

ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ

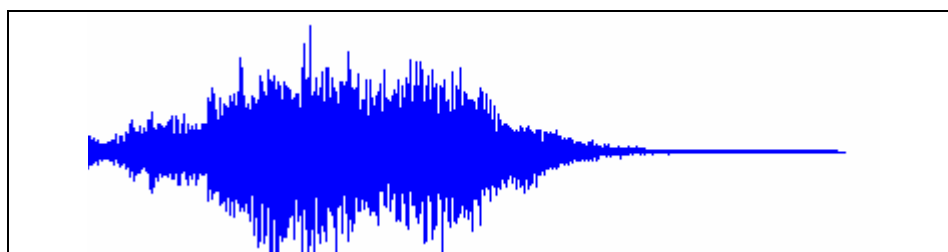
ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΧΑΤΖΗΧΡΗΣΤΟΥ ΧΡΗΣΤΟΣ

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 502308

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: «ΑΝΑΠΤΥΞΗ ΑΛΓΟΡΙΘΜΩΝ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΟΣ ΣΤΟ ΜΑΤΛΑΒ ΓΙΑ ΤΗ ΣΥΜΠΙΕΣΗ ΗΧΟΥ»



ΕΠΙΒΛΕΠΟΥΣΑ: ΠΑΠΑΣΤΕΡΓΙΟΥ ΑΝΑΣΤΑΣΙΑ

*Στην οικογένειά
μου*

ΠΡΟΛΟΓΟΣ

Η συγκεκριμένη πτυχιακή εργασία πραγματοποιείται στα πλαίσια του μαθήματος της Ψηφιακής Επεξεργασίας Σήματος. Κατά καιρούς πολλοί ερευνητές έχουν ασχοληθεί με τον ήχο. Ο βασικότερος λόγος για την ενασχόληση με αυτό το διαχρονικό αντικείμενο είναι πως η χρήση του στις μέρες μας είναι πολύ σημαντική. Χάρη σε αυτόν έχουμε καταφέρει σήμερα να ενημερωνόμαστε για όλα τα θέματα που μας αφορούν και να ακούμε από το ραδιόφωνο και τα mp3 τη μουσική που μας αρέσει.

Μέσα από την εργασία αυτή θα εξεταστούν οι διάφορες μορφές συμπίεσης του ήχου με στόχο να διαπιστωθεί στο τέλος ποια είναι η καλύτερη δυνατή. Στην πορεία θα αναπτυχθούν αλγόριθμοι σε γλώσσα προγραμματισμού MATLAB και με τη βοήθεια των μετασχηματισμών DCT και DOGHT θα εξεταστούν, κατόπιν, τα τελικά αποτελέσματα. Είναι, επίσης, σημαντικό να αναφερθεί πως η παρούσα έρευνα θα επικεντρωθεί κυρίως στη συμπίεση των ψηφιακών ηχητικών σημάτων.

Η εργασία είναι χωρισμένη σε δύο μέρη: το θεωρητικό και το πρακτικό. Το θεωρητικό μέρος ξεκινάει με αναλυτική περιγραφή του ήχου ως φυσικού φαινομένου καθώς και των προβλημάτων που μπορεί να δημιουργηθούν από την αλόγιστη χρήση του. Στη συνέχεια, γίνεται εκτενής αναφορά στο θεώρημα δειγματοληψίας και στον κβαντισμό σημάτων. Ακολούθως, περιγράφονται ορισμένοι μετατροπείς αναλογικού σήματος σε ψηφιακό και αντιστρόφως, όπως και κάποιες κατηγορίες προτύπων ήχου με πιο γνωστές το πρότυπο MPEG και το WAV, τα οποία και αναλύονται. Αμέσως μετά, περιγράφεται η μέθοδος συμπίεσης σημάτων χρησιμοποιώντας το διακριτό ορθογώνιο μετασχηματισμό GAUSS-HERMITE. Τέλος, παρουσιάζεται η γλώσσα προγραμματισμού MATLAB, η οποία θα χρησιμοποιηθεί στο πρακτικό μέρος.

Το πρακτικό μέρος της πτυχιακής εργασίας βασίζεται στο σύνολό του στο MATLAB. Μέσα από τους αλγόριθμους που αφορούν τους μετασχηματισμούς DCT και GAUSS-HERMITE γίνεται σύγκριση των διαφόρων αποτελεσμάτων. Αρχικά, βλέπουμε τις τιμές της ενέργειας για τις δυο περιπτώσεις μετασχηματισμών. Έπειτα, διατυπώνονται τρία παραδείγματα για εξισώσεις διαφόρων μορφών, αλλά κατά κύριο λόγο ημιτονοειδών κυματομορφών καταλήγοντας έτσι σε χρήσιμα συμπεράσματα

για τη χρήση των δυο μετασχηματισμών. Στο τέλος αυτού του μέρους αναλύονται τρία ακόμη παραδείγματα, τα οποία αφορούν συγκεκριμένα στον ήχο. Σε όλα τα παραδείγματα που αφορούν εξισώσεις υπολογίζονται και οι τιμές συμπίεσης, ενώ σε αυτά που αφορούν στον ήχο η συμπίεση φαίνεται τόσο σε κυματομορφή όσο και ακουστικά.

Ύστερα από τη συνοπτική ανάλυση που έγινε παραπάνω για το περιεχόμενο της πτυχιακής εργασίας, προκύπτει ήδη εξ αρχής η ιδιαίτερη χρησιμότητά της σε όλους τους αναγνώστες. Αυτός είναι και ο λόγος, για τον οποίο η γλώσσα του κειμένου είναι απλή και λιτή. Με αυτόν τον τρόπο καθίσταται δυνατή η πλήρης κατανόηση του κειμένου, παρά τις όποιες δυσκολίες μπορεί να προκύπτουν κατά την πρώτη αντιμετώπιση ενός τέτοιου θέματος.

Βέβαια νιώθω υποχρεωμένος να πω ένα μεγάλο ευχαριστώ σε όλους εκείνους που βρίσκονταν κοντά μου σε όλο αυτό το διάστημα διεκπεραίωσης της ερευνητικής μου δραστηριότητας καθώς και κατά τη συγγραφή της. Θα ήθελα, λοιπόν, να εκφράσω αρχικά τις ευχαριστίες μου στην επιβλέπουσα καθηγήτριά μου, την κ. Παπαστεργίου Νατάσσα, που σε οποιαδήποτε δυσκολία είχα στην εργασία με βοήθησε να βρω τη σωστή λύση. Τέλος, θα ήθελα να ευχαριστήσω και την οικογένειά μου για την αμέριστη συμπαράσταση της.

ΠΕΡΙΕΧΟΜΕΝΑ

	σελ.
<u>ΠΡΟΛΟΓΟΣ</u>	i
<u>ΠΕΡΙΕΧΟΜΕΝΑ</u>	iii
<u>ΚΕΦΑΛΑΙΟ 1</u>	
<u>ΓΕΝΙΚΑ ΓΙΑ ΤΟΝ ΗΧΟ</u>	1
1.1 ΟΡΙΣΜΟΣ ΗΧΟΥ-ΦΥΣΙΚΑ ΜΕΓΕΘΗ.....	1
1.2 ΠΑΡΑΓΩΓΗ ΗΧΗΤΙΚΩΝ ΚΥΜΑΤΩΝ.....	2
1.3 ΜΕΓΕΘΗ ΑΝΤΙΛΗΨΗΣ ΗΧΟΥ.....	2
1.4 ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ ΗΧΟΥ.....	3
<u>ΚΕΦΑΛΑΙΟ 2</u>	
<u>Ο ΗΧΟΣ ΣΕ ΨΗΦΙΑΚΗ ΜΟΡΦΗ</u>	4
2.1 ΜΕΤΑΤΡΟΠΕΙΣ ΗΛΕΚΤΡΙΚΟΥ ΣΗΜΑΤΟΣ ΣΕ ΗΧΗΤΙΚΟ ΚΥΜΑ ΚΑΙ ΑΝΤΙΣΤΡΟΦΩΣ.....	4
2.2 ΤΟ ΘΕΩΡΗΜΑ ΔΕΙΓΜΑΤΟΛΗΨΙΑΣ.....	4
2.3 ΖΩΝΟΠΕΡΑΤΑ ΣΗΜΑΤΑ (BANDPASS SIGNALS).....	9
2.4 ΔΙΑΚΡΙΤΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER.....	14
2.5 ΚΒΑΝΤΙΣΜΟΣ ΣΗΜΑΤΩΝ.....	17
2.6 ΣΦΑΛΜΑ ΚΒΑΝΤΙΣΜΟΥ.....	21
2.7 ΨΗΦΙΟΠΟΙΗΣΗ.....	23
2.8 ΜΕΤΑΤΡΟΠΕΙΣ ΑΠΟ ΑΝΑΛΟΓΙΚΟ ΣΕ ΨΗΦΙΑΚΟ ΣΗΜΑ (ANALOG TO DIGITAL CONVERTERS A/D).....	24
2.9 ΜΕΤΑΤΡΟΠΕΙΣ ΑΠΟ ΨΗΦΙΑΚΟ ΣΕ ΑΝΑΛΟΓΙΚΟ ΣΗΜΑ (DIGITAL TO ANALOG CONVERTERS D/A).....	25
2.10 ΥΠΟΛΟΓΙΣΤΗΣ ΚΑΙ ΗΧΟΣ.....	26
2.11 ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΤΥΠΩΝ ΗΧΟΥ.....	26
2.12 ΠΡΟΤΥΠΟ WAVE (WAV).....	27
2.13 ΠΡΟΤΥΠΟ MOTION PICTURE EXPERTS GROUP (MPEG).....	27

<u>ΚΕΦΑΛΑΙΟ 3</u>	
<u>ΜΕΘΟΔΟΣ ΣΥΜΠΙΕΣΗΣ ΣΗΜΑΤΩΝ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟ</u>	
<u>ΔΙΑΚΡΙΤΟ ΟΡΘΟΓΩΝΙΟ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟ</u>	
<u>GAUSS-HERMITE</u>	29
3.1 ΕΙΣΑΓΩΓΗ.....	29
3.2 ΠΕΡΙΓΡΑΦΗ.....	31
<u>ΚΕΦΑΛΑΙΟ 4</u>	
<u>MATLAB</u>	34
4.1 ΤΟ MATLAB ΩΣ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	34
4.2 ΜΕΤΑΒΛΗΤΕΣ ΣΤΟ MATLAB.....	35
4.3 ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ.....	35
4.4 ΠΙΝΑΚΕΣ.....	36
4.5 M-ΑΡΧΕΙΑ ΣΤΟ ΠΡΟΓΡΑΜΜΑ MATLAB.....	36
4.6 ΣΥΝΑΡΤΗΣΕΙΣ.....	37
4.7 ΑΠΟΘΗΚΕΥΣΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ.....	37
4.8 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ.....	38
4.9 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΗΧΟΥ.....	87
<u>ΣΥΜΠΕΡΑΣΜΑΤΑ</u>	111
<u>ΠΕΡΙΛΗΨΗ</u>	114
<u>ΒΙΒΛΙΟΓΡΑΦΙΑ</u>	115

ΚΕΦΑΛΑΙΟ 1

ΓΕΝΙΚΑ ΓΙΑ ΤΟΝ ΗΧΟ

Σε αυτό το κεφάλαιο θα μιλήσουμε γενικά για τον ήχο. Θα προσπαθήσουμε να δώσουμε έναν ορισμό για τον ήχο και να αναλύσουμε όσο μπορούμε το φαινόμενο αυτό.

1.1 ΟΡΙΣΜΟΣ ΗΧΟΥ-ΦΥΣΙΚΑ ΜΕΓΕΘΗ

Σε όλους μας είναι γνωστός ο ήχος ως έννοια, αλλά στο ερώτημα: «*τι είναι ο ήχος;*», λίγοι είναι αυτοί που μπορούν να δώσουν έναν ακριβή ορισμό. Ο ήχος παράγεται ως αποτέλεσμα συνεχών εναλλαγών πύκνωσης και αραιώσης των μορίων του αέρα που μεταβάλλονται ομοιοτρόπως. Η δημιουργία αυτών των συνεχόμενων πυκνωμάτων και αραιωμάτων οδηγεί στην παραγωγή ενός ηχητικού κύματος που διαδίδεται σε όλα τα ελαστικά μέσα, όπως στον αέρα, και μεταφέρεται προς όλες τις κατευθύνσεις. Έτσι, λοιπόν, αντιλαμβανόμαστε πως «ήχος είναι η διαταραχή ενός ελαστικού μέσου που διαδίδεται μέσα σε αυτό και προς όλες τις κατευθύνσεις». Η ταχύτητα διάδοσης της διαταραχής αυτής ονομάζεται ταχύτητα διάδοσης του ηχητικού κύματος, συμβολίζεται με c και εξαρτάται από τα χαρακτηριστικά του μέσου διάδοσης. Όσον αφορά στη διάδοση του ήχου στον αέρα, η ταχύτητα διάδοσης του ηχητικού κύματος προκύπτει από τον ακόλουθο τύπο:

$$C=20.05*\sqrt{T} \text{ m/sec}$$

όπου T η θερμοκρασία του αέρα σε Kelvin (K). Στη συγκεκριμένη, μάλιστα, περίπτωση $T=347\text{m/sec}$ σε ιδανικές συνθήκες θερμοκρασίας.

