποίο έχει γνωρίσει ευρεία αποδοχή μεταξύ χρηστών, εντούτοις, επικρατεί πληθώρα διαφορετικών τεχνολογιών που ολοένα και εξελίσσονται με πρωταρχικό στόχο την καλύτερη συμπίεση ψηφιακού ήχου. Τέτοιες

προσπάθειες, έχουν υλοποιηθεί από την εταιρία Microsoft με το δικό της γνωστό και διαδεδομένο πρότυπο Windows Media Audio (WMA), το AAC (Advanced Audio Coding), το MP3 Pro, το OGG και το AC-3 Dolby Digital. (ΤΑΛΙΕΡΗΣ, 2015)

4.4.1 WMA (Windows Media Audio)

Η μεταφορά, η αποθήκευση και η χρήση ακουστικού υλικού με βάση την μορφή απωλεστικών συμπιεσμένων ηχητικών αρχείων μέσω υπολογιστή, ωθεί τον μεγαλύτερο κατασκευαστή λειτουργικών συστημάτων μα μην μείνει έξω από το παιχνίδι. Το πρότυπο Windows Media Audio (WMA) της εταιρείας Microsoft, προσφέρει όμοιες δυνατότητες με τοMP3, με άριστη ποιότητα τόσο αναπαραγωγής όσο και μεγαλύτερη συμπίεση (64 kbps).Πιο συγκεκριμένα, το WMA αποτελεί ένα σύστημα κωδικοποίησης/αποκωδικοποίησης ήχου, επιτρέποντας την συμπίεση ψηφιακών δεδομένων ήχου στο 1/20 του αρχικού τους όγκου και την εγγραφή τους σε ένα μόνο δίσκο CD με επακόλουθο τα τραγούδια που είναι προστατευμένα να μην μπορούν να μεταδοθούν ελεύθερα. Συμπερασματικά ,για αυτόν ακριβώς το λόγο, ο μεγαλύτερος αριθμός δισκογραφικών εταιριών χρησιμοποιεί στα πλαίσια υλοποίησης του έργου τους το πρότυπο αυτό.

4.4.2 MP3 Pro

Τον Ιανουάριο του 2001 στη CES, παρουσιάστηκε από την Coding Technologies η τεχνολογία mp3Pro, μια βελτιωμένη έκδοση του mp3 με δυνατότητα να προσφέρει όμοια ποιότητα στο μισό μέγεθος των αρχείων,

γεγονός που υλοποιείται με μεγαλύτερη συμπίεση δεδομένων. Συγκεκριμένα, η συμπίεση στα 64kbps και 96kbps, προσφέρει τη ίδια απόδοση ήχου με τα 128kbps και 192kbps του MP3.

4.4.3 MP3 Surround

Το 2004 το Ινστιτούτο Fraunhofer IIS παρουσίασε μία πολυκαναλική έκδοση του MP3, το MP3 Surround το οποίο βασίζεται στην τεχνολογία Binaural Cue Coding της Agere. Η τεχνολογία αυτή, επιτρέπει την μείξη σημάτων από πολλά κανάλια σε δύο, με στόχο την δημιουργία ενός σήματος συμβατού με τον απλό MP3 codec, ενώ κωδικοποιεί μία σειρά από παραμέτρους που περιγράφουν πλήρως το ηχητικό πεδίο surround. Τέτοιες παράμετροι είναι, οι χρονικές διαφορές μεταξύ των καναλιών, οι διαφορές στάθμης μεταξύ των καναλιών και η συσχέτιση μεταξύ των καναλιών.

4.4.4 AAC (Advance Audio Coding)

Το πρότυπο ACC αναπτύχθηκε για πρώτη φορά το 1997 από το Ινστιτούτο Fraunhofer IIS και χρησιμοποιεί όπως και το MP3 το ψυχοακουστικό μοντέλο επικάλυψης, με σκοπό να καλύψει τα προβλήματα που υπήρχαν σχετικά με την ποιότητα των αρχείων MP3 στους μικρούς αριθμούς μετάδοσης των δεδομένων. Το πρότυπο ACC, ως ένα πραγματικό αριστούργημα κωδικοποίησης έχει την ικανότητα να αποδίδει εξαιρετικά υψηλή ποιότητα ήχου σε bitrate 64Kb/sec, επιτρέποντας τόσο την κωδικοποίηση 48 καναλιών ήχου και 16 καναλιών χαμηλής συχνότητας για εφέ όσο και την υποστήριξη πολλών γλωσσών ταυτόχρονα. Παράλληλα, το ACC διακατέχεται από τρεις διαφορετικές όψεις, την «κύρια», την «χαμηλής πολυπλοκότητα» και την «κλιμακούμενη συχνότητα δειγματοληψίας». Η «κύρια» όψη αναφέρεται σε εφαρμογές που η υπολογιστική ισχύει και εφαρμογές δεν είναι περιορισμένη, η «χαμηλής πολυπλοκότητα» σε

εφαρμογές που η ισχύος και η μνήμη βρίσκονται σε μεγάλη ζήτηση, ενώ η τελευταία, είναι φτιαγμένη έτσι ώστε οι αποκωδικοποιητές να έχουν ελάχιστες απαιτήσεις σε μνήμη και ισχύ. Υποκειμενικά τεστ που έχουν πραγματοποιηθεί με καλά εκπαιδευμένους ακροατές, έδειξαν ότι η συγκεκριμένη κωδικοποίηση προσφέρει καλύτερη ποιότητα ήχου από οποιαδήποτε άλλη κωδικοποίηση ήχου με το μισό μόνο bitrate. Στο σημείο αυτό, οφείλουμε να αναφέρουμε ότι το πρότυπο ACC παρέχει καλύτερη απόδοση από τοMP3, ενώ το 2003 η έκδοση του παρουσιάζεται συμβατή με τις προδιαγραφές του Mpeg ούτως ώστε το πρότυπο να αναφέρεται και ως Mpeg-4 ACC.

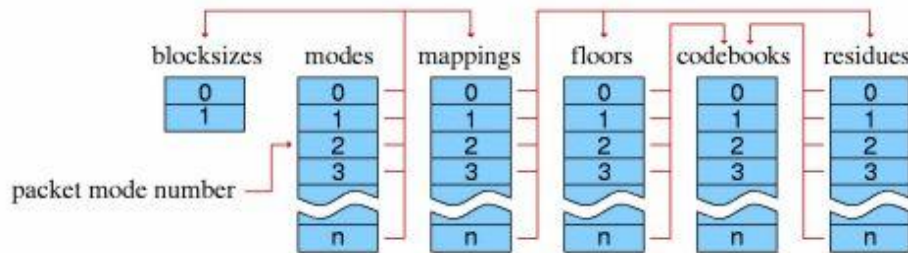
4.4.5 RA-RealAudio

Το RealAudio ως κλειστό πρότυπο, δημιουργήθηκε και υποστηρίχθηκε από την εταιρία Real Network με σκοπό την αναπαραγωγή ήχων στο Διαδίκτυο χωρίς να προηγείται κατέβασμα των ήχων στο σκληρό δίσκο του υπολογιστή. Το πρότυπο RealAudio, είναι αρκετά δημοφιλές και αυτό εξαιτίας της ελεύθερης διάθεσης του λογισμικού ανάγνωσης των αρχείων ήχου τέτοιου τύπου υποστηρίζοντας, μεγάλη συμπίεση και κατά κανόνα χαμηλή ποιότητα ήχου.

4.4.6 OGGVorbis

Το codec Ogg Vorbis αναπτύχθηκε γύρω από το πρότυπο αρχείων Ogg και βασίζεται στην opensource εφαρμογή απωλεστικής συμπίεσης με την ονομασία Vorbis. Ως προς τον τρόπο κωδικοποίησης, ο τρόπος μοιάζει με αυτό του MP3 ενώ ταυτόχρονα ο Ogg Vorbis χρησιμοποιεί MDCT για τον μετασχηματισμό του σήματος από το πεδίο του χρόνου στο πεδίο της συχνότητας, καθώς και μία εναλλακτική μέθοδο επεξεργασίας του φάσματος,

κατά την οποία κωδικοποιείται το φάσμα βάσης του οποίου η κατανομή είναι σχετικώς ομαλή και με περισσότερα ψηφία το απομένον φάσμα που η δομή και η χρονική εξέλιξη είναι πολύ πιο πολύπλοκη. Συμπερασματικά, η τακτική αυτή σε συνδυασμό με την καλή ποιότητα ήχου που προσφέρει, ωθεί το πρότυπο Ogg σε ένα ανταγωνιστικό παιχνίδι ως προς το WMA και MP3.



Εικόνα 4-8 Βαθμίδες του κωδικοποιητή OggVorbis. Η κωδικοποίηση του φάσματος βάσης και του απομένοντος φάσματος

4.4.7 AC3 DolbyDigital

Ένα από τα πιο διαδεδομένα πρότυπα για τον ψηφιακό πολυκάναλο ήχο είναι το AC3, που εκτός σημαντικού απροόπτου άρχισε να γίνεται το διεθνές πρότυπο για την συμπίεση ηχητικών δεδομένων. Στο ψηφιακό σύστημα ήχου AC3, ο ήχος κωδικοποιείται σε έξι συνολικά κανάλια στηριζόμενο στην μέθοδο 5.1. Συγκεκριμένα, υπάρχουν: (α) τρία κανάλια (αριστερό, κεντρικό, δεξί) που αποσκοπούν στο να φέρουν την βασική ηχητική πληροφορία, (β) δύο συνοδευτικά κανάλια περιβάλλοντος ήχου και (γ) ένα κανάλι για τις υπόλοιπες συχνότητες (σύστημα 3/2/1). Ως προς τον τρόπο λειτουργίας τους, τα πέντε πρώτα κανάλια διαχειρίζονται συχνότητες ήχου στο διάστημα 3-20000Hz, ενώ το έκτο συχνότητες 3-120Hz. Συμπερασματικά, ο ρυθμός δειγματοληψία είναι 48KHz μεγαλύτερος από το ρυθμό των 44KHz που χρησιμοποιείται στα CDs, ενώ η συμπίεση των ηχητικών δεδομένων ανέρχεται στην αναλογία 10:1.

5. Περιβάλλον προγραμματισμού Matlab

5.1 Εισαγωγή στην Matlab

Το πακέτο λογισμικού MATLAB (MathWorksInc.) παρέχει ένα δυναμικό, εύχρηστο και ανοικτό υπολογιστικό περιβάλλον για υλοποίηση επιστημονικών εφαρμογών σε ένα μεγάλο φάσμα γνωστικών πεδίων, όπως στη Γραμμική Άλγεβρα, τη Στατιστική, τα Εφαρμοσμένα Μαθηματικά, την Αριθμητική Ανάλυση, την Επεξεργασία Σημάτων και Εικόνας, τη Θεωρία Ελέγχου. Έχει υλοποιηθεί σε πολλές λειτουργικές πλατφόρμες (όπως Windows, MacintoshOS και Unix) και σε δύο βασικές εκδόσεις, την επαγγελματική και την εκπαιδευτική (studentedition).