Τα ηχητικά κύματα, όταν επαναλαμβάνονται ανά τακτά χρονικά διαστήματα, ονομάζονται περιοδικά. Περίοδος ενός ηχητικού κύματος ονομάζεται ο χρόνος μεταξύ δυο διαδοχικών επαναλήψεων, συμβολίζεται με T και μετριέται σε δευτερόλεπτα (sec). Το αντίστροφο της περιόδου ονομάζεται συχνότητα, συμβολίζεται με f και μετριέται σε επαναλήψεις ή αλλιώς κύκλους ανά δευτερόλεπτο (Hz). Θα πρέπει, επίσης, να αναφέρουμε

εδώ και το μήκος κύματος που συμβολίζεται με λ και μετριέται σε (m). Οι σχέσεις που ενώνουν τα παραπάνω μεγέθη είναι οι ακόλουθες:

$$f=1/T$$

$$\lambda=c/f$$

[11]

1.2 ΠΑΡΑΓΩΓΗ ΗΧΗΤΙΚΩΝ ΚΥΜΑΤΩΝ

Για να επιτύχουμε την παραγωγή των ηχητικών κυμάτων, απαιτείται, οπωσδήποτε, η δημιουργία δονήσεων μέσα στο μέσο, στο οποίο επιχειρείται η διαδικασία αυτή. Οι δονήσεις αυτές προκύπτουν από τις ηχητικές πηγές. Ένα κλασικό παράδειγμα ηχητικής πηγής είναι οι φωνητικές χορδές του ανθρώπου που πάλλονται και δημιουργούν πυκνώματα και αραιώματα των μορίων του αέρα μπροστά από το στόμα. Τα σημαντικότερα φαινόμενα που εμφανίζονται κατά την παραγωγή των ηχητικών κυμάτων είναι η ανάκλαση, η διάχυση και η περίθλαση. [11]

1.3 ΜΕΓΕΘΗ ΑΝΤΙΛΗΨΗΣ ΗΧΟΥ

Στην καθημερινή μας ζωή αντιλαμβανόμαστε πολλές φορές πως δεν είμαστε σε θέση να αξιολογήσουμε αντικειμενικά την ποιότητα του ήχου. Ο καθένας μας έχει διαφορετικά κριτήρια, με τα οποία κάνει την κριτική του. Ένα πολύ χαρακτηριστικό παράδειγμα αποτελούν οι οδηγοί των αυτοκινήτων που βάζουν δυνατά τη μουσική έχοντας ένα ηχείο στο πίσω μέρος του οχήματος. Αυτοί θέλουν να δουν, αν έχει καλή ποιότητα ο ήχος που βγάζει το ηχείο. Αυτή, όμως, η μέτρηση είναι εντελώς υποκειμενική, αφού συμμετέχει ο ανθρώπινος παράγοντας. Ορισμένα μεγέθη που πρέπει να προσέχουμε κάθε φορά που θέλουμε να μετρήσουμε την ποιότητα του ήχου είναι η ακουστότητα, το τονικό ύψος και η χροιά, με τα οποία όμως δε θα ασχοληθούμε στη συνέχεια. [11]

1.4 ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ ΗΧΟΥ

Πέρα από τις όποιες δυνατότητες μας προσφέρει ο ήχος, είναι γνωστό σε όλους πως μας επιφέρει και αρκετά προβλήματα. Εμείς θα ασχοληθούμε με το πλέον γνωστό που είναι η βαρηκοΐα.

Είναι ευρύτατα γνωστό πως ο άνθρωπος μπορεί να ακούει συχνότητες συγκεκριμένων ορίων. Συνήθως, ανώτατο όριο των συχνοτήτων αυτών αποτελούν τα 120dB. Στο σημείο αυτό θα αισθανθούμε ενοχλήσεις στο αυτί, γιατί είμαστε ακριβώς πάνω στο όριο. Σε συχνότητες μεγαλύτερες από αυτή των 120dB υπάρχει σοβαρή περίπτωση να προκληθεί μεγάλη φθορά στην ακοή του ανθρώπου. Προβλήματα με την ακοή, επίσης, μπορούμε να έχουμε λόγω ηλικίας ή και επαγγελματικής δραστηριότητας. Ένα πολύ απλό παράδειγμα που αναφέρεται στην επαγγελματική δραστηριότητα είναι ο DJ σε μια καφετερία. Όταν είναι επί ώρες σε ένα χώρο που η μουσική παίζει δυνατά, τότε αρχίζει και μειώνεται η ακοή του. Αυτό συμβαίνει, ιδιαίτερα, όταν γίνεται σε καθημερινή βάση και για πολλές ώρες. Μπορεί, όμως, να αποφευχθεί, αν ο οργανισμός προλαβαίνει να επαναφέρει την ευαισθησία ακοής στα κανονικά για αυτόν επίπεδα. Για να το πετύχει αυτό, ο DJ θα πρέπει να δουλεύει λιγότερες μέρες και ώρες. Και πάλι, όμως, η επαναφορά της ακοής στα κανονικά επίπεδα θα έρθει μετά από καιρό, αν και δεν είναι απόλυτα σίγουρο, γιατί ύστερα από ένα όριο η ευαισθησία της ακοής δεν επανέρχεται καθόλου.

[11]

ΚΕΦΑΛΑΙΟ 2

Ο ΗΧΟΣ ΣΕ ΨΗΦΙΑΚΗ ΜΟΡΦΗ

Στο πρώτο κεφάλαιο αναφερθήκαμε γενικά στον ήχο. Προσπαθήσαμε να δώσουμε έναν κατανοητό προς όλους ορισμό και αναλύσαμε ένα από τα προβλήματα που δημιουργεί. Στο παρόν κεφάλαιο θα ασχοληθούμε με τον ήχο στην ψηφιακή του μορφή.

2.1 ΜΕΤΑΤΡΟΠΕΙΣ ΗΛΕΚΤΡΙΚΟΥ ΣΗΜΑΤΟΣ ΣΕ ΗΧΗΤΙΚΟ ΚΥΜΑ ΚΑΙ ΑΝΤΙΣΤΡΟΦΩΣ

Οι ηλεκτρακουστικοί μετατροπείς διακρίνονται σε δυο μεγάλες κατηγορίες. Στην πρώτη κατηγορία ανήκουν αυτοί που μετατρέπουν τα ηχητικά κύματα σε ηλεκτρικά σήματα. Τέτοιου είδους μετατροπείς είναι τα μικρόφωνα. Αντίθετα, στη δεύτερη κατηγορία ανήκουν αυτοί που επιτελούν την αντίστροφη διαδικασία. Μετατρέπουν, δηλαδή, τα ηλεκτρικά σήματα σε ηχητικά κύματα και τέτοιοι μετατροπείς είναι τα μεγάφωνα. Είναι εύκολο να συμπεράνουμε πως για οποιαδήποτε λειτουργία εγγραφής, αναπαραγωγής αλλά και επεξεργασίας του ήχου οι μετατροπείς παίζουν το βασικότερο ρόλο, ασχέτως αν η δουλειά που θέλουμε να κάνουμε είναι σε αναλογική ή ψηφιακή μορφή. [11]

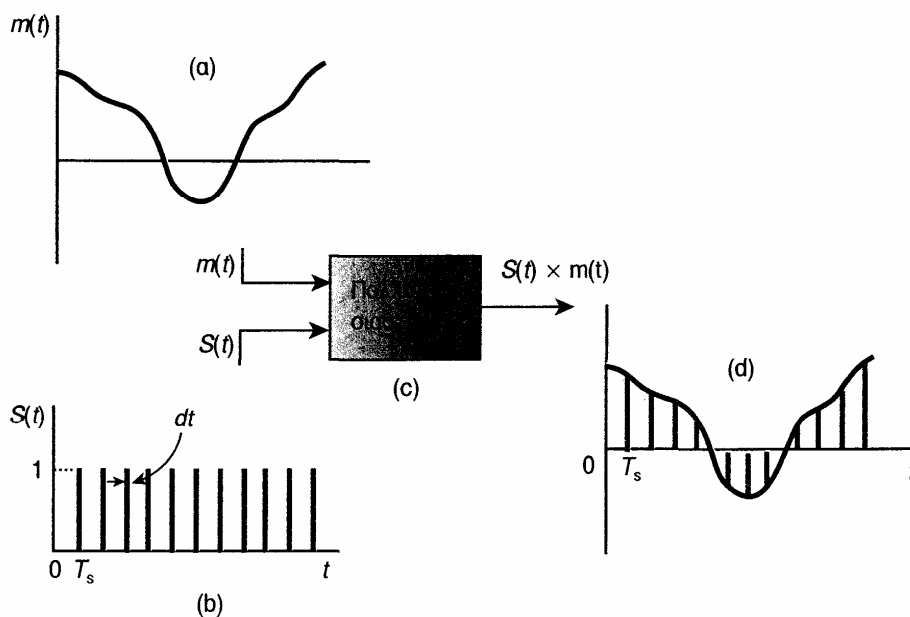
2.2 ΤΟ ΘΕΩΡΗΜΑ ΔΕΙΓΜΑΤΟΛΗΨΙΑΣ

Θα προχωρήσουμε, παρακάτω, στο θεώρημα δειγματοληψίας που αποτελεί τη θεμελιώδη αρχή των ψηφιακών επικοινωνιών:

Έστω $m(t)$ ένα σήμα που είναι περιορισμένου εύρους ζώνης, τόσο όσο η φασματική συνιστώσα της υψηλότερης συχνότητας να είναι f_M . Έστω ότι οι τιμές του $m(t)$ έχουν καθοριστεί σε κανονικά διαστήματα, τα οποία χωρίζονται με χρόνους $T_S \leq (1/2 * f_M)$. Το σήμα, δηλαδή, υφίσταται *περιοδικά* δειγματοληψία κάθε T_S δευτερόλεπτα. Υπό αυτές τις συνθήκες τα δείγματα $m(nT_S)$, στα οποία το n αντιστοιχεί σε ακέραιο αριθμό, καθορίζουν το σήμα με τρόπο μοναδικό και το σήμα είναι δυνατό να επανακτηθεί από αυτά δίχως να παραμορφωθεί.

Ο χρόνος T_s ονομάζεται *χρόνος δειγματοληψίας*. Οφείλουμε να σημειώσουμε πως το θεώρημα αυτό απαιτεί να είναι ιδιαίτερα ταχύς ο *ρυθμός δειγματοληψίας*, ώστε να μπορούμε να είμαστε σε θέση να λαμβάνουμε τουλάχιστον δυο δείγματα κατά τη διάρκεια της περιόδου που αναλογεί στη φασματική συνιστώσα υψηλότερης συχνότητας. Ακολούθως, θα προχωρήσουμε στην απόδειξη του θεωρήματος περιγράφοντας τον τρόπο με τον οποίο μπορεί το σήμα να ανακτηθεί από τα δείγματα του. Στην πρώτη εικόνα του σχήματος που ακολουθεί μπορούμε να δούμε το σήμα $m(t)$ βασικής ζώνης, το οποίο θα υποστεί τη δειγματοληψία. Παράλληλα, στη δεύτερη εικόνα του ίδιου σχήματος παρουσιάζεται μια περιοδική ακολουθία παλμών $S(t)$ μοναδιαίου πλάτους και περιόδου T_s . Οι παλμοί αυτοί είναι στενοί κατά τρόπο αυθαίρετο και εύρος dt . Όπως φαίνεται, εξάλλου, στην τρίτη εικόνα του σχήματος, τα σήματα $m(t)$ και $S(t)$ εφαρμόζονται σε ένα πολλαπλασιαστή που μας δίνει τελικά ως έξοδο το γινόμενο $S(t)*m(t)$. Το γινόμενο αυτό, όπως διαπιστώνουμε από την τέταρτη εικόνα του σχήματος, ταυτίζεται με το σήμα $m(t)$ που έχει υποστεί δειγματοληψία τη στιγμή κατά την οποία εμφανίζεται κάθε παλμός. Συμπεραίνουμε, λοιπόν, από τα παραπάνω πως, κάθε φορά που παρουσιάζεται ένας παλμός, η έξοδος του πολλαπλασιαστή έχει την ίδια ακριβώς τιμή με το $m(t)$. Απεναντίας, η έξοδος του πολλαπλασιαστή παραμένει μηδενική καθ' όλο το υπόλοιπο χρονικό διάστημα.

Επομένως, το σήμα $S(t)$ είναι περιοδικό, με περίοδο T_s και αναπτύσσεται σε σειρά Fourier.



Σχήμα 2.1 (a) Ένα σήμα $m(t)$ που πρόκειται να υποστεί δειγματοληψία. (b) Η συνάρτηση δειγματοληψίας $S(t)$ αποτελείται από μια ακολουθία πολύ στενών παλμών μοναδιαίου πλάτους. (c) Η διαδικασία της δειγματοληψίας επιτελείται σε έναν πολλαπλασιαστή. (d) Τα δείγματα του σήματος $m(t)$.

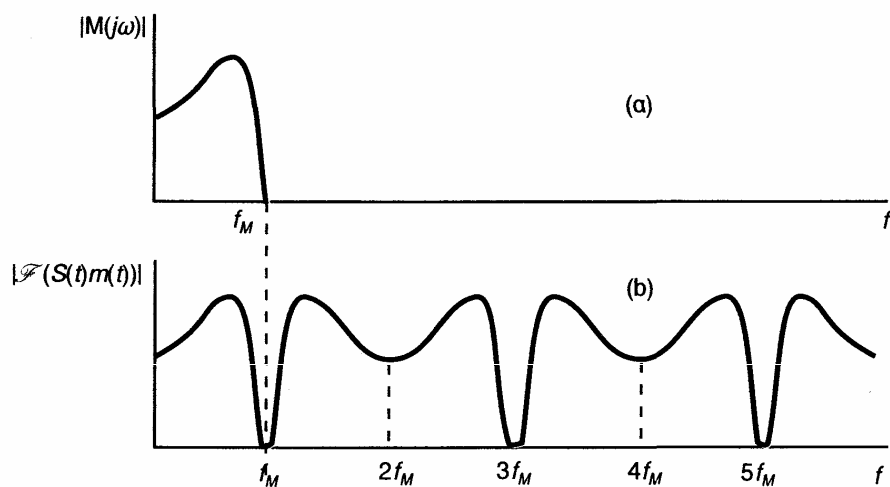
Στην περίπτωση, κατά την οποία $T_s = (1/2 * f_M)$, το γινόμενο $S(t)$ εξάγεται από την παρακάτω σχέση:

$$S(t)m(t) = \frac{dt}{T_s} m(t) + \frac{dt}{T_s} [2m(t) \cos 2\pi(2f_M)t + 2m(t) \cos 2\pi(4f_M)t + \dots] \quad (\text{Σχέση 2.1})$$

Μπορούμε να παρατηρήσουμε πλέον ότι ο πρώτος όρος της σειράς είναι το ίδιο σήμα $m(t)$, με εξαίρεση ένα σταθερό παράγοντα. Επιπροσθέτως, ο δεύτερος όρος, εκτός του ότι αποτελεί ένα πολλαπλασιαστικό παράγοντα, ισούται με το γινόμενο του $m(t)$ και ενός ημιτονοειδούς σήματος, το οποίο έχει συχνότητα $2 * f_M$. Το γινόμενο αυτό έχει ως αποτέλεσμα ένα σήμα διπλής πλευρικής ζώνης με κατεσταλμένη φέρουσα συχνότητας $2 * f_M$. Παρομοίως, οι διαδοχικοί όροι δίνουν σήματα με συχνότητες $4 * f_M$, $6 * f_M$ κ.τ.λ.

Έστω ότι το σήμα $m(t)$ είναι περιορισμένου εύρους ζώνης στο διάστημα συχνοτήτων κάτω από την f_M . Το φάσμα του πρώτου όρου της εξίσωσης εκτείνεται από 0 έως f_M . Από την άλλη μεριά, το φάσμα του δεύτερου όρου εμφανίζεται συμμετρικό γύρω από τη συχνότητα $2 * f_M$ και

κυμαίνεται από $2 \cdot f_M - f_M = f_M$ έως $2 \cdot f_M + f_M = 3 \cdot f_M$. Η μορφή, λοιπόν, με την οποία παρουσιάζεται στο σύνολο του το φάσμα του σήματος που έχει υποστεί δειγματοληψία είναι ξεκάθαρη στο σχήμα που ακολουθεί. Μπορούμε τότε να υποθέσουμε πως το σήμα από δειγματοληψία διήλθε μέσα από ένα χαμηλοπερατό φίλτρο έχοντας συχνότητα αποκοπής f_M . Εάν η μετάδοση του φίλτρου ήταν τελείως σταθερή στη ζώνη διέλευσης και η αποκοπή ήταν απείρως οξεία στην f_M , το φίλτρο θα επέτρεπε να περάσει μόνο το σήμα $m(t)$ και τίποτα άλλο.



Σχήμα 2.2 (α) Η γραφική παράσταση του μέτρου της φασματικής πυκνότητας ενός σήματος περιορισμένου εύρους ζώνης στην f_M (β) Γραφική παράσταση του πλάτους του φάσματος του σήματος από δειγματοληψία.

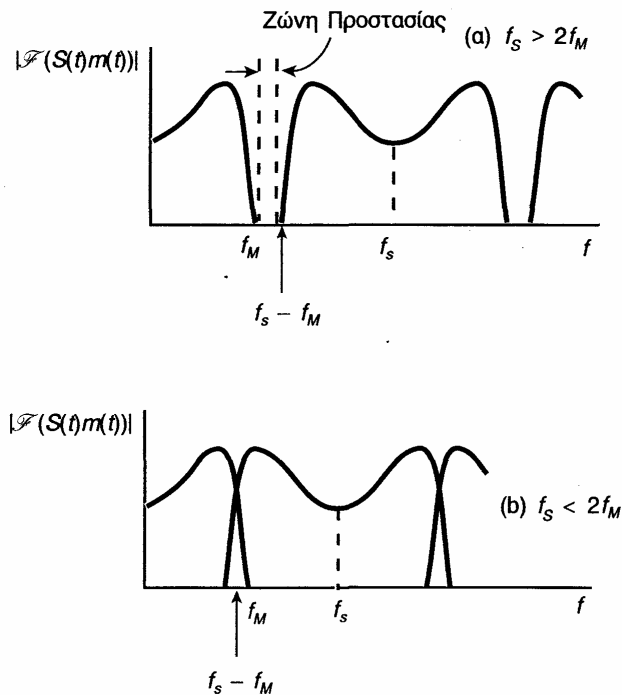
Στο επόμενο σχήμα παρατηρούμε το φασματικό διάγραμμα που αντιστοιχεί στο προαναφερθέν σχήμα, μόνο για την περίπτωση κατά την οποία ο ρυθμός δειγματοληψίας $f_S = (1/T_S)$ είναι μεγαλύτερος από $2 \cdot f_M$. Σε αυτή την περίπτωση, όμως, εμφανίζεται ένα χάσμα ανάμεσα στο άνω όριο f_M του φάσματος του σήματος βασικής ζώνης και στο κάτω όριο του φάσματος, το οποίο έχει κέντρο γύρω από τη συχνότητα φέροντος $f_S = 2 \cdot f_M$. Εξαιτίας αυτού του χάσματος το χαμηλοπερατό φίλτρο που χρησιμοποιήθηκε προκειμένου να επιλεγεί το σήμα $m(t)$, δεν είναι αναγκαίο να έχει μια απείρως οξεία αποκοπή. Αντιθέτως, η εξασθένηση του φίλτρου είναι δυνατόν να αρχίζει στην f_M , αλλά να μην είναι απαραίτητο να πλησιάζει μια υψηλή τιμή ως τη συχνότητα $f_S - f_M$. Το διάστημα αυτό από f_M έως $f_S - f_M$ ονομάζεται *ζώνη προστασίας* και είναι

πάντοτε άκρως απαραίτητο στην πράξη, στην περίπτωση που ένα φίλτρο με απείρως οξεία αποκοπή είναι, φυσικά, μη πραγματώσιμο. Οφείλουμε να αναφέρουμε αντιπροσωπευτικά ότι, όταν εφαρμόζουμε δειγματοληψία που σχετίζεται με μηνύματα φωνής στις γραμμές του τηλεφώνου, το σήμα φωνής περιορίζεται στην $f_M=3.3\text{KHz}$, ενώ η f_S στα 8KHz . Στην περίπτωση αυτή η ζώνη προστασίας ανέρχεται σε $8-2*3.3=1.4\text{KHz}$.

Η κατάσταση που αναπαρίσταται στο ακόλουθο σχήμα προκύπτει όταν $f_S \leq 2*f_M$. Μπορούμε εύκολα να διαπιστώσουμε ότι στη συγκεκριμένη περίπτωση υπάρχει επικάλυψη μεταξύ των δυο φασμάτων. Συνεπώς, η ακριβής ανάκτηση του $m(t)$ δε μπορεί να επιτραπεί από καμιά λειτουργία φιλτραρίσματος.

Εφόσον έχουμε καταλήξει στο συμπέρασμα ότι το σήμα που έχει υποστεί, αρχικά, δειγματοληψία και είναι δυνατόν να ανακτηθεί επακριβώς, όταν $T_S \leq 1/(2*f_M)$, καταφέραμε να αποδείξουμε το *θεώρημα δειγματοληψίας*. Επιπλέον, δείξαμε παραπάνω για ποιο λόγο ο ελάχιστος ρυθμός δειγματοληψίας είναι $2*f_M$. Ο ελάχιστος ρυθμός δειγματοληψίας είναι γνωστός στην επιστημονική κοινότητα ως *ρυθμός Nyquist*. Αν αυξήσουμε το ρυθμό δειγματοληψίας περισσότερο από το ρυθμό Nyquist, τότε το εύρος της ζώνης προστασίας αυξάνεται διευκολύνοντας με αυτό τον τρόπο την εύρεση λύσης στο πρόβλημα του φιλτραρίσματος. Θα δούμε όμως παρακάτω ότι μια αύξηση στο ρυθμό έχει ως αποτέλεσμα την επέκταση του εύρους ζώνης που απαιτείται για τη μετάδοση του σήματος, το οποίο έχει υποστεί δειγματοληψία. Κατά συνέπεια, κρίνεται απαραίτητος ο συμβιβασμός των δυο κατά την εφαρμογή. Ιδιαίτερο ενδιαφέρον παρουσιάζει η ειδική περίπτωση, κατά την οποία η δειγματοληψία αφορά σε ένα ημιτονοειδές σήμα με συχνότητα f_M . Σε αυτή την περίπτωση, όλη η ισχύς του σήματος συγκεντρώνεται ακριβώς στη συχνότητα αποκοπής του χαμηλοπερατού φίλτρου, και, επομένως, είναι κάπως αμφίβολο, αν η συχνότητα του σήματος βρίσκεται εντός ή εκτός της ζώνης διέλευσης του φίλτρου. Για να απομακρυνθεί αυτή η αμφιβολία ορίζουμε ως απαιτούμενη τη σχέση: $f_S > 2*f_M$ αντί της $f_S \geq 2*f_M$. Η αναγκαιότητα αυτής της συνθήκης γίνεται εύκολα αντιληπτή, αν υποθέσουμε ότι $f_S = 2*f_M$, αλλά τη στιγμή που το ημιτονοειδές σήμα περνάει από το μηδέν, λαμβάνουμε ένα αρχικό δείγμα. Τότε, μαζί με αυτό, και όλα τα άλλα διαδοχικά δείγματα θα

είναι, επίσης, μηδέν. Η κατάσταση αυτή μπορεί να αποφευχθεί, εφόσον απαιτήσουμε $f_s > 2f_M$.



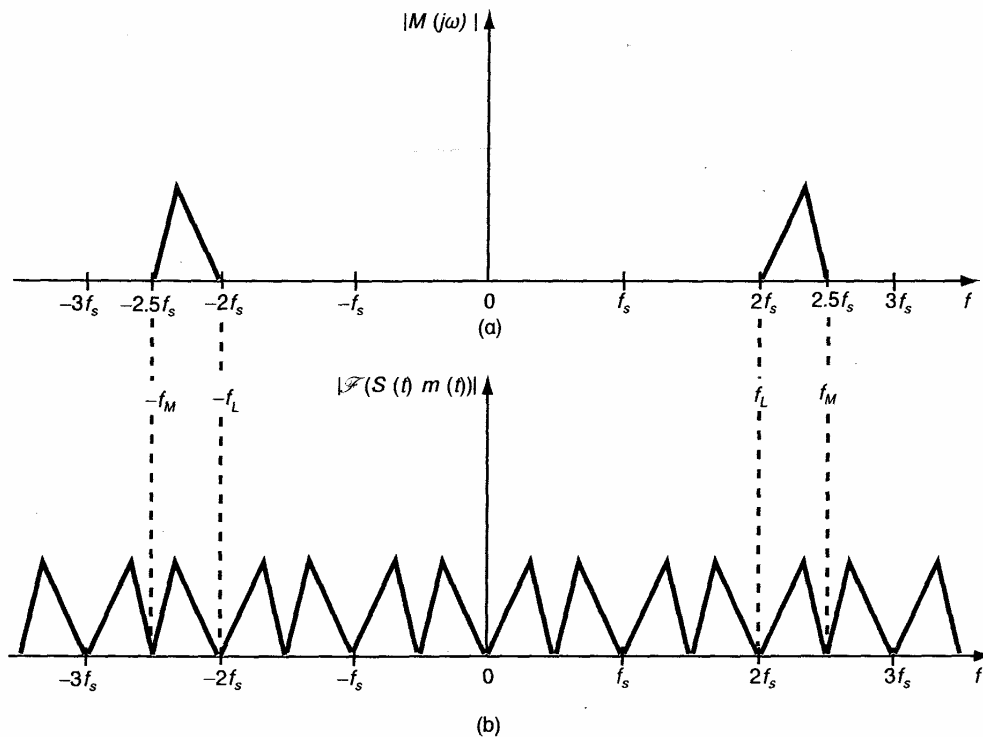
Σχήμα 2.3 (a) Μια ζώνη προστασίας εμφανίζεται όταν $f_s > 2f_M$. (b) Επικάλυψη φασμάτων όταν $f_s < 2f_M$

[2]

2.3 ΖΩΝΟΠΕΡΑΤΑ ΣΗΜΑΤΑ (BANDPASS SIGNALS)

Η συχνότητα δειγματοληψίας f_s δεν πρέπει να είναι μικρότερη από $f_s = 2f_M$ για ένα σήμα στο οποίο η φασματική συνιστώσα της υψηλότερης συχνότητας είναι f_M , παρά μόνο αν η φασματική συνιστώσα του $m(t)$ της χαμηλότερης συχνότητας είναι $f_L = 0$. Η συχνότητα δειγματοληψίας δεν είναι αναγκαίο να είναι μεγαλύτερη από $f_s = 2(f_M - f_L)$, εφόσον ισχύει η γενικότερη περίπτωση, όπου $f_L \neq 0$. Για παράδειγμα, στην περίπτωση που το φασματικό εύρος ενός σήματος κυμαίνεται από 10 έως 10.1 MHz, είναι δυνατόν να ανακτήσουμε το σήμα από σήματα που λαμβάνονται με συχνότητα $f_s = 2(10.1 - 10) = 0.2$ MHz.

Ας επιλέξουμε τώρα τη συχνότητα δειγματοληψίας $f_s=2*(f_M-f_L)$ και, για να αντιληφθούμε καλύτερα το θεώρημα της δειγματοληψίας για τέτοιου είδους ζωνοπερατά σήματα, ας υποθέσουμε καταρχήν ότι η συχνότητα f_L καταλήγει να ισοδυναμεί με ένα ακέραιο πολλαπλάσιο της f_s . Ας υποθέσουμε, δηλαδή, ότι ισχύει ο τύπος $f_L=n*f_s$, όπου το n ακέραιο. Μια τέτοια περίπτωση περιγράφεται στο παρακάτω σχήμα:



Σχήμα 2.4. (a) Το φάσμα ενός ζωνοπερατού σήματος. (b) Το φάσμα του ζωνοπερατού σήματος που έχει υποστεί δειγματοληψία.