Το περιβάλλον του MATLAB υποστηρίζει ένα μεγάλο αριθμό ενδογενών λειτουργιών και συναρτήσεων, καθώς και εξωτερικές βιβλιοθήκες (Toolboxes) για εξειδικευμένες περιοχές εφαρμογών. Υποστηρίζει επίσης μία ευέλικτη, απλή και δομημένη γλώσσα προγραμματισμού (scriptlanguage), η οποία έχει πολλές ομοιότητες με τη γλώσσα C, και παρέχει δυνατότητες εύκολης δημιουργίας, διασύνδεσης και χρήσης βιβλιοθηκών σε κώδικα γραμμένο στη γλώσσα αυτή (εφεξής θα ονομάζεται γλώσσα MATLAB).

Το MATLAB εκτελεί από απλούς μαθηματικούς υπολογισμούς έως και προγράμματα με εντολές παρόμοιες εκείνων που υποστηρίζει μία γλώσσα προγραμματισμού υψηλού επιπέδου (**Βαρσάμης, 2015**). Συγκεκριμένα, εκτελεί απλές μαθηματικές πράξεις αλλά εξίσου εύκολα χειρίζεται μιγαδικούς αριθμούς, δυνάμεις, ειδικές μαθηματικές συναρτήσεις, πίνακες, διανύσματα και πολυώνυμα. Μπορεί επίσης να αποθηκεύει και να ανακαλεί δεδομένα, να δημιουργεί και να εκτελεί ακολουθίες εντολών που αυτοματοποιούν διάφορους υπολογισμούς και να σχεδιάζει γραφικά.

Οι λειτουργίες του MATLAB διακρίνονται στις τυποποιημένες, δηλαδή σε αυτές που χειρίζονται αριθμητικά δεδομένα και εξάγουν αριθμητικά αποτελέσματα, και στις συναρτήσεις του Symbolic Toolbox, οι οποίες

χειρίζονται και υπολογίζουν συμβολικές εκφράσεις, δηλαδή επεξεργάζονται μαθηματικά σύμβολα.

Η γλώσσα προγραμματισμού του MATLAB δίνει την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα. Συχνά θα γράφουμε η MATLAB (εννοώντας τη γλώσσα προγραμματισμού) και όχι το (πακέτο) MATLAB. Το MATLAB είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε αριθμητική πεπερασμένης ακρίβειας (finite-precision arithmetic), δηλαδή δεν βρίσκει την ακριβή αλλά μια προσεγγιστική λύση ενός προβλήματος. Αυτή είναι και η βασική του διαφορά από τα συστήματα συμβολικών υπολογισμών όπως η Maple και το Mathematica.

5.2 PCM κωδικοποίηση - Παλμοκωδική Διαμόρφωση

Για να εκμεταλλευτούμε τα πλεονεκτήματα της ψηφιακής μετάδοσης σημάτων πρέπει να μετατρέψουμε το αναλογικό σήμα σε ψηφιακό. Για να επιτευχθεί αυτός ο σκοπός χρειαζόμαστε έναν δειγματολήπτη, που θα παίρνει τιμές (δείγματα) από το αναλογικό σήμα με τέτοιο τρόπο ώστε να μπορούμε αργότερα να το επεξεργαστούμε χωρίς μεγάλη απώλεια πληροφορίας σε σχέση με την αρχική. Στην συνέχεια περνάμε το σήμα μέσα από έναν κβαντιστή όπου αντιστοιχούμε τιμές που βρίσκονται μέσα σε ορισμένα διαστήματα σε μια συγκεκριμένη τιμή ώστε να συμπιέσουμε το σήμα μας (με απώλεια πληροφορίας). Τέλος κωδικοποιούμε το σήμα μας μέσω ενός κωδικοποιητή ώστε στην συνέχεια να μπορέσουμε να το μεταδώσουμε αποδοτικά. Στην πηγή αποκωδικοποιείται το σήμα ώστε να μπορέσουμε να διαβάσουμε την μεταδιδόμενη πληροφορία. Ένα επιπρόσθετο βήμα που υπάρχει σε αρκετές περιπτώσεις είναι η χρήση συμπιεστή πριν την κβάντιση ώστε να έχουμε καλύτερα αποτελέσματα ως προς την ποιότητα του σήματος.

Η παλμοκωδική διαμόρφωση (PCM) είναι το παλιότερο και απλούστερο σχήμα ψηφιακής διαμόρφωσης αναλογικών δεδομένων. Αποτελείται από τρία βασικά μέρη :

- δειγματολήπτη
- κβαντιστή
- κωδικοποιητή

όπως φαίνονται και στο παρακάτω σχήμα:



Εικόνα 5-1 Διάγραμμα βαθμίδων ενός συστήματος PCM



Πριν την εισαγωγή στο δειγματολήπτη ενός σήματος, αυτό έχει περάσει (συνήθως) από ένα προδειγματοληπτικό φίλτρο που εμποδίζει την είσοδο συνιστωσών του σήματος πέρα από το εύρος ζώνης W που μας ενδιαφέρει.

Αφού το σήμα περάσει από τον δειγματολήπτη (όπου συνήθως δειγματοληπτείται με συχνότητα μεγαλύτερη από αυτή του Nyquist), εισέρχεται σε έναν βαθμωτό κβαντιστή. Ο κβαντιστής αυτός μπορεί να είναι είτε ομοιόμορφος είτε μη ομοιόμορφος ανάλογα με τα στατιστικά χαρακτηριστικά της εξόδου της πηγής. Μετά την κβάντιση του το σήμα

κωδικοποιείται από τον κωδικοποιητή με μια δυαδική ακολουθία μήκους ν όπου $N = 2^\nu$ είναι ο αριθμός των σταθμών κβάντισης.

5.2.1 Ομοιόμορφος κβαντιστής

Ο ομοιόμορφος κβαντιστής περιορίζει την δυναμική περιοχή του σήματος εισόδου στις τιμές $[-\max_value, \max_value]$ θέτοντας όποιες τιμές είναι εκτός αυτού του εύρους στις αντίστοιχες ακραίες τιμές. (Τσίνογ, 2014)

Η διαδικασία η οποία ακολουθείται είναι αφού ο κβαντιστής χωρίσει την περιοχή σε N τμήματα, αντιστοιχεί τις τιμές που πέφτουν σε κάθε ένα από αυτά τα τμήματα στην τιμή του μέσου της αντίστοιχης περιοχής κβάντισης. Οι περιοχές στις οποίες χωρίζεται το αρχικό εύρος τιμών του σήματος είναι ισομήκης Σύμφωνα με τα παραπάνω, κάθε περιοχή έχει εύρος $\Delta = 2x_{\max}/N = x_{\max}/2^{(\nu-1)}$. Συνεπώς και τα κέντρα κβάντισης απέχουν μεταξύ τους την ίδια απόσταση Δ .

Ως μέση τετραγωνική παραμόρφωση (distortion) καλούμε το μέτρο $D = E[(X-Q(X))^2]$ με X τυχαία μεταβλητή που αντιστοιχεί στο σήμα εισόδου και $Q(X)$ το κβαντισμένο σήμα. Ένα από τα πιο σημαντικά μέτρα της επίδοσης της κβάντισης είναι μια κανονικοποιημένη έκδοση του θορύβου κβάντισης ως προς την ισχύ του αρχικού σήματος

(1)

= (2)

Παρακάτω παραθέτουμε τον κώδικα στην `matlab` που προσομοιώνει την λειτουργία του ομοιόμορφου κβαντιστή:

```
function [xq, centers, p, D] = my_quantizer(x, N, max_value)
```

```

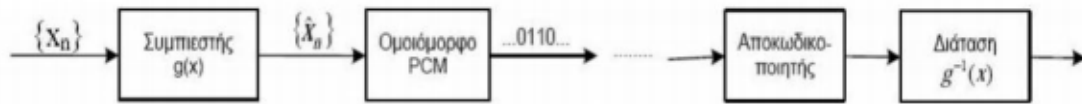
% Input arguments
% x: input signal, already in discrete form
% N: number of bits that will be used
% max_value: maximum acceptable value of the signal
% Output arguments
% xq: the vector of output signal
% centers: centers of quantized region
% p: vector that includes the appearance probability of each
center of the
% quantizer
% D: distortion of the quantized signal
% the number of quantized regions
num_of_regions = 2^N;
% how many times we meet a value in a specific interval
occurrences = zeros(num_of_regions, 1);
% length of each region
% each region has the same length
length_of_region = 2 * max_value / num_of_regions;
% the limits of each region without the "infinities"
vector_of_intervals = zeros(num_of_regions - 1, 1);
num_of_intervals = length(vector_of_intervals);
for j=1:1:num_of_intervals
vector_of_intervals(j, 1) = - max_value + j *
length_of_region;
end
% the number of input samples
num_of_samples = length(x);
centers = zeros(num_of_regions, 1);
% calculate the center of each region
for j = 1: 1: num_of_intervals
centers(j, 1) = vector_of_intervals(j, 1) - length_of_region /
2;
end

```

```
centers(num_of_regions, 1) =  
vector_of_intervals(num_of_intervals, 1) + length_of_region /  
2;  
for j = 1: 1: num_of_samples  
pos = binarysearch(x(j, 1), vector_of_intervals);  
xq(j, 1) = centers(pos, 1);  
occurrences(pos, 1) = occurrences(pos, 1) + 1;  
end  
for j = 1: 1: num_of_regions  
p(j, 1) = occurrences(j, 1) / num_of_samples;  
end  
D = distortion(x, xq);  
return;
```

5.2.2 Μη ομοιόμορφο PCM

Κατά την κωδικοποίηση ορισμένων σημάτων όπως η ομιλία, η κατανομή της εισόδου απέχει πολύ από την ομοιόμορφη, σε αυτές τις περιπτώσεις προτιμούμε το μη ομοιόμορφο PCM. Η συνηθέστερη μέθοδος για την υλοποίηση της μη ομοιόμορφης κβάντισης είναι τα δείγματα να διέλθουν πρώτα από ένα μη γραμμικό στοιχείο προκειμένου να συμπιεστούν τα μεγάλα πλάτη (μείωση δυναμική περιοχή του σήματος) και στην συνέχεια η έξοδος του μη γραμμικού σημείου να κβαντιστεί ομοιόμορφα. Στη λήψη εφαρμόζεται η αντίστροφη λειτουργία της συμπίεσης (διάταση) για να ανακτήσουμε τις τιμές των δειγμάτων. Η τεχνική αυτή ονομάζεται *companding* και περιγράφεται από το σχήμα 5-3.