Στο μέρος α παρατηρούμε το αμφίπλευρο φασματικό διάγραμμα ενός σήματος $m(t)$ με μετασχηματισμό Fourier $M(j\omega)$. Εδώ ο ακέραιος n έχει οριστεί να είναι 2. Η f_L , δηλαδή, συμπίπτει σε αυτή την περίπτωση με τη δεύτερη αρμονική της συχνότητας δειγματοληψίας, ενώ η συχνότητα δειγματοληψίας είναι ακριβώς $f_s=2*(f_M-f_L)$. Στο δεύτερο σκέλος μπορούμε να δούμε το φασματικό διάγραμμα του σήματος που έχει υποστεί δειγματοληψία $S(t)m(t)$. Το γινόμενο του $m(t)$ και του DC όρου του $S(t)$ αντιγράφει τη μορφή του φασματικού διαγράμματος του πρώτου σκέλους στο δεύτερο σκέλος και το αφήνει στο ίδιο εύρος συχνοτήτων από f_L έως f_M . Ως αποτέλεσμα του γινομένου του $m(t)$ και της φασματικής συνιστώσας του $S(t)$ συχνότητας f_s

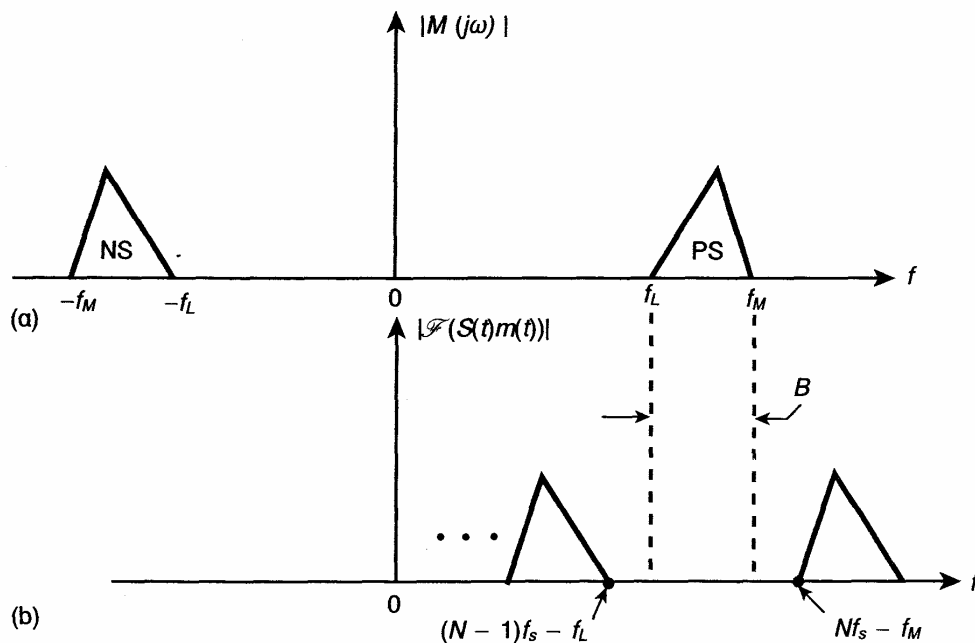
(ίσο με $1/T_S$), δημιουργείται στο δεύτερο σκέλος ένα φασματικό διάγραμμα, το οποίο προκύπτει από τη μετατόπιση του διαγράμματος του σκέλους α προς τα δεξιά και προς τα αριστερά κατά την ποσότητα f_S . Παρομοίως, οι υψηλότερες αρμονικές της f_S στο $S(t)$ έχουν ως αποτέλεσμα αντίστοιχες μετατοπίσεις τόσο δεξιά όσο και αριστερά του φασματικού διαγράμματος του σκέλους α . Οφείλουμε να σημειώσουμε σε αυτό το σημείο ότι στην περίπτωση που το σήμα, το οποίο έχει υποστεί δειγματοληψία $S(t)m(t)$, διέλθει μέσα από ένα ζωνοπερατό φίλτρο με αυθαίρετα οξείες αποκοπές και με ζώνη διέλευσης από f_L έως f_M , το σήμα $m(t)$ θα ανακτηθεί ακριβώς.

Στο παραπάνω σχήμα το φάσμα του $m(t)$ εκτείνεται στο πρώτο μισό του διαστήματος συχνοτήτων ανάμεσα στις αρμονικές της συχνότητας, κυμαίνεται δηλαδή από 2 έως $2.5 f_S$. Αποτέλεσμα των παραπάνω είναι η ανυπαρξία επικάλυψης στο φάσμα και η δυνατότητα ανάκτησης του σήματος. Επιπλέον, είναι εμφανές από το ίδιο σχήμα ότι σε περίπτωση που το φασματικό διάστημα του $m(t)$ εκτεινόταν στο δεύτερο ήμισυ του διαστήματος από 2.5 έως $3 f_S$, δε θα υπήρχε κατά παρόμοιο τρόπο επικάλυψη. Παράλληλα, αν το φάσμα του $m(t)$ δεν ήταν περιορισμένο ούτε στο πρώτο ούτε στο δεύτερο μισό του διαστήματος μεταξύ των αρμονικών της συχνότητας δειγματοληψίας, θα υπήρχε τότε επικάλυψη μεταξύ των φασματικών διαγραμμάτων και δεν θα καθίστατο δυνατή η ανάκτηση του σήματος. Επομένως, δεδομένου ότι είτε η f_M είτε η f_L είναι αρμονική της f_S , η ελάχιστη επιτρεπτή συχνότητα δειγματοληψίας είναι $f_S=2*(f_M-f_L)$.

Αν ούτε η f_M ούτε η f_L είναι αρμονικές της f_S , τότε είναι απαραίτητη μια γενικότερη ανάλυση. Στο πρώτο μέρος του ακόλουθου σχήματος έχουμε αναπαραγάγει το φασματικό διάγραμμα του προηγούμενου σχήματος. Το τμήμα αρνητικών συχνοτήτων ονομάζεται NS, ενώ το τμήμα αντίστοιχα των θετικών συχνοτήτων ονομάζεται PS. Για να απλοποιήσουμε κάπως την ανάλυση, θα λάβουμε χωριστά το PS και το NS καθώς και τον τρόπο, με τον οποίο μετατοπίζονται λόγω της δειγματοληψίας, και θα θεωρήσουμε αρχικά τους περιορισμούς, οι οποίοι είναι απαραίτητο να τεθούν, ώστε να μην προκληθεί επικάλυψη, για παράδειγμα, στο PS. Το γινόμενο του $m(t)$ και της DC συνιστώσας της κυματομορφής δειγματοληψίας αφήνει το PS αμετακίνητο. Αυτό είναι το σημείο του φάσματος, το οποίο εδώ επιλεκτικά θα εξάγουμε, προκειμένου να αναπαραγάγουμε το αρχικό σήμα. Εφόσον

ορίσουμε την ελάχιστη τιμή της f_s να είναι $f_s=2*(f_M-f_L)=2*B$, τότε τα μετατοπισμένα διαγράμματα PS δε θα επικαλύπτουν το PS. Εξάλλου, το NS θα παράγει τόσο προς τα αριστερά όσο και προς τα δεξιά μια σειρά μετατοπισμένων διαγραμμάτων. Οι αριστερές μετατοπίσεις δεν είναι δυνατό να επικαλύψουν το PS. Οι δεξιές, όμως, μετατοπίσεις του NS θα ήταν δυνατόν να έχουν ως αποτέλεσμα μια επικάλυψη και αυτές οι δεξιές μετατοπίσεις του NS αποτελούν τη μόνη δυνατή πηγή μιας τέτοιας επικάλυψης στο PS.

Στο ίδιο σχήμα εμφανίζονται, επίσης, τα μετατοπισμένα προς τα δεξιά διαγράμματα του NS εξαιτίας της (N-1)-οστής και της N-οστής αρμονικής της κυματομορφής δειγματοληψίας.



Σχήμα 2.5 (α) Το φάσμα του ζωνοπερατού σήματος. (β) Το φάσμα του NS μετατοπισμένου από την (N-1)-οστή και τη N-οστή αρμονική της κυματομορφής δειγματοληψίας.

Είναι φανερό ότι για να αποφύγουμε την επικάλυψη είναι αναγκαίο να ισχύει:

$$\text{και} \quad (N-1)f_s - f_L \leq f_L \quad (\text{Σχέση 2.2})$$

$$Nf_s - f_M \geq f_M \quad (\text{Σχέση 2.3})$$

έτσι ώστε με $B=f_M-f_L$ να έχουμε

$$(N-1)f_s \leq 2(f_M - B) \quad (\text{Σχέση 2.4})$$

και

$$Nf_s \geq 2f_M \quad (\text{Σχέση 2.5})$$

Εάν δώσουμε $k \equiv f_M/B$, οι εξισώσεις 2.4 και 2.5 γίνονται

$$f_s \leq 2B \left(\frac{k-1}{N-1} \right) \quad (\text{Σχέση 2.6})$$

και

$$f_s \geq 2B \left(\frac{k}{N} \right) \quad (\text{Σχέση 2.7})$$

Οι εξισώσεις 2.6 και 2.7 στηρίζουν τον περιορισμό που πρέπει να λάβουμε υπόψιν μας, προκειμένου να αποφευχθεί το ενδεχόμενο επικάλυψης στο PS. Ο ίδιος περιορισμός διασφαλίζει επίσης την αποφυγή επικάλυψης στο NS. Αυτό γίνεται φανερό τόσο από τη συμμετρία του αρχικού φάσματος όσο και από τη συμμετρία των μετατοπίσεων που απαιτούνται.

Στο σχήμα που ακολουθεί φαίνεται η γραφική παράσταση των εξισώσεων 2.6 και 2.7 για διάφορες τιμές του N. Οι περιοχές, στις οποίες ισχύει ο περιορισμός είναι οι διαγραμμισμένες. Αντιθέτως οι μη διαγραμμισμένες περιοχές είναι εκείνες, στις οποίες οι περιορισμοί δεν ικανοποιούνται και, επομένως, θα συμβεί επικάλυψη. Για να καταλάβουμε τη χρήση αυτών των διαγραμμάτων, μπορούμε να αναλογιστούμε την περίπτωση, κατά την οποία ένα σήμα στη βασική ζώνη έχει φάσμα που εκτείνεται από $f_L=2.5\text{KHz}$ έως $f_M=3.5\text{KHz}$. Επίσης, εδώ ισχύει ότι $B=1\text{KHz}$ και $k=f_M/B=3.5$. Κατόπιν τούτου, στο ακόλουθο έχουμε φέρει μια εστιγμένη κατακόρυφη γραμμή στο $k=3.5$. Γίνεται φανερό, λοιπόν, από το σχήμα ότι για αυτή την τιμή του k, η επιλογή μιας συχνότητας δειγματοληψίας $f_s=2*B=2\text{KHz}$ μας οδηγεί σε σημείο της περιοχής επικάλυψης.

Καθώς αυξάνει η f_s μπορούμε να παρατηρήσουμε την ύπαρξη ενός μικρού διαστήματος της, το οποίο αντιστοιχεί σε $N=3$, όπου δεν υπάρχει επικάλυψη. Αν προχωρήσουμε σε μεγαλύτερη αύξηση της f_s οδηγούμαστε και πάλι σε περιοχή επικάλυψης, ενώ η ακόμα μεγαλύτερη αύξηση της έχει ως αποτέλεσμα ένα διάστημα μη επικάλυψης που αντιστοιχεί σε $N=2$ (από $f_s=3.5B$ σε $f_s=5B$). Αυξάνοντας ακόμα περισσότερο την f_s φτάνουμε και πάλι σε περιοχή επικάλυψης, ενώ, όταν η $f_s=7B$ και $N=1$, μπαίνουμε ξανά σε περιοχή μη επικάλυψης. Αν, τέλος, ισχύει ο τύπος $f_s^3=7B$ δεν εισερχόμαστε και πάλι σε περιοχή επικάλυψης. Στη συγκεκριμένη περιοχή ισχύει η εξίσωση $f_s^3=2*f_M$, γεγονός που μας οδηγεί στην υπόθεση πως πρόκειται μάλλον για ένα χαμηλοπερατό παρά για ένα ζωνοπερατό σήμα. [2]

2.4 ΔΙΑΚΡΙΤΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER

Υπάρχουν πολλές περιπτώσεις στις οποίες η μόνη διαθέσιμη πληροφορία που μπορούμε να έχουμε για ένα σήμα είναι ένα σύνολο από τιμές N δειγμάτων, τα οποία τα έχουμε λάβει σε κανονικά χωρισμένα διαστήματα T_s μιας χρονικής περιόδου T_0 . Με βάση αυτά τα δεδομένα δειγματοληψίας θα ήταν συχνά επιθυμητό να καταλήξουμε σε κάποια λογική προσέγγιση του φασματικού περιεχομένου του σήματος. Είμαστε όντως σε θέση να εκτιμήσουμε το φασματικό περιεχόμενο του σήματος, μόνο όταν η περίοδος δειγματοληψίας και ο αριθμός των δειγμάτων είναι τόσο επαρκή, ώστε να μπορούμε να είμαστε κάπως βέβαιοι για το γεγονός πως αυτό που έχει παρατηρηθεί από το σήμα αντιπροσωπεύει γενικά το σήμα.

Υποθέτουμε πως το σήμα είναι περιοδικό με περίοδο T_0 και, επιπλέον, πως ο ρυθμός δειγματοληψίας είναι αρκετά επαρκής, προκειμένου να κατασταθεί δυνατή η ικανοποίηση του κριτηρίου Nyquist. Στο σχήμα που παρατίθεται παρακάτω βλέπουμε ένα αντιπροσωπευτικό σύνολο τιμών δειγμάτων. Μάλιστα, για να απλοποιήσουμε το παράδειγμά μας, έχουμε λάβει περιττό αριθμό δειγμάτων, ώστε να είναι σε θέση να τοποθετηθούν με συμμετρία στο διάστημα της περιόδου T_0 και συμμετρικά επίσης ως προς την αρχή των αξόνων. Οι τιμές των δειγμάτων βρίσκονται στα $\pm T_s/2$, $\pm 3T_s/2$ κ.ο.κ.

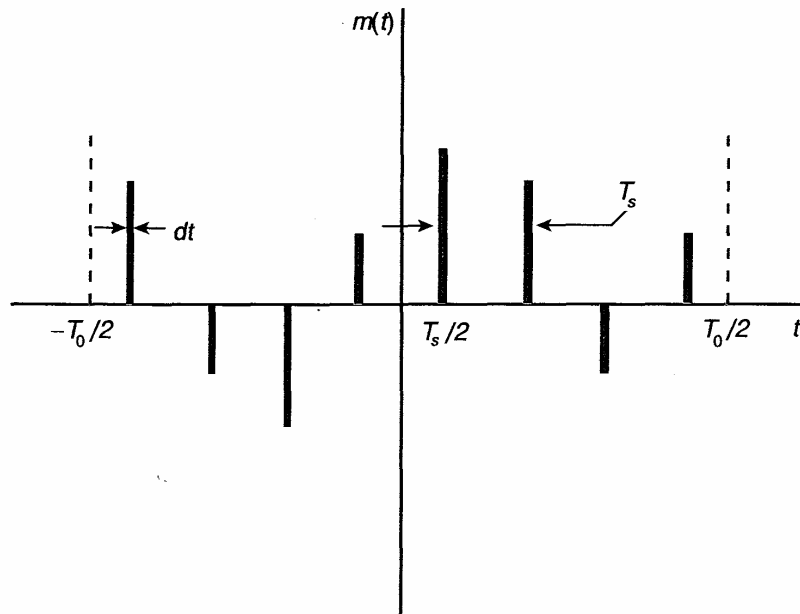
Εφόσον υπάρχουν N δείγματα, τότε τα δείγματα που απέχουν περισσότερο από την αρχή του συστήματος συντεταγμένων τοποθετούνται στα $\pm[(N-1)/2] \cdot T_s$.

Στην περίπτωση που η κυματομορφή, η οποία θα υποστεί τη δειγματοληψία είναι $m(t)$, η κυματομορφή που διαθέτουμε μετά τη δειγματοληψία είναι η $S(t) \cdot m(t)$, όπου η συνάρτηση $S(t)$ αναλογεί στη συνάρτηση δειγματοληψίας. Δηλαδή, κατά τη διάρκεια της δειγματοληψίας dt που εμφανίζεται στο ακόλουθο σχήμα ισχύει ότι $S(t)=1$, ενώ σε άλλη περίπτωση μπορεί να ισχύσει ότι $S(t)=0$. Με βάση τα σχήματα 2.2 και 2.3α παρατηρείται ταύτιση στη μορφή του φάσματος του $m(t)$ και τμήματος του φάσματος του $S(t) \cdot m(t)$ έως την f_M . Λαμβάνοντας υπόψιν αυτά τα δεδομένα μπορούμε, όταν θέλουμε να βρούμε το φάσμα του $m(t)$, να αποτιμήσουμε αντί για αυτό το φάσμα του $S(t) \cdot m(t)$. Τα φασματικά πλάτη του

$$M_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} m(t) e^{-j2\pi nt/T_0} dt \quad (\text{Σχέση 2.8})$$

Εάν υπάρχουν N δείγματα στο σύνολο, τότε οι χρόνοι δειγματοληψίας είναι οι ακόλουθοι:

$$-\left(\frac{N-1}{2}\right)T_s, -\left(\frac{N-1}{2}\right)T_s + T_s, -\left(\frac{N-1}{2}\right)T_s + 2T_s, \dots, \left(\frac{N-1}{2}\right)T_s + \dots$$



Σχήμα 2.6 Ένα πιθανό σύνολο τιμών δειγμάτων μιας κυματομορφής $m(t)$, που λαμβάνονται κάθε T_s σε ένα χρονικό διάστημα T_0

Εφόσον χρησιμοποιήσουμε αυτές τις τιμές στην ποσότητα ολοκλήρωσης, διαπιστώνουμε ότι η παραπάνω εξίσωση διαμορφώνεται ως εξής:

$$M_n = \frac{dt}{T_0} \sum_{k=-(N-1)/2}^{k=(N-1)/2} m(kT_s) e^{-j2\pi nkT_s/T_0} \quad (\text{Σχέση 2.9})$$

Θα μπορούσαμε να χρησιμοποιήσουμε την παραπάνω εξίσωση λαμβάνοντας την ως μια καθαρά μαθηματική άσκηση, για να υπολογίσουμε τις φασματικές συνιστώσες v_n για οποιαδήποτε τιμή του n . Αλλά, επειδή οφείλουμε να είμαστε συνεπείς με τις υποθέσεις μας, η μεγαλύτερη επιτρεπτή τιμή του n ορίζεται αναγκαστικά με βάση το κριτήριο Nyquist. Επομένως, η υψηλότερης συχνότητας συνιστώσα θα έχει περίοδο $2^* T_s$, έτσι ώστε $f_n(\max)=1/(2^* T_s)$. Η θεμελιώδης περίοδος είναι T_0 και, κατά συνέπεια, η θεμελιώδης συχνότητα είναι $f_0=1/T_0$ και, εφόσον $T_0=N^* T_s$, έχουμε:

$$f_n(\max) = \frac{N}{2} \frac{1}{T_0} = \frac{N}{2} f_0$$

(Σχέση 2.10)

Λαμβάνοντας, λοιπόν, υπόψιν μας όλα τα δεδομένα που προέκυψαν παραπάνω, μπορούμε να καταλήξουμε εύκολα στο συμπέρασμα πως η υψηλότερη τιμή του n είναι $n=N/2$. [2]

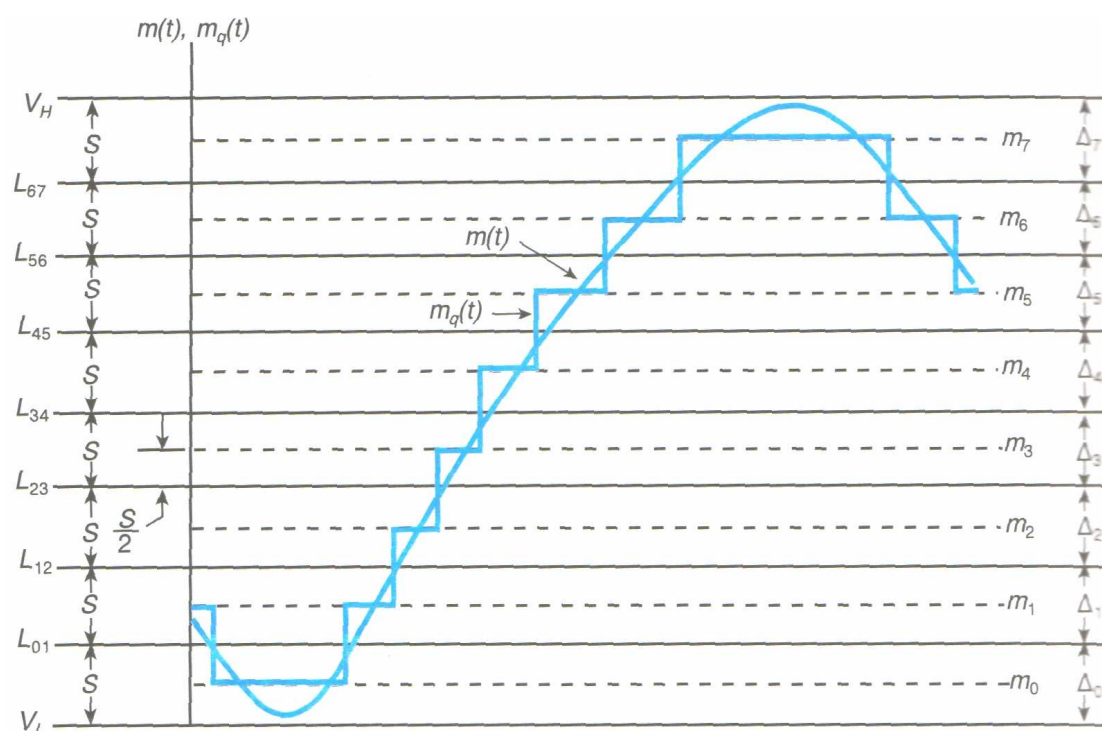
2.5 ΚΒΑΝΤΙΣΜΟΣ ΣΗΜΑΤΩΝ

Το σύστημα που έχουμε περιγράψει για τη δυνατότητα επικοινωνίας με τη χρήση καναλιών μεγάλου μήκους περιορίζεται στο γεγονός ότι κατά την εισαγωγή ενός θορύβου σε οποιοδήποτε σημείο κατά μήκος του καναλιού ενοποιούμε με αυτόν. Στη συνέχεια θα περιγραφεί ο τρόπος τροποποίησης της κατάστασης, στην περίπτωση κατά την οποία ένα σήμα υπόκειται στη διαδικασία του κβαντισμού. Κατά τον κβαντισμό ενός σήματος $m(t)$, δημιουργείται ένα νέο σήμα $m_q(t)$, το οποίο αποτελεί προσέγγιση του αρχικού σήματος $m(t)$. Παρά το γεγονός αυτό, η υψηλή αξία ενός κβαντισμένου σήματος $m_q(t)$ οφείλεται στο ότι είναι κατά μεγάλο ποσοστό δυνατό να διαχωριστεί από τον προσθετικό θόρυβο.

Στο Σχήμα 2.7 παρουσιάζεται η διαδικασία με την οποία πραγματοποιείται ο κβαντισμός. Εδώ θεωρούμε ένα σήμα $m(t)$ του οποίου η διαδρομή, δηλαδή η διακύμανση, περιορίζεται στο διάστημα από V_L έως V_H . Έχει διαιρεθεί αυτό το συνολικό διάστημα σε M ίσα διαστήματα, το μέγεθος των οποίων είναι ίσο με S . Σύμφωνα με αυτά, το S , το οποίο ονομάζεται *μέγεθος βήματος*, είναι $S=(V_L-V_H)/M$. Στο σχήμα αυτό, παριστάνεται μία συγκεκριμένη περίπτωση, στην οποία το $M=8$. Στο κέντρο αυτών των βημάτων τοποθετούνται οι *στάθμες κβαντισμού* $m_0, m_1, m_2, m_3, m_4, m_5, m_6$ και m_7 . Το κβαντισμένο σήμα $m_q(t)$ δημιουργείται με την ακόλουθη διαδικασία: όταν το $m(t)$ βρίσκεται στο διάστημα Δ_0 , το σήμα $m_q(t)$ διατηρεί τη σταθερή στάθμη m_0 , ενώ όταν το $m(t)$ βρίσκεται στο διάστημα Δ_1 , το κβαντισμένο σήμα διατηρεί τη σταθερή στάθμη m_1 κλπ. Με τον τρόπο αυτό, το σήμα $m_q(t)$ σε οποιαδήποτε χρονική στιγμή θα βρίσκεται σε διαφορετική κάθε φορά στάθμη

από την m_0 έως την m_7 . Η μετάβαση του $m_q(t)$ από $m_q(t)=m_0$ σε $m_q(t)=m_1$ πραγματοποιείται απότομα στην περίπτωση διέλευσης του $m(t)$ από τη στάθμη μετάβασης L_{01} , η οποία βρίσκεται μεταξύ των m_0 και m_1 , στο μέσον κλπ. Ένας διαφορετικός τρόπος παρουσίασης της διαδικασίας αυτής είναι να πούμε ότι σε κάθε χρονική στιγμή το $m_q(t)$ παίρνει την τιμή της εγγύτερης στο $m(t)$ στάθμης κβαντισμού.

Έτσι, κατά αυτόν τον τρόπο, το σήμα $m_q(t)$ είτε παραμένει σταθερό με το πέρασμα του χρόνου είτε πραγματοποιεί ένα “quantum” άλμα μεγέθους βήματος S . Σε αυτό το σημείο αξίζει να σημειωθεί η διάταξη των σταθμών κβαντισμού στο διάστημα από V_L έως V_H . Οι στάθμες αυτές διαχωρίζονται μεταξύ τους κατά ποσότητα S , αλλά η απόσταση καθενός από τα άκρα V_L και V_H από την κοντινότερή τους στάθμη κβαντισμού είναι μόλις $S/2$. Επιπλέον, σε κάθε χρονική στιγμή το μέτρο του σφάλματος κβαντισμού $m(t)-m_q(t)$ είναι μικρότερο ή ίσο του $S/2$.



Σχήμα 2.7 Η λειτουργία του κβαντισμού.

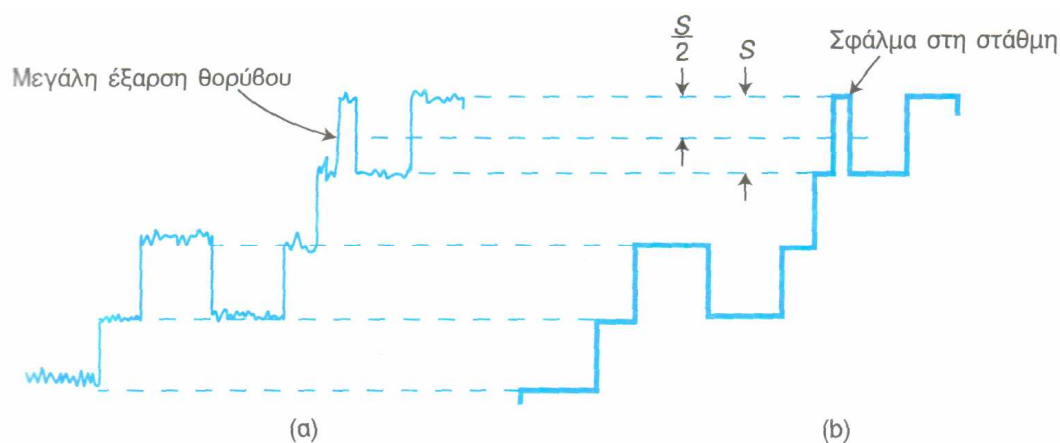
Συμπεραίνουμε, λοιπόν, ότι το κβαντισμένο σήμα αποτελεί μία προσέγγιση του αρχικού σήματος. Η ποιότητα αυτής της προσέγγισης είναι δυνατό να βελτιωθεί με τη μείωση του μεγέθους των βημάτων, με

αποτέλεσμα, βέβαια, την αύξηση του αριθμού των επιτρεπτών σταθμών. Τελικό αποτέλεσμα αυτής της χρησιμοποίησης αρκετών μικρών βημάτων αποτελεί η ανικανότητα του ανθρώπινου ματιού ή αυτιού να διαχωρίσει το αρχικό από το κβαντισμένο σήμα. Για να έχει κάποιος μία ιδέα του αριθμού των σταθμών που είναι απαραίτητες για τη λειτουργία ενός εφαρμόσιμου συστήματος, σημειώνεται ότι είναι δυνατό να χρησιμοποιηθούν 256 στάθμες για τη λήψη της ποιότητας της εμπορικής έγχρωμης τηλεόρασης, ενώ οι 64 στάθμες αποδίδουν αρκετά καλά σε μία έγχρωμη τηλεόραση. Επίσης, βρίσκεται ότι αυτά τα αποτελέσματα ισχύουν κατά τον κβαντισμό φωνής.