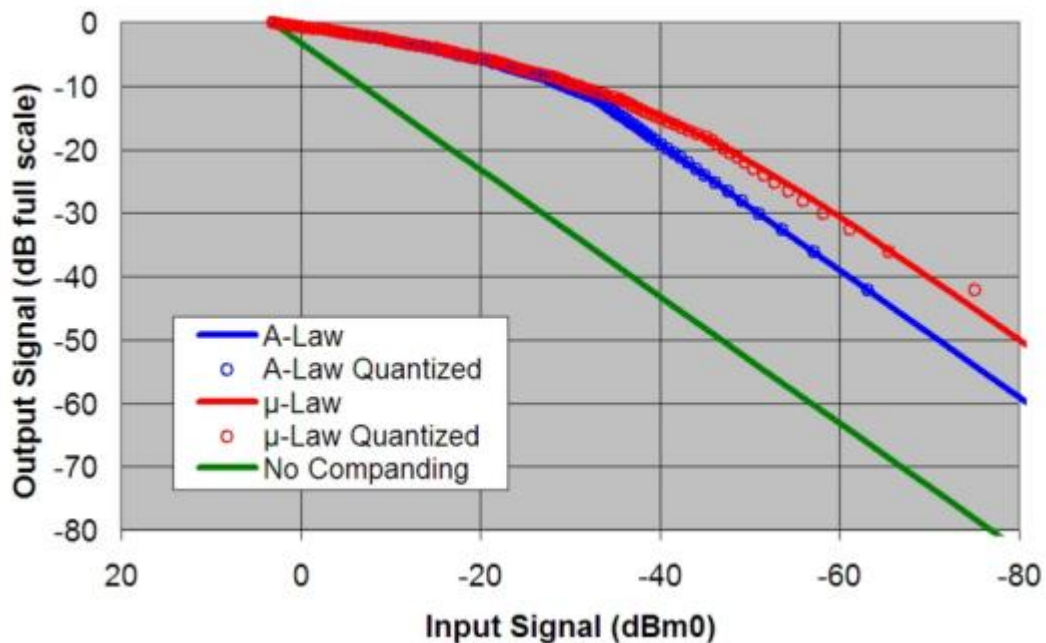


Εικόνα 5-2 Διάγραμμα βαθμίδων ενός συστήματος μη ομοιόμορφου PCM

Οι αλγόριθμοι συμπίεσης έχουν την ιδιότητα να μειώνουν δυναμικό εύρος ενός σήματος ήχου. Στα ψηφιακά συστήματα μπορούν να αυξήσουν το SQNR μειώνοντας το σφάλμα κβάντισης.

Για την κωδικοποίηση ομιλίας δύο τύποι συμπίεστων χρησιμοποιούνται ευρέως. Ο συμπίεστος τύπου-μ, ο οποίος χρησιμοποιείται στην Βόρεια Αμερική και Ιαπωνία, και ο συμπίεστος τύπου-A που χρησιμοποιείται κυρίως στην Ευρώπη. Ο συμπίεστος τύπου-μ παρέχει ένα ελαφρώς δυναμικό εύρος σε σχέση με τον συμπίεστη τύπου-A με το κόστος χειρότερης αναλογικής παραμόρφωσης για μικρά σήματα. Από σύμβαση, ο συμπίεστος τύπου-A χρησιμοποιείται για διεθνής επικοινωνίες όταν τουλάχιστον μια χώρα τον χρησιμοποιεί.

Στο γράφημα 5-4 συγκρίνονται ο μ-συμπίεστος με τον A-συμπίεστη. Οι τύποι που περιγράφουν την διαδικασία συμπίεσης και διάτασης των αντίστοιχων αλγορίθμων παρουσιάζονται παρακάτω :



Εικόνα 5-3Γράφημα του μ-συμπίεστη και A-συμπίεστη

Υλοποίηση συμπιεστών και αποσυμπιεστών

Συμπιεστής Τύπου-μ

```

function [xc] = compressor_m(x, m)
% Input arguments
% x: input signal, already in discrete form
% m: coefficient of compression
% Output arguments
% xc: the x signal in compressed form
% the number of input samples
num_of_samples = length(x);
% error message
Error = 'We could not compress this signal because it is
expanded over a large area';
for i = 1: 1: num_of_samples
% if the signal x is expanded over the limits -1<=x<=1, can
not be compressed
if (x(i, 1) < -1) || (x(i, 1) > 1)
Error
return;
end
% compress each value of the discrete signal
xc(i, 1) = sign( x(i, 1) ) * reallog(1 + m * abs( x(i, 1) ) ) /
reallog( 1 + m ) ;
end
return;

```

Αποσυμπιεστής Τύπου-μ

```

function [xd] = decompressor_m(x, m)
% Input arguments

```

```
% x: input signal, already in discrete form
% m: coefficient of decompression
% Output arguments
% xd: the x signal in uncompressed form
% the number of input samples
num_of_samples = length(x);
% error message
Error = 'We could not decompress this signal because it is
expanded over a large area';
for i = 1: 1: num_of_samples
% if the signal x is expanded over the limits -1<=x<=1, can
not be decompressed
if (x(i, 1) < -1) || (x(i, 1) > 1)
Error
return;
end
% decompress each value of the discrete signal
xd(i, 1) = sign( x(i, 1) ) * ( (1 + m) ^ abs( x(i, 1) ) - 1 ) / m
;
end
return;
```

Συμπιεστής Τύπου-A

```
function [xc] = compressor_a(x, a)
% Input arguments
% x: input signal, already in discrete form
% a: coefficient of compression
% Output arguments
% xc: the x signal in compressed form
```

```
% the number of input samples
num_of_samples = length(x);
% error message
Error = 'We could not compress this signal because it is
expanded over a large area';
for i = 1: 1: num_of_samples
% if the signal x is expanded over the limits 0<=|x|<=1,
cannot be compressed
if (x(i, 1) < -1) || (x(i, 1) > 1)
Error
return;
% compress each value of the discrete signal
elseif ( abs( x(i, 1) ) <= 1/a )
xc(i, 1) = sign( x(i, 1) ) * a * abs( x(i, 1) ) / ( 1 +
reallog(a) );
else
xc(i, 1) = sign( x(i, 1) ) * ( 1 + reallog( a * abs( x(i, 1) ) )
) / ( 1 + reallog(a));
end
end
return;
```

Αποσυμπιεστής Τύπου-A

```
function [xd] = decompressor_a(x, a)
% Input arguments
% x: input signal, already in discrete form
% a: coefficient of decompression
```

```
% Output arguments

% xd: the x signal in decompressed form

% the number of input samples
num_of_samples = length(x);

% error message
Error = 'We could not decompress this signal because it is
expanded over a large area';

for i = 1: 1: num_of_samples

% if the signal x is expanded over the limits 0<=|x|<=1, can
not be decompressed
if (x(i, 1) < -1) || (x(i, 1) > 1)

Error

return;

% decompress each value of the discrete signal
elseif ( abs( x(i, 1) ) <= 1/( 1 + reallog(a) ) )
xd(i, 1) = sign( x(i, 1) ) * ( 1 + reallog( a) ) * abs( x(i, 1) )
/ a;
else
xd(i, 1) = sign( x(i, 1) ) * exp( abs( x(i, 1) ) * ( 1 +
reallog(a) ) - 1 ) / a;
end
end

return;
```

5.2.2 Μη ομοιόμορφος κβαντιστής

Χαλαρώνοντας την συνθήκη από τον ομοιόμορφο κβαντιστή όπου όλες οι περιοχές κβάντισης (εκτός πρώτης και τελευταίας) πρέπει να έχουν το ίδιο

εύρος, μπορούμε να δημιουργήσουμε έναν κβαντιστή που θα λειτουργεί με καλύτερες επιδόσεις σε σύγκριση με έναν ομοιόμορφο ίδιων σταθμών. Αυτό συμβαίνει γιατί μπορούμε να κάνουμε την ελαχιστοποίηση της παραμόρφωσης με λιγότερους περιορισμούς, αν και ο κβαντιστής που θα προκύψει θα είναι πιο πολύπλοκος σε σχέση με έναν ομοιόμορφο.

Για να έχουμε έναν βέλτιστο κβαντιστή, τα άκρα των περιοχών κβάντισης θα πρέπει να δίνονται από τον αριθμητικό μέσο των γειτονικών τιμών κβάντισης. Έτσι η κβάντιση γίνεται με βάση την ελάχιστη απόσταση, δηλαδή κάθε τιμή x κβαντίζεται στο πλησιέστερο $\{x_i\}$

Σύμφωνα με την παραπάνω παρατήρηση, για να έχουμε έναν βέλτιστο βαθμωτό κβαντιστή, πρέπει να πληρούνται οι συνθήκες Lloyd-Max. Τα κριτήρια για την κβάντιση συνοψίζονται :

1. Τα άκρα των περιοχών κβάντισης δίνονται από τον αριθμητικό μέσο των γειτονικών τιμών κβάντισης (νόμος πλησιέστερου γείτονα)
2. Οι τιμές κβάντισης είναι τα κέντρα μάζας των περιοχών κβάντισης

Δυστυχώς, παρόλου που αυτοί οι κανόνες είναι πολύ απλοί, δεν δίνουν αναλυτικές λύσεις για την σχεδίαση του βέλτιστου κβαντιστή. Για αυτό τον λόγο η πιο συνηθισμένη μέθοδος σχεδιασμού είναι να ξεκινήσουμε με ένα σύνολο περιοχών κβάντισης και συνεχίζουμε χρησιμοποιώντας το δεύτερο κριτήριο. Ακολουθώντας επανασχεδιάζουμε τις περιοχές κβάντισης και επαναλαμβάνουμε τα δυο παραπάνω βήματα μέχρι η παραμόρφωση από βήμα σε βήμα να έχει μειωθεί κάτω από μια επιθυμητή τιμή.

Ο αλγόριθμος συνοψίζεται στα εξής βήματα. Σε κάθε επανάληψη του Lloyd-Max:

1. Υπολογίζουμε τα όρια των ζωνών κβάντισης, που πρέπει να είναι στο μέσο των επιπέδων κβαντισμού, δηλαδή :

2. Υπολογίζουμε το κβαντισμένο σήμα με βάση τις περιοχές αυτές και μετράμε την μέση παραμόρφωση D με βάση το δοθέν σήμα.

$$\hat{x}_k^{(i+1)} = E[x | T_{k-1} < x < T_k]$$

3. Τα νέα επίπεδα κβαντισμού είναι τα κεντροειδή των ζωνών :

$$|D_i - D_{i-1}| < \epsilon$$

Παρακάτω παραθέτουμε τον κώδικα στην Matlab που προσομοιώνει την λειτουργία του μη ομοιόμορφου κβαντιστή χρησιμοποιώντας τα κριτήρια Lloyd-Max και σύμφωνα με τον παραπάνω αλγόριθμο.

```
function [xq, centers, D] = Lloyd_Max(x, N, max_value)
% Input arguments
% x: input signal, already in discrete form
% N: number of bits that will be used
% max_value: maximum acceptable value of the signal
% Output arguments
% xq: the vector of output signal after K_{max} loops
% centers: centers of quantized region
% D: vector of distortions of the quantized signal of each
iteration
% fault tolerance
epsilon = eps;
% initial value of distortion
D(1, 1) = 1;
% the number of quantized regions
num_of_regions = 2^N;
% length of each region
% each region has the same length
length_of_region = 2 * max_value / num_of_regions;
% the limits of each region without the "infinities"
vector_of_intervals = zeros(num_of_regions - 1, 1);
num_of_intervals = length(vector_of_intervals);
```

```
% initialization of the intervals according to a uniform
distribution
for j=1: 1: num_of_intervals
vector_of_intervals(j, 1) = - max_value + j *
length_of_region;
end
% the vector that has the expected values of each region
expected_value = zeros(num_of_regions, 1);
% Initialization of the expected values
for j=1: 1: num_of_intervals
expected_value(j, 1) = vector_of_intervals(j, 1) -
length_of_region / 2;
end
expected_value(num_of_regions, 1) =
vector_of_intervals(num_of_intervals, 1) + length_of_region /
2;
% the number of input samples
num_of_samples = length(x);
for i = 2: 1: 10^5
% how many times we meet a value in a specific interval
occurrences = zeros(num_of_regions, 1);
% the vector that has the mass value the interval
mass_value = zeros(num_of_regions, 1);
% Find in which interval each sample belongs
% Count how many occurrences there are in a specific interval
and calculate
% their sum in order to find the expected value of this
interval
for j = 1: 1: num_of_samples
pos = binarysearch(x(j, 1), vector_of_intervals);
occurrences(pos, 1) = occurrences(pos, 1) + 1;
mass_value(pos, 1) = mass_value(pos, 1) + x(j, 1);
xq(j, 1) = expected_value(pos, 1);
```



```
end
% Calculate the expected value for the next iteration
for j = 1: 1: num_of_regions
if mass_value(j, 1) == 0
if j ~= 1
expected_value(j, 1) = vector_of_intervals(j-1, 1);
else
expected_value(j, 1) = vector_of_intervals(j, 1);
end
else
expected_value(j, 1) = mass_value(j, 1) / occurrences(j, 1);
end
end
% Redefine the limits of each region for the next iteration
for j = 1: 1: num_of_intervals
vector_of_intervals(j, 1) = ( expected_value(j, 1) +
expected_value(j + 1, 1) ) / 2;
end
% calculate the distortion
D(i, 1) = distortion(x, xq);
if abs(D(i, 1) - D(i - 1, 1)) < epsilon
centers = expected_value;
return;
end
end
return;
```

5.3 Πειραματική Αξιολόγηση

Κωδικοποιήσαμε τα δείγματα της πηγής για $N = 2, 4, 6$ bits σύμφωνα με τις Παραπάνω μεθόδους και αξιολογήσαμε τα αποτελέσματα βασιζόμενοι :

- Στις τιμές του SQNR

- Στο ακουστικό αποτέλεσμα κάθε μεθόδου χρησιμοποιώντας την συνάρτηση `sound()` της Matlab
- Στις κυματομορφές εξόδου.

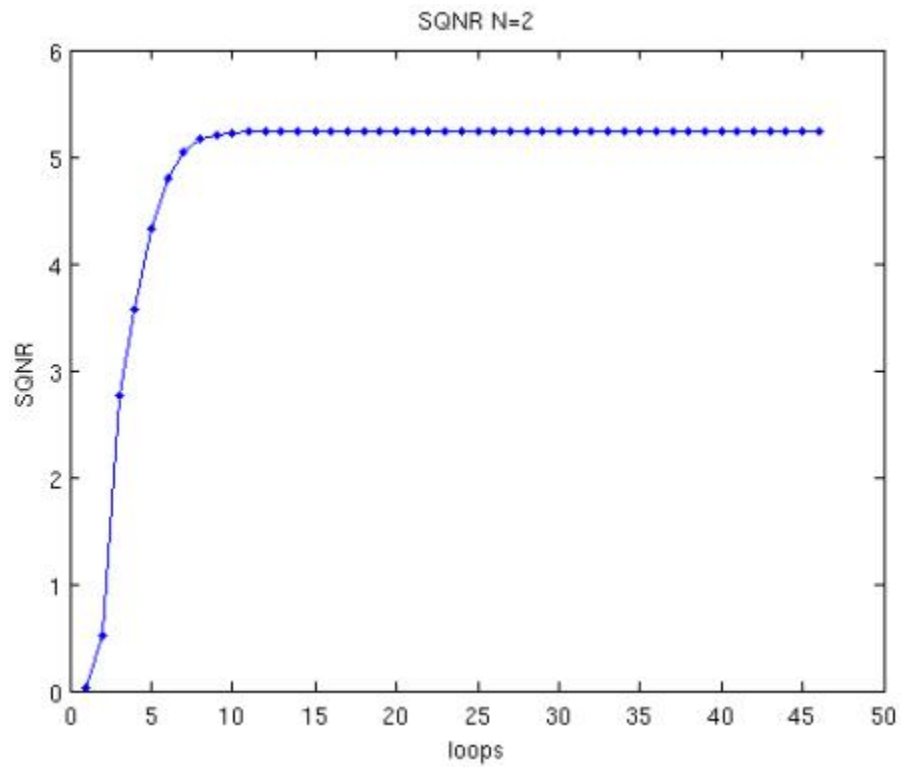
5.3.1 Με βάση το SQNR

Με βάση τον τύπο υπολογισμού του SQNR όπως αναφέρθηκε υπολογίσαμε τις παρακάτω τιμές που δείχνουν μια μετρική της ποιότητας του σήματος μετά την κβάντιση του. Τα αποτελέσματα για τις τρεις πρώτες μεθόδους φαίνονται στον Πίνακα 1.

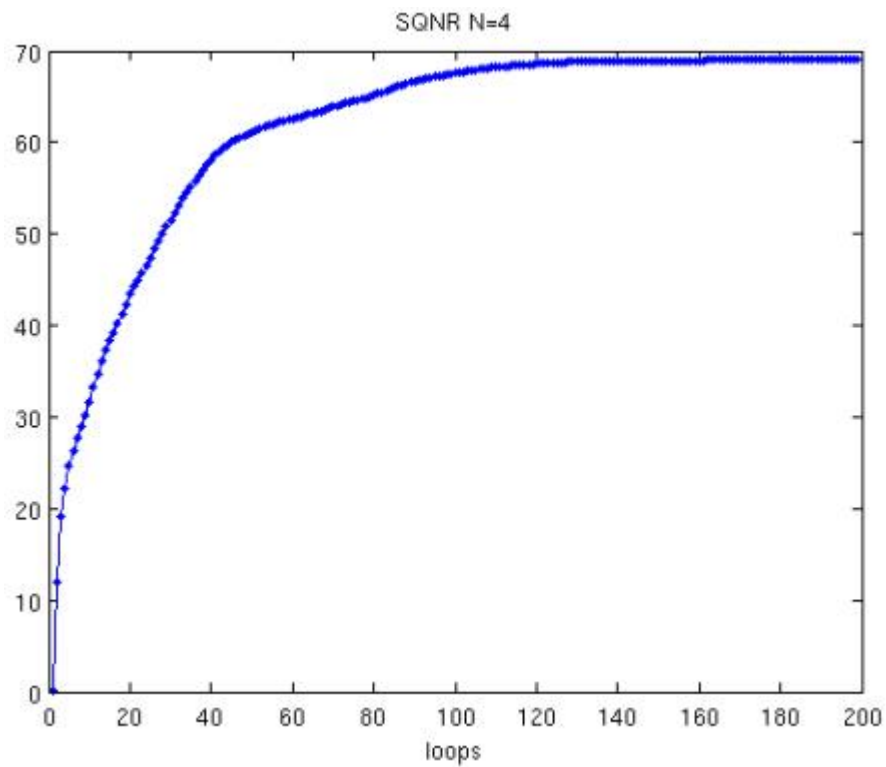
Πίνακας 5 - Τιμές SQNR

	N=2	N=4	N=6
Ομοιόμορφο PCM	0,513	11,9	234,69
Συμπίεσής τύπου-μ	2,33	22,18	389,84
Συμπίεσής τύπου-A	2,41	23,59	414,94

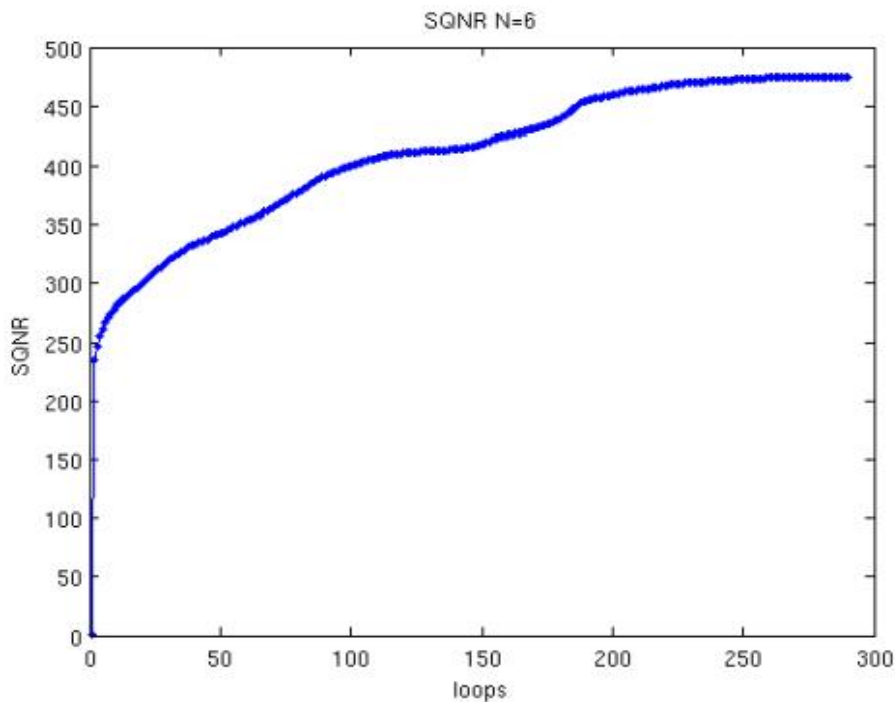
Ενώ για την τέταρτη μέθοδο τα αποτελέσματα απεικονίζονται στα διαγράμματα 5, 6, 7 και συγκεκριμένα σε αυτά παρουσιάζεται η μεταβολή του SQNR σε σχέση με τον αριθμό των επαναλήψεων του αλγορίθμου Lloyd-Max.