Στη συνέχεια, μπορούμε να θεωρήσουμε ότι το κβαντισμένο σήμα μας έχει φτάσει σε ένα σχετικά εξασθενημένο και κάπως αλλοιωμένο λόγω του θορύβου επαναλήπτη. Αυτή τη στιγμή, ο επαναλήπτης αυτός συγκροτείται από έναν κβαντιστή και έναν ενισχυτή. Μάλιστα, υπάρχει θόρυβος που υπερτίθεται στις κβαντισμένες στάθμες του $m_q(t)$. Ας υποθέσουμε, όμως, τώρα ότι ο επαναλήπτης έχει τοποθετηθεί σε ένα σημείο του καναλιού επικοινωνιών, στο οποίο η στιγμιαία τάση του θορύβου είναι σχεδόν σε όλες τις περιπτώσεις μικρότερη της απόστασης ανάμεσα στις κβαντισμένες στάθμες. Σε αυτήν την περίπτωση, η έξοδος του κβαντιστή θα σχηματίζεται από μία αλληλουχία σταθμών, οι οποίες αντιγράφουν το αρχικό σήμα που υπέστη τη διαδικασία του κβαντισμού *με το θόρυβο να έχει αφαιρεθεί*. Σε ελάχιστες περιπτώσεις, ο θόρυβος έχει ως τελικό αποτέλεσμα κάποιο σφάλμα στη στάθμη κβαντισμού. Στο πρώτο τμήμα του σχήματος 8 παριστάνεται ένα κβαντισμένο σήμα με θόρυβο. Οι διακεκομμένες γραμμές του σχήματος, οι οποίες είναι διαχωρισμένες κατά την ποσότητα S , τονίζουν τις στάθμες που επιτρέπονται στην έξοδο του κβαντιστή. Στο δεύτερο μέρος του ίδιου σχήματος παρουσιάζεται η έξοδος του κβαντιστή. Η έξοδος του κβαντιστή είναι εκείνη η στάθμη που βρίσκεται εγγύτερα στην είσοδο. Αυτό έχει ως συνέπεια τη μη ύπαρξη θορύβου στην έξοδο, αφού το στιγμιαίο πλάτος του θορύβου είναι μικρότερο από $S/2$. Στο σχήμα αυτό, επισημαίνεται μία στιγμή, κατά την οποία ο θόρυβος πραγματοποιεί μια υπέρβαση του $S/2$ και αυτό έχει ως αποτέλεσμα να υπάρχει ένα σφάλμα στη στάθμη. Η στατιστική φύση του θορύβου είναι τέτοια, ώστε να υπάρχει πάντα μια πεπερασμένη πιθανότητα το μέτρο του θορύβου να υπερβεί το $S/2$, ακόμα και στην περίπτωση που το μέσο μέτρο του θορύβου είναι κατά πολύ λιγότερο από $S/2$. Σε αυτό το

σημείο, αξίζει να σημειωθεί ότι σε καμία περίπτωση δεν υπάρχει δυνατότητα να καταστείλουμε πλήρως σφάλματα σε στάθμη, όπως για παράδειγμα αυτό που φαίνεται στο σχήμα 2.8.

Έχει αποδειχτεί ότι μέσω της διαδικασίας του κβαντισμού σήματος, είναι δυνατή μια σημαντική μείωση της επίδρασης του προσθετικού θορύβου. Με τη μείωση της απόστασης ανάμεσα στους επαναλήπτες, μειώνεται η υφιστάμενη εξασθένηση από το $m_q(t)$. Συνέπεια αυτής της εξασθένησης είναι η αποτελεσματική μείωση της σχετικής ισχύος του θορύβου και μείωση της πιθανότητας P_q ενός σφάλματος σε στάθμη. Η μείωση της πιθανότητας P_q είναι δυνατό να επιτευχθεί και με την αύξηση του μεγέθους βήματος S . Παρόλα αυτά, στην περίπτωση αύξησης του S , καταλήγουμε σε μία αυξημένη ασυμφωνία ανάμεσα στο πραγματικό σήμα $m(t)$ και το κβαντισμένο σήμα $m_q(t)$. Η διαφορά $m(t)-m_q(t)$ μπορεί να θεωρηθεί θόρυβος και ονομάζεται *θόρυβος κβαντισμού*. Συμπεραίνουμε, λοιπόν, ότι το σήμα στο δέκτη δεν αποτελεί ένα πιστό αντίγραφο του σήματος $m(t)$, το οποίο μεταδόθηκε. Η διαφορά ανάμεσα στα δύο σήματα είναι αποτέλεσμα των σφαλμάτων που προκλήθηκαν λόγω του προσθετικού θορύβου και του θορύβου κβαντισμού.



Σχήμα 2.8 (α) Ένα κβαντισμένο σήμα με προστιθέμενο θόρυβο. (β) Το σήμα μετά τον επανακβαντισμό. Καταγράφεται μια περίπτωση όπου η στάθμη θορύβου είναι τόσο μεγάλη που προκύπτει σφάλμα.

[2]

2.6 ΣΦΑΛΜΑ ΚΒΑΝΤΙΣΜΟΥ

Όπως έχει επισημανθεί, το κβαντισμένο σήμα διαφέρει κατά τυχαίο τρόπο από το αρχικό σήμα από το οποίο αυτό προέκυψε. Το σφάλμα αυτό ή η διαφορά μπορεί να θεωρηθεί θόρυβος, ο οποίος ανάγει την αιτία του στη διαδικασία κβαντισμού και ονομάζεται *σφάλμα κβαντισμού*. Υπολογίζουμε τώρα το μέσο τετραγωνικό σφάλμα κβαντισμού \bar{e}^2 , τύπος στον οποίο το e αντιστοιχεί στη διαφορά μεταξύ των τάσεων του κβαντισμένου σήματος και του αρχικού σήματος.

Στη συνέχεια προχωρούμε στη διαίρεση του συνολικού διαστήματος από κορυφή σε κορυφή του σήματος του μηνύματος $m(t)$ σε M ίσα διαστήματα τάσης, από τα οποία το καθένα έχει μέγεθος S volt. Στο μέσο κάθε διαστήματος τάσης βάζουμε, όπως φαίνεται στο παρακάτω σχήμα, μια στάθμη κβαντισμού m_1, m_2, \dots, m_M . Η στιγμιαία τιμή του σήματος του μηνύματος $m(t)$ σε κάποιο χρόνο t αναπαρίσταται στο σχήμα με τη διακεκομμένη γραμμή. Αφού το $m(t)$ είναι πιο κοντά στη στάθμη m_k , η έξοδος του κβαντιστή, η οποία αποτελεί την τάση που αντιστοιχεί σε αυτή την στάθμη θα είναι m_k . Το σφάλμα είναι $e = m(t) - m_k$.

Υποθέτουμε ότι η πιθανότητα να κείτεται το $m(t)$ στο διάστημα τάσης από $m - dm/2$ έως $m + dm/2$ είναι $f(m)dm$. Στην περίπτωση αυτή το μέσο τετραγωνικό σφάλμα κβαντισμού είναι:

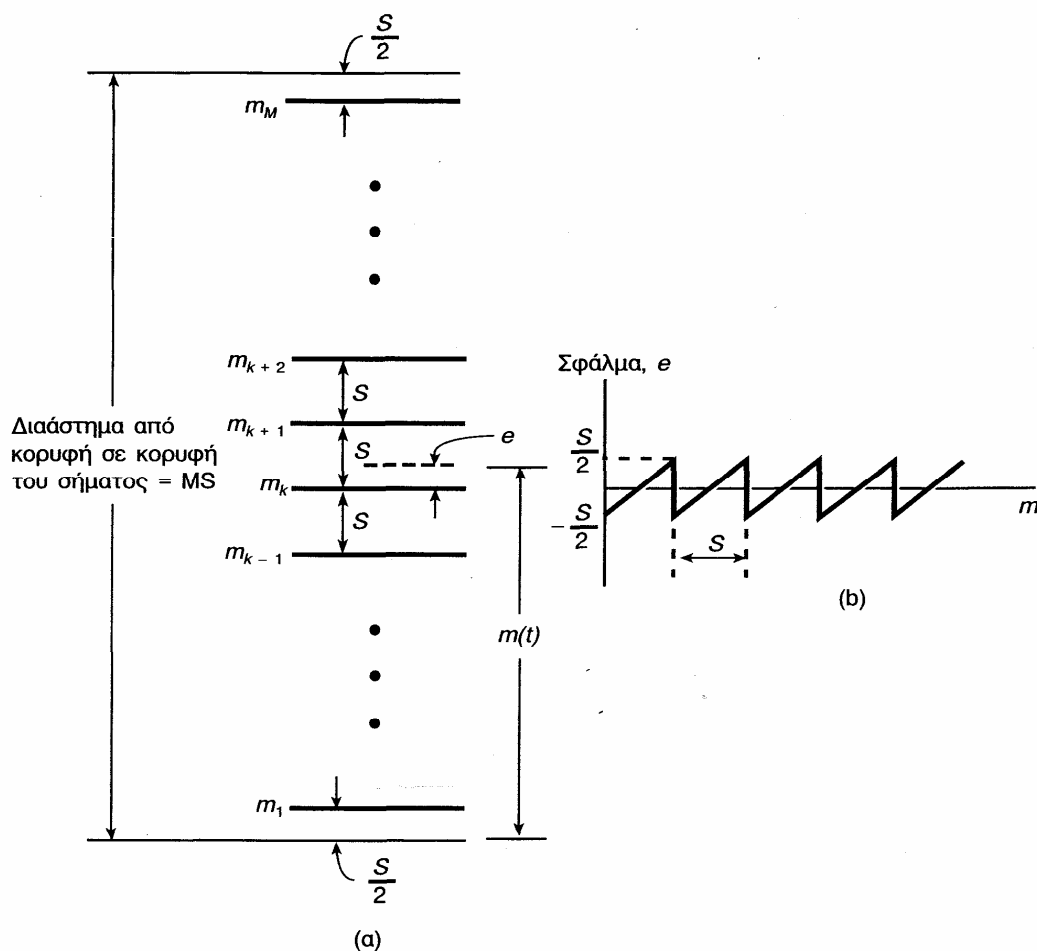
$$\bar{e}^2 = \int_{m_1 - S/2}^{m_1 + S/2} f(m)(m - m_1)^2 dm + \int_{m_2 - S/2}^{m_2 + S/2} f(m)(m - m_2)^2 dm + \dots \quad (\text{Σχέση 2.11})$$

Επιπλέον, η συνάρτηση πυκνότητας πιθανότητας $f(m)$ του σήματος του μηνύματος $m(t)$ είναι σχεδόν βέβαιο ότι συνήθως δε θα είναι σταθερή. Οφείλουμε, όμως, να αναλογιστούμε πως ο αριθμός M του κβαντισμού είναι μεγάλος, ώστε το μέγεθος του βήματος S να είναι μικρό σε σύγκριση με το διάστημα από κορυφή σε κορυφή του σήματος του μηνύματος. Είναι, λοιπόν, λογικό να υποθέσουμε ότι σε αυτή την περίπτωση η $f(m)$ είναι σταθερή μέσα σε κάθε διάστημα κβαντισμού. Έτσι, στον πρώτο όρο της παραπάνω

εξίσωσης ορίζουμε μια σταθερά, την $f(m)=f^{(1)}$. Στο δεύτερο όρο της ίδιας εξίσωσης ορίζουμε την $f(m)=f^{(2)}$ κ.ο.κ. Είμαστε πλέον σε θέση να βγάλουμε τις $f^{(1)}$, $f^{(2)}$, κ.ο.κ. μέσα από το ολοκλήρωμα. Αντικαθιστούμε στη συνέχεια

$$\bar{e}^2 = (f^{(1)} + f^{(2)} + \dots) \int_{-S/2}^{S/2} x^2 dx = (f^{(1)} + f^{(2)} + \dots) \frac{S^3}{12} \quad (\text{Σχέση 2.12α})$$

$$= (f^{(1)}S + f^{(2)}S + \dots) \frac{S^2}{12} \quad (\text{Σχέση 2.12β})$$



Σχήμα 2.9 (α) Ένα διάστημα τάσης, στο οποίο ένα σήμα $m(t)$ εκτυλίσσεται, διαιρείται σε M διαστήματα κβαντισμού, καθένα μεγέθους S . Οι στάθμες κβαντισμού τοποθετούνται στο κέντρο του διαστήματος. (β) Το σφάλμα τάσης $e(t)$ σαν συνάρτηση της στιγμιαίας τιμής του σήματος $m(t)$.

Η πιθανότητα να βρίσκεται η τάση του σήματος $m(t)$ στο πρώτο διάστημα κβαντισμού αναπαρίσταται με τον τύπο $f^{(1)}S$. Αντίστοιχα, η πιθανότητα να βρίσκεται η ίδια τάση στο δεύτερο διάστημα κβαντισμού αναπαρίσταται με $f^{(2)}S$ και παρομοίως αναπαριστούμε και τις υπόλοιπες πιθανότητες. Συνεπώς, η συνολική τιμή του αθροίσματος στην παρένθεση της Σχέσης 2.12β ισούται με 1. Επομένως, το μέσο τετραγωνικό σφάλμα κβαντισμού προκύπτει από τον τύπο:

$$\overline{e^2} = \frac{S^2}{12}$$

[2]

2.7 ΨΗΦΙΟΠΟΙΗΣΗ

Όταν το ηχητικό κύμα μετατρέπεται στο μικρόφωνο σε ηλεκτρικό σήμα, τότε το σήμα έχει αναλογική μορφή. Η διαδικασία της μετατροπής του αναλογικού σήματος σε ψηφιακό ονομάζεται ψηφιοποίηση. Η ψηφιοποίηση είναι υποχρεωτική κάθε φορά που θέλουμε να επεξεργαστούμε ήχο ψηφιακής μορφής. Η ψηφιοποίηση αποτελείται από τη χρονική δειγματοληψία, δηλαδή τη μεταβολή της συνεχούς ροής του χρόνου σε διακριτές χρονικές τιμές και την κβάντιση πλάτους, δηλαδή τη μετατροπή των συνεχών μεταβολών της τάσης σε διακριτές αριθμητικές τιμές. Με την ψηφιοποίηση μπορούμε να αναπαραστήσουμε με ακρίβεια τον ήχο με μια ακολουθία αριθμών, γεγονός που μεταβάλλει όλα τα δεδομένα κατά την εγγραφή, επεξεργασία, μετάδοση και αναπαραγωγή του ήχου. [11]

2.8 ΜΕΤΑΤΡΟΠΕΙΣ ΑΠΟ ΑΝΑΛΟΓΙΚΟ ΣΕ ΨΗΦΙΑΚΟ ΣΗΜΑ (ANALOG TO DIGITAL CONVERTERS A/D)

Ο μετατροπέας από αναλογικό σε ψηφιακό σήμα είναι ιδιαίτερα σημαντικός για τα περισσότερα συστήματα ψηφιακής επεξεργασίας δεδομένων σήμερα. Έχουν δημιουργηθεί διάφοροι τρόποι που να επιτελούν το σκοπό αυτό, αλλά κάθε περίπτωση έχει άμεση εξάρτηση από το ρυθμό δειγματοληψίας και τη διακριτική ικανότητα. Στη συνέχεια αναλύονται 5 γνωστοί μέθοδοι μετατροπής:

Όπου κρίνεται απαραίτητη η ύπαρξη υψηλής ακρίβειας, χρησιμοποιείται η μέθοδος της διπλής ολοκλήρωσης με αρκετά μεγάλη διακριτική ικανότητα (resolution) 10-14 bits αλλά με πολύ χαμηλούς ρυθμούς δειγματοληψίας (sampling rate) 2-50sps.