Εικόνα 5-5 Μεταβολή SQNR για κβάντιση 2-bit



Εικόνα 5-4 Μεταβολή SQNR για κβάντιση 4-bit



Εικόνα 5-7 Μεταβολή SQNRγια κβάντιση 6-bit

Παρατηρούμε πως όσο αυξάνονται τα bit που χρησιμοποιούμε για την κβάντιση, τόσο καλύτερα αποτελέσματα παίρνουμε. Αναμενόμενο καθώς όπως γνωρίζουμε ισχύει ότι :

$$H(X) \leq \bar{R} \leq H(X) + \frac{1}{n}$$

Από τα παραπάνω αποτελέσματα φαίνεται πως ο ομοιόμορφος κβαντιστής εμφανίζει τη χαμηλότερη επίδοση όσον αφορά το εισαγόμενο σφάλμα μετά τη κβάντιση. Αυτό συμβαίνει καθώς κβαντίζει το σήμα μας σε προκαθορισμένες στάθμες ίδιου εύρους, ανεξάρτητα από το σήμα. Αντίθετα, η μη-ομοιόμορφη κβάντιση χρησιμοποιώντας τα κριτήρια Lloyd-Max δίνει τα καλύτερα αποτελέσματα μιας και σκοπό της έχει να κβαντίσει το σήμα με την όσο το δυνατόν λιγότερη παραμόρφωση, υπό το κόστος όμως περισσότερων υπολογισμών και αυξημένης πολυπλοκότητας υλοποίησης.

Όσον αφορά, τους συμπιεστές, παρατηρούμε πως είναι μια πολύ καλή πρακτική λύση καθώς αν και δεν εμφανίζουν την πολύ καλή απόδοση του μη ομοιόμορφου κβαντιστή, αλλά χρησιμοποιώντας μόνο ένα μη γραμμικό

στοιχείο μπορούμε να επιτύχουμε αρκετά ικανοποιητικά αποτελέσματα στην πράξη.

Τέλος, παρατηρούμε πως για την μέθοδο Lloyd-Max όσο αυξάνονται τα bits που χρησιμοποιούμε, τόσα πιο πολλά iterations χρειάζονται για τον τερματισμό του αλγορίθμου φθάνοντας στην τιμή στόχο, που καθορίζεται από την μικρή μεταβολή της παραμόρφωσης (distortion). Αυτό είναι και αναμενόμενο αν λάβουμε υπόψη μας πως για λίγα bits αναμένουμε μεγάλη παραμόρφωση και επομένως θα την φθάσουμε σχετικά γρήγορα και από εκεί και πέρα θα αλλάζει ελάχιστα.

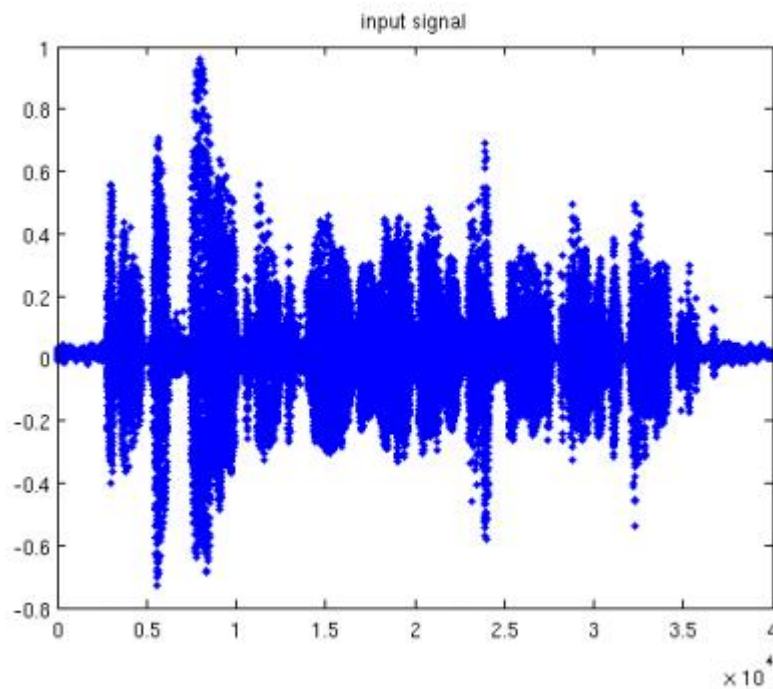
5.3.2 Με βάση το ακουστικό αποτέλεσμα

Για να κάνουμε αυτές τις παρατηρήσεις ακούσαμε το αρχικό σήμα και στην συνέχεια το κβαντισμένο μέσω της συνάρτησης `sound()`.

Αυξάνοντας τον αριθμό των bits κβάντισης παρατηρούσαμε μεγάλη διαφορά στην ποιότητα, και ειδικότερα από τα 2-bit στα 4-bit όπου ο περισσότερος θόρυβος έφευγε. Αντιληπτή ήταν επίσης και η διαφορά στην ποιότητα ανάμεσα σε ομοιόμορφο κβαντιστή και μη ομοιόμορφο σύμφωνα με Lloyd-Max, αλλά όχι στον ίδιο βαθμό. Για $N = 6$ το σήμα που ακούσαμε στην έξοδο ακουγόταν σχεδόν το ίδιο με αυτό της εισόδου.

Αυτό ήταν και το πιο εντυπωσιακό μέρος της εργασίας μαζί με τις κυματομορφές εξόδου καθώς μπορούσαμε να ακούσουμε το σήμα μέσα από το διάγραμμα με το οποίο περιγραφόταν.

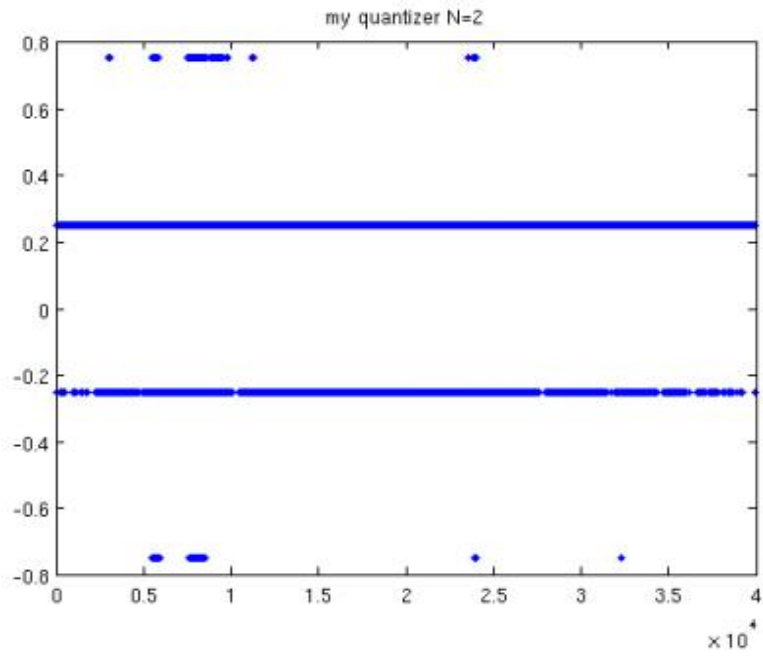
5.3.3 Με βάση τις κυματομορφές εξόδου



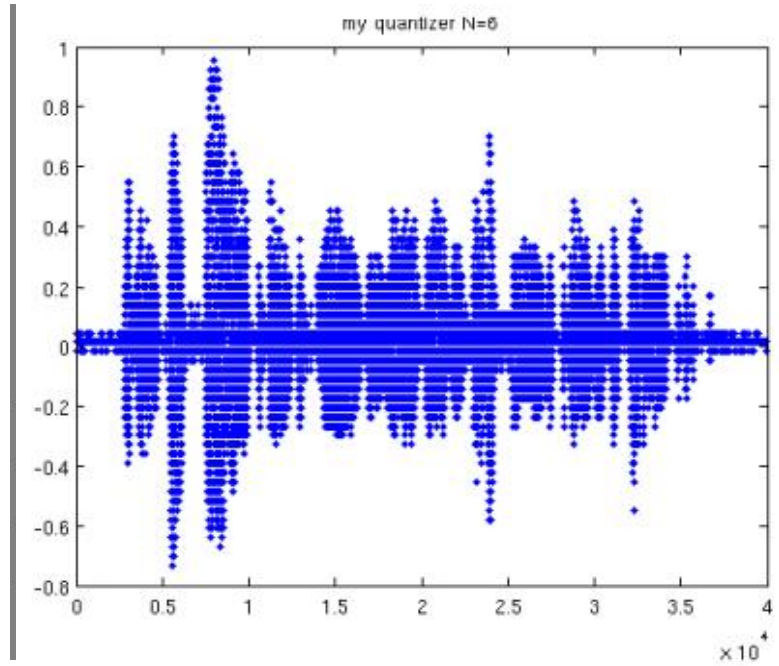
Εικόνα 5-6 Κυματομορφή αρχικού σήματος

Για να συγκρίνουμε το αποτέλεσμα μετά τη κβάντιση τού σήματος με το αρχικό πήραμε την κυματομορφή εισόδου του αρχικού σήματος (Σχήμα 8 και την συγκρίναμε με τη κυματομορφή του κβαντισμένου σήματος. Όπως μπορεί να διαπιστώσει κάποιος από τα παρακάτω σχήματα μπορούν να εξαχθούν παρόμοια συμπεράσματα όπως και παραπάνω, συγκρίνοντας το κατά πόσο το κάθε κβαντισμένο σήμα προσεγγίζει το αρχικό μας.

5.3.3.1 Ομοιόμορφη Κβάντιση

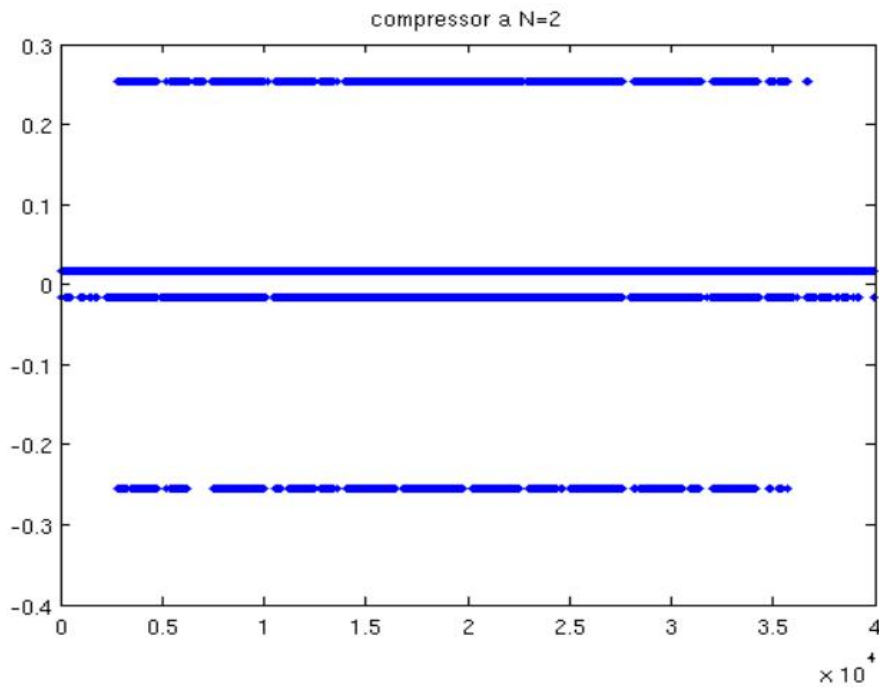


Εικόνα 5-7 Κυματομορφή εξόδου για ομοιόμορφη κβάντιση 2-bit

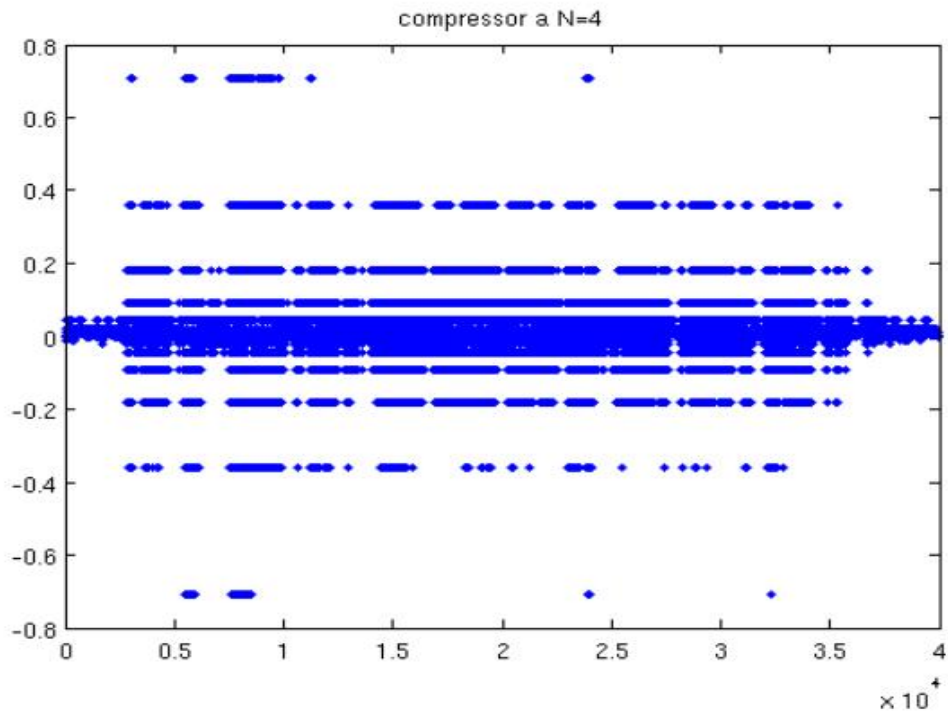


Εικόνα 5-8 Κυματομορφή εξόδου για ομοιόμορφη κβάντιση 6-bit

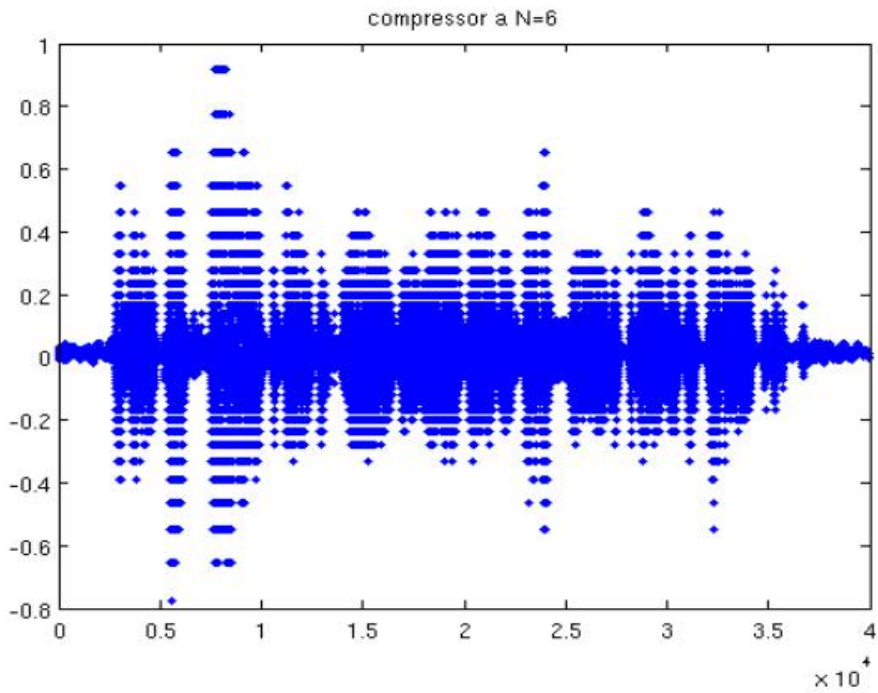
3.5.3.2 Μη ομοιόμορφο PCM με συμπιεστή τύπου-μ