Τα τελευταία χρόνια έγινε, επίσης, γνωστή η μέθοδος Σίγμα-Δέλτα που χρησιμοποιείται κυρίως για ρυθμούς δειγματοληψίας μικρότερους από 100Ksps και ιδιαίτερα στις περιπτώσεις που απαιτείται μεγάλη διακριτική ικανότητα 14-24 bits.

Επιπλέον, υπάρχει η μέθοδος των διαδοχικών προσεγγίσεων (successive approximation) που χρησιμοποιείται ευρέως τα 20 τελευταία χρόνια. Με τη βοήθεια της μεθόδου αυτής δημιουργείται μια μεγάλη ομάδα ADCs με διακριτική ικανότητα από 8-16 bits. Οι μετατροπείς αυτοί παρουσιάζουν ρυθμούς δειγματοληψίας αντιστρόφως ανάλογους με την διακριτική τους ικανότητα και αυτό γιατί με βάση αυτή τη μέθοδο απαιτείται ένας παλμός ρολογιού, για να παραχθεί κάθε bit του αποτελέσματος εξόδου. Αυτό έχει ως αποτέλεσμα τον περιορισμό της μέγιστης πρακτικής ταχύτητας δειγματοληψίας τους.

Ακόμη, η μέθοδος των παράλληλων συγκριτών (flash) παρέχει τη δυνατότητα για ρυθμούς δειγματοληψίας υψηλότερους από 15Msps. Το μειονέκτημα αυτής της μεθόδου είναι πως απαιτούνται $2^n - 1$ συγκριτές για το σχηματισμό n bit. Αυτό συνεπάγεται και τη χρήση 255 συγκριτών για το σχηματισμό ενός ADC 8 bit.

Με βάση, εξάλλου, τη μέθοδο της ημιπαράλληλης τεχνικής (semiflash ή subbranging) μπορούμε να έχουμε ως αποτέλεσμα τη συνύπαρξη της πολυπλοκότητας του κυκλώματος και των υψηλών ρυθμών δειγματοληψίας.

Τέλος, υπάρχουν και οι μετατροπείς καναλιού (pipelined converters) που μας δίνουν έναν τέλειο συνδυασμό από υψηλές ταχύτητες δειγματοληψίας, σχετικά μικρή πολυπλοκότητα και μέτριο κόστος. Αυτοί οι μετατροπείς είναι ιδανικοί για την πραγματοποίηση συνεχούς δειγματοληψίας, εκεί που η εσωτερική καθυστέρηση του καναλιού δεν επιφέρει αλλαγές στην όλη λειτουργικότητα του συστήματος. [1]

2.9 ΜΕΤΑΤΡΟΠΕΙΣ ΑΠΟ ΨΗΦΙΑΚΟ ΣΕ ΑΝΑΛΟΓΙΚΟ ΣΗΜΑ (DIGITAL TO ANALOG CONVERTERS D/A)

Η μετατροπή των δεδομένων από ψηφιακή σε αναλογική μορφή είναι δυνατόν να επιτευχθεί με διάφορους τρόπους, από τους οποίους θα αναλύσουμε δυο πολύ γνωστούς στη συνέχεια.

Η πρώτη μέθοδος ονομάζεται μετατροπέας κλιμακωτών αντιστάσεων (Resistor string ADC). Στην περίπτωση αυτή η τάση αναφοράς διαιρείται σε 2^n-1 μέρη από τα οποία το καθένα έχει μέγεθος ίσο με 1 LSB. Ένα σύνολο διακοπών, όπως φαίνεται στο σχήμα παρακάτω, επιλέγει την τάση εξόδου, από κλιμακωτές αντιστάσεις ανάλογα με τον ψηφιακό κώδικα εισόδου. Για να μην προκύψουν σφάλματα ως αποτέλεσμα των ρευμάτων φορτίου, είναι ανάγκη να υπάρχει μια βαθμίδα απομόνωσης (buffer) στην έξοδο του DAC που αρκετές φορές εμπεριέχεται στο ίδιο ολοκληρωμένο. Το πλεονέκτημα αυτής της τεχνικής είναι ότι η συνάρτηση μεταφοράς είναι μονίμως μονοτονική και έτσι έχουμε ως αποτέλεσμα την απλότητα του εσωτερικού σχεδίου του DAC. Αντιθέτως, το μειονέκτημά της είναι πως χρειάζονται 2^n-1 ισοσταθμισμένες αντιστάσεις που περιορίζουν την τελική διακριτική ικανότητα και ως εκ τούτου είναι απαραίτητος ένας ενισχυτής που να περιορίζει την τελική ταχύτητα.

Η δεύτερη μέθοδος ονομάζεται μετατροπέας οδήγησης ρεύματος (Current Steering DAC). Αυτή η τεχνική στηρίζεται σε διακοπτόμενες πηγές ρεύματος. Αποτελείται από δυο εξόδους ρεύματος, από τις οποίες η μια συμπληρώνει το ρεύμα της άλλης. Σε κάθε περίπτωση, όμως, το συνολικό ρεύμα εξόδου παραμένει σταθερό. Η ψηφιακή ελέγχει σε αυτή τη μέθοδο μια σειρά από διακόπτες, οι οποίοι οδηγούν το ρεύμα των πηγών σε καθεμιά από τις εξόδους. Τα βασικά πλεονεκτήματα των DAC οδήγησης ρεύματος, όπως

συμβαίνει και στους DAC κλιμακωτών αντιστάσεων, είναι η μονοτονικότητα και οι πολύ μεγαλύτερες ταχύτητες από διατάξεις με έξοδο τάσης. [1]

2.10 ΥΠΟΛΟΓΙΣΤΗΣ ΚΑΙ ΗΧΟΣ

Για να καταφέρουμε να έχουμε ήχο στον υπολογιστή μας, είναι υποχρεωτικό να τοποθετήσουμε σε αυτόν κάρτα ήχου. Οι κάρτες ήχου είναι ηλεκτρονικά κυκλώματα που τοποθετούνται πάνω στη μητρική κάρτα του υπολογιστή. Έχουν μέσα τους εγκατεστημένα έτοιμα προγράμματα μετατροπής του αναλογικού σήματος σε ψηφιακό και το αντίστροφο. Υπάρχουν πολλές κατηγορίες καρτών ήχου οι οποίες καλύπτουν τις ανάγκες των περισσότερων πολιτών τώρα πια, αφού είτε είναι κάποιος ερασιτέχνης είτε επαγγελματίας μπορεί εύκολα να βρει την κάρτα ήχου που ψάχνει. Είναι προφανές πως τα τελευταία χρόνια η χρήση της μουσικής μέσω του υπολογιστή παίρνει όλο και μεγαλύτερες διαστάσεις. Ο καθένας μπορεί να μπει στο διαδίκτυο και να κατεβάσει στον υπολογιστή του όποιο ήχο του αρέσει. Έτσι, καταλήγουμε στο συμπέρασμα πως, ενώ πριν από μερικά χρόνια είχαμε ελάχιστους ήχους, τώρα πια έχουμε ποικίλους πολυφωνικούς ήχους. Και βέβαια με τη βοήθεια του κατάλληλου λογισμικού η ποιότητα του ήχου ανεβαίνει όλο και περισσότερο. [11]

2.11 ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΤΥΠΩΝ ΗΧΟΥ

Τα πρότυπα ήχου καθορίζουν τον τρόπο, με τον οποίο αποθηκεύεται και μεταδίδεται η μουσική σε ψηφιακή μορφή. Με βάση τα πρότυπα αυτά, επιτρέπεται η μεταφορά δεδομένων από υπολογιστή σε υπολογιστή. Η εξέλιξη της τεχνολογίας οδήγησε στην αύξηση των δυνατοτήτων συμπίεσης, απαιτώντας πλέον λιγότερα bits για την αναπαράσταση κάθε ηχητικής πληροφορίας και παρέχοντας υψηλότερους ρυθμούς μετάδοσης ήχου, μέσω του διαδικτύου. Όλοι γνωρίζουμε πως έχουμε σήμερα τη δυνατότητα τόσο να επικοινωνήσουμε τηλεφωνικά όσο και να ακούσουμε κάθε ραδιοφωνικό σταθμό ενταγμένο στο διαδίκτυο ή ακόμα και μεμονωμένα μουσικά κομμάτια. Ορισμένα από τα πιο γνωστά πρότυπα, θα αναλύσουμε στη συνέχεια. [11]

2.12 ΠΡΟΤΥΠΟ WAVE (WAV)

Στο πρότυπο Waveform Audio File Format (wav) ο ήχος που έχει ψηφιοποιηθεί αποθηκεύεται, όπως ακριβώς έχει προκύψει από την δειγματοληψία, σε μορφή, δηλαδή, παλμοκωδικής διαμόρφωσης PCM (Pulse Coded Modulation). Συνήθως χρησιμοποιούνται δείγματα 8 και 16 bit. Μερικές φορές, επίσης, χρησιμοποιούνται και 24 ή 32 bit. Ένα αρχείο ήχου του προτύπου αυτού αποτελείται από δυο τμήματα. Το πρώτο τμήμα λέγεται format chunk. Περιλαμβάνει την πληροφορία που θα μετατραπεί σε αρχείο ήχου τύπου wav, τον απαιτούμενο ρυθμό δειγματοληψίας και το μήκος του κάθε δείγματος σε bit. Το δεύτερο τμήμα που λέγεται data chunk περιλαμβάνει τα pcm δείγματα της κυματομορφής αλλά και άλλες πληροφορίες, όπως η αναπαραγωγή ενός τραγουδιού, όταν βρίσκεται σε αναμονή από μια λίστα τραγουδιών. Αξίζει να σημειωθεί πως υπάρχουν και συμπιεσμένες μορφές του προτύπου wav, με τις οποίες, όμως, δε θα ασχοληθούμε. Τέλος, πρέπει να αναφέρουμε πως το πρότυπο wav χρησιμοποιείται περισσότερο στην περίπτωση της απλής αποθήκευσης pcm ηχητικών κυματομορφών, απαιτώντας, όμως, μεγάλες ποσότητες ψηφιακής μνήμης. Το πρότυπο wav αναπτύχθηκε από τις εταιρίες Microsoft και IBM για τους υπολογιστές που χρησιμοποιούν Windows 3.1. [11]

2.13 ΠΡΟΤΥΠΟ MOTION PICTURE EXPERTS GROUP (MPEG)

Το πρότυπο MPEG αποτελεί το διεθνές πρότυπο multimedia και περιλαμβάνει μετάδοση ήχων και video κωδικοποιημένα σε διάφορους ρυθμούς. Ο μικρότερος ρυθμός μετάδοσης που υπάρχει σε αυτό το πρότυπο είναι τα 32kbps.

Το MPEG πρότυπο αποτελείται από 3 τύπους κωδικοποίησης: MPEG-Layer1, MPEG-Layer2 και MPEG-Layer3. Ο τελευταίος τύπος κωδικοποίησης είναι πιο γνωστός ως mp3. Θα πρέπει να θεωρείται αυτονόητο πως, όσο εξελίσσονται οι τύποι του προτύπου MPEG, έχουμε όλο και καλύτερη ποιότητα παραγόμενου ήχου σε ολοένα μικρότερους ρυθμούς μετάδοσης. Επίσης, όσο εξελίσσεται το πρότυπο αυτό, ο χρόνος κωδικοποίησης και η πολυπλοκότητα του κωδικοποιητή ελαττώνονται, ενώ,

αντίθετα, αυξάνεται η ποιότητα του ψηφιοποιημένου ήχου. Αυτό μπορούμε να το καταλάβουμε, αν σκεφτούμε πως ο τύπος κωδικοποίησης 2 προσφέρει καλύτερη ποιότητα από τον τύπο 1 και ο τύπος 3, αν και είναι ο πιο πολύπλοκος από τους τρεις, έχει εκπληκτική ποιότητα σε πολύ μικρούς ρυθμούς μετάδοσης.

Κατά την κωδικοποίηση ήχων τύπου MPEG ο κωδικοποιητής αποφασίζει από μόνος του ποιες από τις πληροφορίες αυτές είναι απαραίτητες στον άνθρωπο και απορρίπτει αυτές που κρίνει ο ίδιος ανωφελείς. Έτσι οποιοδήποτε σήμα είναι ασθενές δεν το αντιλαμβανόμαστε. Το πρότυπο MPEG, τέλος, έχει τη δυνατότητα ακόμα και να διαγράφει ηχογραφημένη πληροφορία. Ολοκληρώνοντας το κεφάλαιο αυτό, οφείλουμε να σημειώσουμε πως, σε αντίθεση με τις παραπάνω μορφές συμπίεσης, στις συμπίεσεις απλής μορφής η διαφορά συμπιεσμένου και αρχικού σήματος δε διακρίνεται. [11]

ΚΕΦΑΛΑΙΟ 3

ΣΥΜΠΙΕΣΗ ΣΗΜΑΤΩΝ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟ ΔΙΑΚΡΙΤΟ ΟΡΘΟΓΩΝΙΟ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟ GAUSS-HERMITE

3.1 ΕΙΣΑΓΩΓΗ

Στις μέρες μας η συμπίεση ψηφιακών σημάτων ήχου είναι εξαιρετικά αναγκαία τόσο για τη διευκόλυνση της αποθήκευσής τους λ.χ. σε CD-ROM ή σε σκληρούς δίσκους λογικής χωρητικότητας, όσο και για την πραγματοποίηση της ασύρματης και ενσύρματης μετάδοσής τους μέσω καναλιών που επιτρέπουν περιορισμένο εύρος ζώνης. Έτσι, για παράδειγμα, μία κοινή σύνδεση Internet μέσω μίας απλής τηλεφωνικής γραμμής και ενός modem προσφέρει ανώτατο ρυθμό μετάδοσης 56 Kbits/sec = 7 Kbytes/sec, σε αντίθεση με μία ασύρματη σύνδεση μέσω GSM/GPRS, η οποία προσφέρει ρυθμό μετάδοσης της τάξης των 40 Kbits/sec = 5 Kbytes/sec.