Εικόνα 5-9Κυματομορφή εξόδου για μη ομοιόμορφο PCM 2-bitτύπου-A

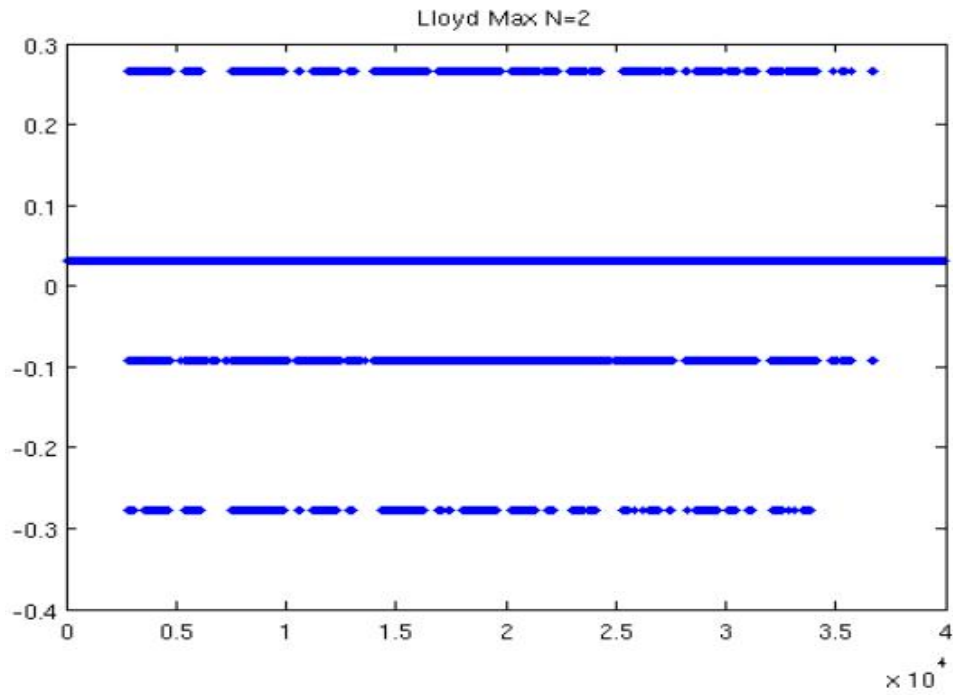


Εικόνα 5-12 Κυματομορφή εξόδου για μη ομοιόμορφο PCM 4-bit τύπου-A

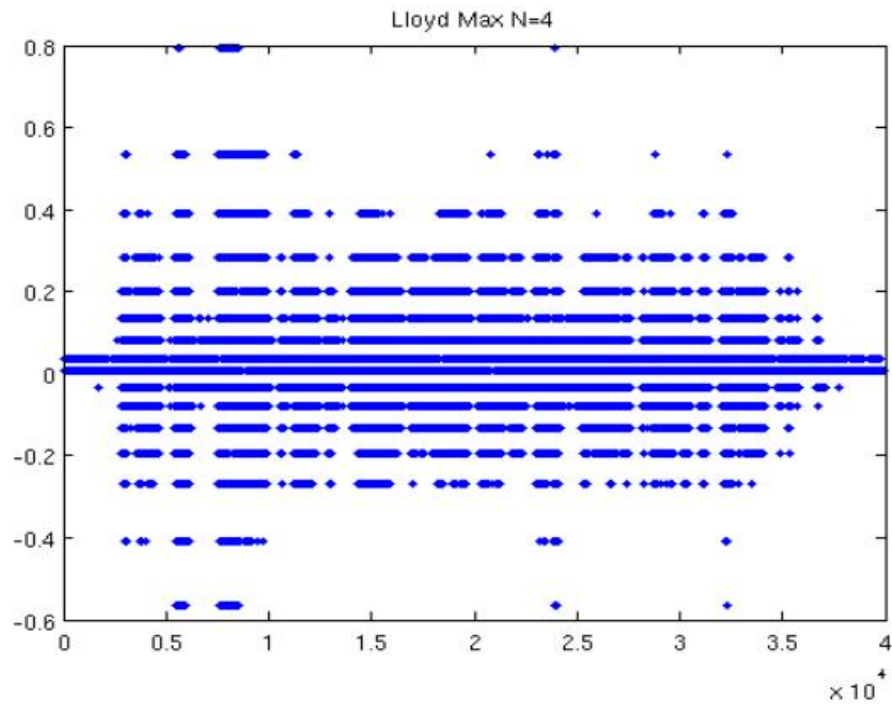


Εικόνα 5-13 Έξοδος για μη ομοιόμορφο PCM 6-bit τύπου-A

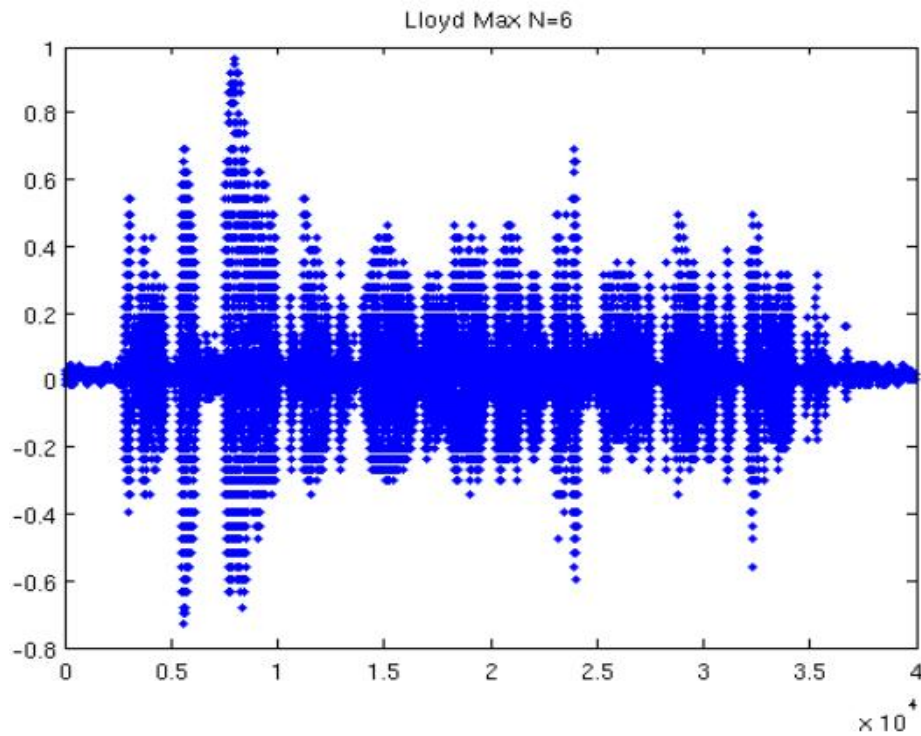
5.3.3.3 Μη ομοιόμορφη κβάντιση



Εικόνα 5-14 Κυματομορφή εξόδου για μη ομοιόμορφο κβαντιστή 2-bit



Εικόνα 5-15Κυματομορφή εξόδου για μη ομοιόμορφο κβαντιστή 4-bit



Εικόνα 5-16Κυματομορφή εξόδου για μη ομοιόμοφο κβαντιστή 6-bit

5.4 Υλοποίηση σε Matlab

Παρακάτω παραθέτουμε τον κώδικα που χρησιμοποιήσαμε σε Matlab για την εξαγωγή των παραπάνω συμπερασμάτων. Πρόκειται για ένα αυτοματοποιημένο script όπου μετά την εκτέλεση του παράγονται όλα τα παραπάνω αποτελέσματα. Οι συναρτήσεις που η υλοποίησή τους δεν έχει παρουσιαστεί παραπάνω, υπάρχει στο παράρτημα.

```
% SCRIPT FOR THE EXERCISE 2
%we print the results with long format
format long;
% we read the input signal
A=wavread('speech');
% the number of the samples of the input signal
length_x = length(A);
% the data that we print on the axis x when we design the
```

```
% desired signal
axis_x=[1:1:length_x];
% a plot of input signal
plot( axis_x, A, ' . ');
title('input signal');
%-----
-----
% MY_QUANTIZER
% initialization of the vector that keeps the SQNRs after the
% quantization with my_quantizer
SQNR_my_quantizer = zeros(3, 1);
% titles that we give to our plots
title_of_plot=['my quantizer N=2'; 'my quantizer N=4'; 'my
quantizer N=6'];
% the i is actually the N of the exercise, the bits that we
use for the
% quantization of the signal
for i = 2: 2: 6
[xq, C, P, D] = my_quantizer(A, i, 1);
SQNR_my_quantizer(i/2, 1) = sqnr(A, D);
% make a figure in a new window
figure;
% a plot for each quantized signal
plot( axis_x, xq, ' . ');
title( title_of_plot(i/2, :) );
% play the quantized signal
sound(xq)
end
% print the three different SQNRs for the corresponding
different quantized
% signal
SQNR_my_quantizer
```

```

%-----
-----
%COMPRESSOR_M
% initialization of the vector that keeps the SQNRs of the
% quantization with compressor_m->my_quantizer->decompressor_m
SQNR_compressor_m = zeros(3, 1);
% titles that we give to our plots
title_of_plot=['compressor      m      N=2'; 'compressor      m
N=4'; 'compressor m N=6' ];
% the i is actually the N of the exercise, e.g. the bits that
we use for
% the quantization of the signal
for i = 2: 2: 6
xc =compressor_m(A, 255);
[xq, F, P, D] = my_quantizer(xc, i, 1);
xd = decompressor_m(xq, 255);
D = distortion(A, xd);
SQNR_compressor_m(i/2, 1)=sqr(A, D);
% make a figure in a new window
figure;
% a plot for each quantized signal
plot( axis_x, xd, '.');
title( title_of_plot(i/2, :) );
% play the quantized signal
sound(xd)
end
% print the three different SQNRs for the corresponding
different quantized
% signal
SQNR_compressor_m
%-----
-----
%COMPRESSOR_A

```

```

% initialization of the vector that keeps the SQNRs of the
% quantization with compressor_a->my_quantizer->decompressor_a
SQNR_compressor_a = zeros(3, 1);
% titles that we give to our plots
title_of_plot=['compressor      a      N=2'; 'compressor      a
N=4'; 'compressor a N=6'];
% the i is actually the N of the exercise, the bits that we
use for the
% quantization of the signal
for i = 2: 2: 6
xc =compressor_a(A, 87.6);
[xq, F, P, D] = my_quantizer(xc, i, 1);
xd = decompressor_a(xq, 87.6);
D = distortion(A, xd);
SQNR_compressor_a(i/2, 1)=sqnr(A, D);
% make a figure in a new window
figure;
% a plot for each quantized signal
plot( axis_x, xd, '.');
title( title_of_plot(i/2, :) );
% play the quantized signal
sound(xd)
end
% print the three different SQNRs for the corresponding
different quantized
% signal
SQNR_compressor_a
%-----
-----
%LLOYD_MAX
% initialization of the vector that keeps the SQNRs of after
the
% quantization with Lloyd_Max

```

```

SQNR_Lloyd=zeros(3, 1);
% titles that we will give to our plots
title_of_plot=['SQNR N=2'; 'SQNR N=4'; 'SQNR N=6'];
% initialization of the quantized signals
xq=zeros( length_x, 3);
% the i is actually the N of the exercise, the bits that we
use for the
% quantization of the signal
for i = 2: 2: 6
[xq( :, i/2 ), C, D] = Lloyd_Max(A, i, 1);
kmax = length(D);
loops = [1:1:kmax];
% initialization of a vector that will keep all the SQNRs from
the
% iterations
Sqnr = zeros(kmax, 1);
% calculate the SQNRs from the iterations
for j = 1: 1: kmax
Sqnr(j, 1)=sqr(A, D(j, 1));
end
% make a figure in a new window
figure;
% make a plot of each of the collection of SQNRs for N=[2:2:6]
plot(loops, Sqnr, '-');
title( title_of_plot(i/2, :) );
xlabel('loops');
ylabel('SQNR');
% last SQNR value
SQNR_Lloyd(i/2, 1) = Sqnr(kmax, 1);
% play the quantized signal
sound(xd)
end
% titles that we will give to our plots

```



```
title_of_plot=['Lloyd Max N=2'; 'Lloyd Max N=4'; 'Lloyd Max  
N=6'];  
for i = 1: 1: 3  
% make a figure in a new window  
figure;  
% a plot for each quantized signal  
plot( axis_x, xq( :, i ), '.');  
title( title_of_plot(i, :) );  
end  
% print the three different SQNRs for the corresponding  
different quantized  
% signal SQNR_Lloyd
```

5.5 Κωδικοποίηση πηγής κατά Huffman

Κώδικας Huffman

Όπως γνωρίζουμε, η εντροπία H της πηγής μας δίνει ένα άνω φράγμα για την μέγιστη συμπίεση που μπορούμε να επιτύχουμε ώστε να έχουμε επακριβή ανακατασκευή του σήματος με μηδενική πιθανότητα σφάλματος.

Στην κωδικοποίηση Huffman, που είναι κωδικοποίηση από σταθερό σε μεταβλητό μήκος, απεικονίζουμε μπλοκ σταθερού μήκους σε μεταβλητού μήκους μπλοκ δυαδικών συμβόλων. Ένα κύριο πρόβλημα σε περιπτώσεις κωδικοποίησης μεταβλητού μήκους είναι ο συγχρονισμός του δέκτη με την πηγή. Για αυτό τον λόγο απαιτούμε οι κώδικες που χρησιμοποιούμε να είναι άμεση και μονοσήμαντα αποκωδικοποιήσιμοι. Ικανή και αναγκαία συνθήκη για να ικανοποιούνται οι δυο παραπάνω απαιτήσεις σε έναν κώδικα, είναι αυτός να ικανοποιεί την συνθήκη του προθέματος. Την συνθήκη αυτή την ικανοποιεί ο κώδικας Huffman και σε συνδυασμό με την ιδιότητα του να παράγει το μικρότερο μήκος λέξεων, τον τοποθετούν στην κατηγορία των

βέλτιστων κωδικοποιητών ως προς το μέσο μήκος λέξης. Την αποδοτικότητα του αλγορίθμου την υπολογίζουμε από την εξίσωση:

$$\eta = \frac{H(S)}{L}$$

Όπου η εντροπία της πηγής $H(S) = -\sum P_i \log(P_i)$ και το μέσο μήκος λέξης $L = \sum P_i L_i$. Σε περίπτωση όπου το $P_i = 0$, ορίζουμε ως $H(S_i) = 0$.

Πειραματική Αξιολόγηση

Κωδικοποιήσαμε την πηγή για $N = 4, 6$ bit μέσω της ρουτίνας Huffman. Στη συνέχεια μετρήσαμε την αποδοτικότητα του κώδικα Huffman για κάθε ένα από τα σχήματα κωδικοποίησης που αναφέραμε παραπάνω. Τα αποτελέσματα φαίνονται στον Πίνακα 6. Όπως αναμέναμε, βλέπουμε πως ο κώδικας Huffman φθάνει σχεδόν το βέλτιστο σε απόδοση πλησιάζοντας την μονάδα.

Πίνακας 6 - Αποδοτικότητα Κώδικα Huffman

	N = 4	N = 6
Ομοιόμορφο PCM	0,9912	0,9914
Συμπίεστος τύπου-μ	0,9868	0,9926
Συμπίεστος τύπου-A	0,9877	0,9936
Μη ομοιόμορφος κβαντιστής με Lloyd-Max	0,9876	0,9942

Υλοποίηση σε Matlab

```
% SCRIPT FOR THE EXERCISE 2
% we print the results in long format
format long;
% we read the input signal
A=wavread('speech');
```

```

%-----
-----
% MY_QUANTIZER
% the i is actually the N of the exercise, the bits that we
use for the
% quantization of the signal
for i = 4: 2: 6
[xq, C, P, D] = my_quantizer(A, i, 1);
[encoded_A, efficiency_my_quantizer((i-2)/2, 1)] =
encoder(A, P, i);
end
efficiency_my_quantizer
%-----
-----
%COMPRESSOR_M
% the i is actually the N of the exercise, the bits that we
use for the
% quantization of the signal
for i = 4: 2: 6
xc =compressor_m(A, 255);
[xq, F, P, D] = my_quantizer(xc, i, 1);
[encoded_A, efficiency_compressor_m((i-2)/2, 1)] =
encoder(A, P, i);
end
efficiency_compressor_m
%-----
-----
%COMPRESSOR_A
% the i is actually the N of the exercise, the bits that we
use for the
% quantization of the signal
for i = 4: 2: 6
xc =compressor_a(A, 87.6);

```

```

[xq, F, P, D] = my_quantizer(xc, i, 1);
[encoded_A, efficiency_compressor_a((i-2)/2, 1) ]=
encoder(A, P, i);
end
efficiency_compressor_a
%-----
-----
%LLOYD_MAX
% the i is actually the N of the exercise, the bits that we
use for the
% quantization of the signal
for i = 4: 2: 6
[xq, C, D] = Lloyd_Max(A, i, 1);
[encoded_A, efficiency_lloyd_max((i-2)/2, 1) ]=
encoder_lloyd_max(A, C, i);
end
efficiency_lloyd_max

```

6. Παράρτημα

Binary Search

```

function index = binarysearch(val, vector_of_intervals)
% Find the interval where the val belongs
% val: The value that is being searched into
vector_of_intervals
% vector_of_intervals: The vector that holds the intervals
% index: The index of the corresponding interval
% find the length of the vector_of_intervals
num_of_intervals = length(vector_of_intervals);
% set the limits in where the function searches for the index
in the
% vector_of_intervals

```

```

low = 1;
high = num_of_intervals + 1;
% find the expected value
index = fix((low+high)/2);
% search for the index through binary search
while (low < high) && (index~=1)
if (val >= vector_of_intervals(index-1, 1)) &&
(val < vector_of_intervals(index, 1))
return;
elseif val >= vector_of_intervals(index, 1)
low = index;
else
high = index - 1;
end
index = fix((low+high)/2);
% we have found the interval where val belongs
if (high - low) == 1
index = high;
return;
end
end
index = low;
return;

```

Συνάρτηση εύρεσης Παραμόρφωσης

```

function D = distortion(x, xq)
%function that calculate the destortion of the input signal x
in contrast
%to the quantized signal xq
D=mean( (x - xq).^2 );

```

Συνάρτηση εύρεσης SQNR

```

function SQNR = sqnr(x, D)

```

```

% sqnr is a function that calculate the ratio of our signal
divided from
% the noise of the quantization
% Input arguments
% x: input signal, already in discrete form
% D: the distortion
% Output arguments
% SQNR: the "sqnr"
SQNR=mean(x.^2)/D;
return;

```

Κωδικοποιητής Huffman

```

function [code, len]=huffman(p);
% Huffman Coding.
% [code, len]=huffman(p),
% INPUTS
% p(vector): contains the probabilities of each symbol
% OUTPUTS
% code(vector): the code for each symbol (in ascii format)
% len(vector): the number of bits needed for each code
p = p(:)';
if length(find(p<0))~=0,
error('Not a probability vector, negative component(s)')
end;
if abs(sum(p)-1)>10e-10,
error('Not a probability vector, components do not add up to
1')
end;
n=length(p);
q=p;
m=zeros(n-1, n);
for i=1:n-1,
[q, l]=sort(q);

```

```

m(i, :)= [1 (1:n-i+1), zeros(1, i-1)];
q=[q(1)+q(2), q(3:n), 1];
end;
for i=1:n-1,
c(i, :)=blanks(n*n);
end;
c(n-1,n)='0';
c(n-1, 2*n)='1';
for i=2:n-1,
c(n-i, 1:n-1)=c(n-i+1, n*(find(m(n-i+1, :)==1))-(n-2):n*(find(m(n-i+1, :)==1)));
c(n-i, n)='0';
c(n-i, n+1:2*n-1)=c(n-i, 1:n-1);
c(n-i, 2*n)='1';
for j=1:i-1,
c(n-i, (j+1)*n+1:(j+2)*n)=c(n-i+1, ...
n*(find(m(n-i+1, :)==j+1)-1)+1:n*find(m(n-i+1, :)==j+1));
end;
end;
for i=1:n,
code(i, 1:n)=c(1, n*(find(m(1, :)==i)-1)+1:find(m(1, :)==i)*n);
len(i)=length(find(abs(code(i, :))~=32));
end;

```

Κωδικοποιητής

```

function [encoded_x, efficiency] = encoder(x, P, i)
% The ecoder is a function that uses the huffman algorithm in
order to
% encode our quantized signal
% Input arguments:
% x: the quantized signal
% P: the probability of appearance of each level
% i: the number of bits that we use to quantize our signal

```

```

% Output arguments:
% encoded_x: the signal x encoded
% efficiency: the efficiency of algorithm huffman for the
specific encoding
% length of vector x
length_x = length(x);
max_value = 1;
num_of_regions = 2^i;
% the limits of each region without the "infinities"
vector_of_intervals = zeros(num_of_regions - 1, 1);
% length of each region
% each region has the same length
length_of_region = 2 * max_value / num_of_regions;
% vector that keeps the bounds of intervals
num_of_intervals = num_of_regions - 1;
for j=1:1:num_of_intervals
vector_of_intervals(j, 1) = - max_value + j *
length_of_region;
end
% encode the quantized signal
[code, len]=huffman(P);
for j = 1: 1: length_x
% find in which level the sample belongs
pos = binarysearch( x(j, 1), vector_of_intervals);
% encode each of the quantized samples of the input signal
encoded_x(j, 1) = cellstr(code(pos, :));
end
% initialization of the entropy
entropy = 0;
% the length o P vector
length_P = length(P);
%calculate the entropy
for j = 1: 1: length_P

```



```

if P(j, 1) ~= 0
entropy = entropy - P(j, 1) * log2( P(j, 1) );
end
end
% calculate the efficiency of the encoding
efficiency = entropy / sum(P .* len');
return;

```

Κωδικοποιητής Lloyd Max

```

function [encoded_x, efficiency] = encoder_lloyd_max(x, C, i)
% The encoder is a function that uses the Huffman algorithm in
order to
% encode our quantized signal
% Input arguments:
% x: the quantized signal
% C: centers of quantized region
% i: the number of bits that we use to quantize our signal
% Output arguments:
% encoded_x: the signal x encoded
% efficiency: the efficiency of algorithm Huffman for the
specific encoding
% length of vector x
length_x = length(x);
max_value = 1;
num_of_regions = 2^i;
num_of_intervals = num_of_regions - 1;
% initialization of the vector of intervals
vector_of_intervals = zeros(num_of_intervals, 1);
% calculate the bounds of each region
for j = 1: 1: num_of_intervals
vector_of_intervals(j, 1) = ( C(j, 1) + C( j + 1, 1) ) / 2;
end

```

```

% initialization of the vector with the occurrences
occurrences = zeros(num_of_regions, 1);
for j = 1: 1: length_x
pos = binarysearch(x(j, 1), vector_of_intervals);
occurrences(pos, 1) = occurrences(pos, 1) + 1;
end
% initialization of the vector with the probabilities
P = zeros(num_of_regions, 1);
for j = 1: 1: num_of_regions
P(j, 1) = occurrences(j, 1) / length_x;
end
% encode the quantized signal
[code, len]=huffman(P);
for j = 1: 1: length_x
% find in which level the sample belongs
pos = binarysearch( x(j, 1), vector_of_intervals);
% encode each of the quantized samples of the input signal
encoded_x(j, 1) = cellstr(code(pos, :));
end
% initialization of the entropy
entropy = 0;
% the length o P vector
length_P = length(P);
%calculate the entropy
for j = 1: 1: length_P
if P(j, 1) ~= 0
entropy = entropy - P(j, 1) * log2( P(j, 1) );
end
end
% calculate the efficiency of the encoding
efficiency = entropy / sum(P .* len');
return;

```

Αναφορές

- s Καλουμπιτσιδης Νίκος, *Σήματα Συστήματα και Αλγόριθμοι*, εκδ. Δίαυλος , Αθήνα 1994 (5^η έκδοση).
- s Σ.Ν Δημητριάδης, Α.Σ Πομπόρτσης, Ε.Γ Τριανταφύλλου, *Τεχνολογία Πολυμέσων*, εκδ. Τσίολα, Θεσσαλονίκη 2004.
- s British Library, *A manual of sound Archive Administration*, Alan Word 1990.
- s Σημειώσεις Μαθήματος: Φλώρος Αντρέας, *Τεχνολογία Ψηφιακού Ήχου*.
- s Ηλιάδου Ελεάνα, *Δημιουργία Ψηφιακού Αρχείου* του Γιάννη Χρήστου.
- s Ηλιάδου Ελεάνα, *Μετεπεξεργασία Δημιουργία Πρωτοτύπου: Καταγραφή και Αρχαιοθήτηση του συνόλου των ηχογραφημένων συναυλιών του Τμήματος Μουσικών Σπουδών του Ιονίου Πανεπιστημίου*, Κέρκυρα 2000
- s Χρυσάνθου Μαρία, *Διαδικασία Ψηφιοποίησης Ηχητικών Αρχείων και Τεχνολογίες Ψηφιοποιημένου Ήχου*, Κέρκυρα 2006
- s Μανωλόπουλος Βασίλειος, *Διάθεση πολυμέσων σε ετερογενή δίκτυα κατά το πρότυπο MPEG21*, Θεσσαλονίκη 2007