Από την άλλη πλευρά, το μέγεθος ενός ασυμπίεστου αρχείου μουσικής (.wav) είναι 44.1 KHz (ρυθμός δειγματοληψίας σε χιλιάδες δείγματα ανά δευτερόλεπτο)* 16 bits (κβάντιση του κάθε δείγματος σε $2^{16} = 65\ 536$ επίπεδα)*2 (stereo = 2 κανάλια) = 1.4 Mbits/sec. Αυτό σημαίνει ότι για ένα τραγούδι, το οποίο διαρκεί 4 λεπτά, χρειαζόμαστε $4 \times 60 \text{secs} \times 1.4 \text{Mbits/sec} = 336 \text{Mbits} = 42 \text{Mbytes}$. Έτσι, για να μεταδοθεί το παραπάνω ασυμπίεστο αρχείο μουσικής .wav μέσω Internet, απαιτούνται $42\ 000 \text{Kbytes} / (7 \text{Kbytes/sec}) = 1\text{h}40\text{min}$. Είναι κατάδηλο ότι ο προαναφερθέν χρόνος μετάδοσης ήχου είναι πολύ μεγάλος.

Τα τελευταία, λοιπόν, δώδεκα χρόνια, έχουν προταθεί διάφοροι αλγόριθμοι συμπίεσης σημάτων, από τους οποίους οι περισσότερο διαδεδομένοι και κανονικοποιημένοι, ώστε να αποτελούν διεθνείς σταθερές, είναι οι ακόλουθοι: JPEG (Joint Photographic Expert Group) που από το 1991 βοηθά στη συμπίεση ακίνητης εικόνας και MPEG (Moving Picture Expert Group) που από το 1992 έχει τη δυνατότητα να συμπίεσει ήχο. Πιο συγκεκριμένα, η πιο απλή συμπίεση MPEG-1 παρουσιάστηκε το 1992. Αντίθετα, οι πολυπλοκότερες αλλά και πιο βελτιωμένες εκδόσεις MPEG-2 και MPEG-4 εμφανίστηκαν μεταγενέστερα, το 1994 και το 1996 αντιστοίχως. Για τη συμπίεση του ήχου χρησιμοποιείται ο αλγόριθμος MPEG Layer 3 ή εν

συντομία MP3. Επιπροσθέτως, από το 1997, την χρονιά δηλαδή που χρησιμοποιήθηκαν για πρώτη φορά δωρεάν codecs MP3 στο Διαδίκτυο, υπήρξε εκρηκτική η αύξηση της χρήσης της συμπίεσης κατά MP3, όσον αφορά στην ανταλλαγή μουσικών κομματιών.

Το κοινό στοιχείο όλων των μεθόδων που έχουν προαναφερθεί είναι το γεγονός ότι η καρδιά του αλγορίθμου συμπίεσης είναι ο διακριτός μετασχηματισμός συνημιτόνου (Discrete Cosine Transform – DCT), για τον υπολογισμό του οποίου τα τελευταία χρόνια έχουν δημιουργηθεί εξειδικευμένα ολοκληρωμένα κυκλώματα. Οι αλγόριθμοι MP3 και JPEG χρησιμοποιούν το μετασχηματισμό DCT με $N = 18$ σημεία και δισδιάστατο DCT $8 \times 8 = 64$ σημείων αντίστοιχα. Οι αλγόριθμοι αυτοί έχουν τη δυνατότητα να συμπίεσουν ένα αρχείο περίπου 10 φορές, δίχως να υπάρχει αισθητή μείωση της ποιότητάς του. Έτσι, το τραγούδι, το οποίο σύμφωνα με το παραπάνω παράδειγμα διαρκεί τέσσερα λεπτά, συμπιέζεται με ρυθμό 128 Kbits/sec, δηλαδή 11 φορές συμπίεση, και αποθηκεύεται μνήμη της τάξεως των 4 Mbytes. Από την άλλη μεριά, ο χρόνος αποστολής του μέσω Internet είναι περίπου 9 λεπτά.

Μέσα από τις παρατηρήσεις και τα πειράματα που πραγματοποιήθηκαν τα τελευταία δέκα χρόνια χρησιμοποιώντας ως μεθόδους συμπίεσης εικόνας και ήχου JPEG και MP3, διαπιστώθηκε ότι το όριο συμπίεσης δίχως να γίνεται αισθητή η παραμόρφωση των 10 φορών οφείλεται βασικά στο μετασχηματισμό DCT, ο οποίος είναι ο ίδιος και στις δύο παραπάνω μεθόδους. Έτσι, ερευνώντας για ακόμη μεγαλύτερους λόγους συμπίεσης από το 1 : 10 και ιδιαίτερα ερευνώντας για μία περισσότερο ήπιας μορφής αύξηση της παραμόρφωσης του αντικειμένου, όσον αφορά στους υψηλούς λόγους συμπίεσης εικόνας, όπως για παράδειγμα 1 : 100, τέθηκε η πρόταση να αποσυρθεί ο μετασχηματισμός DCT και να χρησιμοποιείται πλέον ο μετασχηματισμός DWT (Discrete Wavelet Transform), ο οποίος όχι μόνο δε χρησιμοποιεί συναρτήσεις τύπου συνημιτόνου, αλλά και χρησιμοποιεί συναρτήσεις βάσης τύπου wavelets. Η χρησιμοποίηση τόσο μεγαλύτερων λόγων συμπίεσης σε εφαρμογές ήχου όσο και αλγορίθμων σχετικά απλών και όχι πολύπλοκων κατά τη διάρκεια υπολογισμών παραμένει εξαιρετικά αναγκαία.

Ο αλγόριθμος DOGHT είναι ικανός να χρησιμοποιηθεί στη θέση των παραπάνω αλγορίθμων DCT και DWT, υπερέχει στην πληθώρα των πρακτικών περιπτώσεων και των δύο, ενώ δίνεται σε απλή αναλυτική μορφή παρόμοια με αυτή του DCT. Επιπλέον, όσον αφορά στον αλγόριθμο DOGHT, ισχύουν ορισμένες μαθηματικές ιδιότητες, όπως η σχέση Parseval, οι οποίες είναι σχεδόν όμοιες με τις ιδιότητες του DCT, γεγονός το οποίο καθιστά ευκολότερη την εφαρμογή του. [7,8,9]

3.2 ΠΕΡΙΓΡΑΦΗ

Ο διακριτός ορθογώνιος μετασχηματισμός Gauss-Hermite (Discrete Orthogonal Gauss-Hermite Transform – DOGHT) ορίζεται από τις σχέσεις:

$$c_n = \sum_{i=0}^{N-1} w_i f\left(\frac{t_i}{T}\right) h_n(t_i) \quad (\text{Σχέση 3.1})$$

$$\tilde{f}\left(\frac{t_i}{T}\right) = \sum_{n=0}^{N-1} c_n h_n(t_i) \quad (\text{Σχέση 3.2})$$

όπου T είναι ένας συντελεστής κλιμάκωσης του χρόνου και

$$h_n(t) = \frac{1}{(2^n n! \sqrt{\pi})^{1/2}} \exp\left(-\frac{t^2}{2}\right) H_n(t) \quad (\text{Σχέση 3.3})$$

είναι οι κανονικοποιημένες συναρτήσεις Gauss-Hermite.

Επιπλέον,

$$\int_{-\infty}^{+\infty} h_m(t) h_n(t) dt = \delta_{mn} \quad (\text{Σχέση 3.4})$$

είναι η σχέση ορθογωνιότητας συνεχούς χρόνου των κανονικοποιημένων συναρτήσεων Gauss-Hermite τάξης n : h_n και H_n είναι τα κλασικά πολυώνυμα

Hermite τάξης n . Τα σημεία δειγματοληψίας: $t_0 < t_1 < t_2 < \dots < t_{N-1}$ είναι τα μηδενικά του πολυωνύμου Hermite τάξης N , τα οποία είναι γνωστά και από τη θεωρία προσεγγιστικής ολοκλήρωσης κατά Gauss. Τα βάρη w_i δίνονται από τη σχέση:

$$w_i = \frac{2}{[h_N(t_i)]^2} \quad (\text{Σχέση 3.5})$$

Οι κανονικοποιημένες συναρτήσεις Gauss-Hermite ικανοποιούν τις μεγάλης σημασίας σχέσεις διακριτής ορθογωνιότητας (ορθογωνιότητας διακριτού χρόνου):

$$\sum_{i=0}^{N-1} w_i h_m(t_i) h_n(t_i) = \delta_{mn} \quad (\text{Σχέση 3.6})$$

$$\sum_{n=0}^{N-1} w_n h_m(t_i) h_n(t_j) = \delta_{ij} \quad (\text{Σχέση 3.7})$$

Λόγω της εξίσωσης (3.6) οι συντελεστές του μετασχηματισμού c_n υπολογίζονται με βάση την εξίσωση (3.1). Επιπλέον, η εξίσωση (3.7) οδηγεί στη σχέση:

$$\tilde{f}\left(\frac{t_i}{T}\right) = f\left(\frac{t_i}{T}\right) \quad (\text{Σχέση 3.8})$$

η οποία είναι μία από τις βασικές και εξαιρετικής σημασίας ιδιότητες του μετασχηματισμού DOGHT (collocation property). Οι παραπάνω ιδιότητες είναι παρόμοιες με αυτές του διακριτού μετασχηματισμού συνημιτόνου (Discrete Cosine Transform - DCT) και αποτέλεσμα αυτών είναι η ύπαρξη του *ακριβούς αντίστροφου μετασχηματισμού DOGHT* στα σημεία t_i της δειγματοληψίας κατά Gauss-Hermite.

Για τη διευκόλυνση της σύγκρισης, αξίζει να αναφέρουμε ότι ο διακριτός μετασχηματισμός συνημιτόνου (DCT) ορίζεται από τις σχέσεις:

$$c_n = \frac{2\alpha}{N} \sum_{k=0}^{N-1} f(t_k) \cos\left[\frac{n\pi(k+1/2)}{N}\right] \quad (\text{Σχέση 3.9})$$

$$\tilde{f}(t_k) = \sum_{n=0}^{N-1} \alpha c_n \cos\left[\frac{n\pi(k+1/2)}{N}\right] \quad (\text{Σχέση 3.10})$$

$$\alpha = 1/\sqrt{2}, n=0$$

$$\alpha = 1, n>0$$

Η δειγματοληψία είναι ομοιόμορφη και δίνεται στο συμμετρικό διάστημα $[-t_{\max}, +t_{\max}]$ από τη σχέση:

$$t_k = -t_{\max} + k \Delta t, \quad \Delta t = \frac{2t_{\max}}{N-1}$$

Στα πλαίσια του μετασχηματισμού DCT ισχύει, επίσης, η ιδιότητα του ακριβούς αντίστροφου μετασχηματισμού $\tilde{f}(t_k)=f(t_k)$ στα σημεία t_k της ομοιόμορφης δειγματοληψίας. [7,8,9]

ΚΕΦΑΛΑΙΟ 4

MATLAB

4.1 ΤΟ MATLAB ΩΣ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Σε αυτό το κεφάλαιο, θα κάνουμε αρχικά μια ανάλυση, για να καταλάβουμε ποια είναι η γλώσσα προγραμματισμού MATLAB και ποιες δυνατότητες έχουμε μέσα από αυτό. Στη συνέχεια, θα γίνει μια λεπτομερής ανάλυση όλων των προγραμμάτων που χρησιμοποιήθηκαν και των αποτελεσμάτων τους. Τα ανύσματα και οι πίνακες είναι η «καρδιά» του MATLAB. Στο MATLAB, όπως και σε άλλες γλώσσες προγραμματισμού, είναι εύκολο να συνδυαστούν αριθμητικοί υπολογισμοί με γραφήματα και προγραμματισμό. Επίσης, έχει τη δυνατότητα να χρησιμοποιηθούν εργαλεία GUI (Graphic User Interface – Γραφικό Περιβάλλον χρήστη).

Έχει ιδιαίτερες ικανότητες σε αριθμητικούς υπολογισμούς χρησιμοποιώντας περισσότερα από 200 αξιόπιστα και με μεγάλη ακρίβεια μαθηματικά υποπρογράμματα, όπως το Simulink, το Signal processing και πολλά άλλα. Τα υποπρογράμματα αυτά δίνουν λύση σε πολλών και διαφόρων ειδών προβλήματα. Χαρακτηριστικά παραδείγματα τέτοιων προβλημάτων:

- άλγεβρα πινάκων
- διαφορικές εξισώσεις και πολυώνυμα
- μη γραμμικά συστήματα
- πολύπλοκες αριθμητικές πράξεις
- δημιουργία αλγορίθμων
- ανάλυση και οπτικοποίηση δεδομένων
- γραφικές παραστάσεις

Ένα μεγάλο πλεονέκτημα του MATLAB είναι ότι διατίθεται σε όλα τα λειτουργικά συστήματα (Dos, Windows 95/98/NT/2000/XP, Unix/Linux, Macintosh) και, μάλιστα, η σύνταξη των εντολών είναι η ίδια για κάθε σύστημα. Ακόμη, μπορεί να γίνει χρήση του MATLAB σε συνδυασμό με τις γλώσσες FORTRAN και C.

Πέρα από τις μεταβλητές που εισάγει ο χρήστης, υπάρχουν και *προκαθορισμένες μεταβλητές και σταθερές* με ειδική σημασία. Ορισμένες από αυτές είναι το αποτέλεσμα κάθε αριθμητικής πράξης, που συμβολίζεται με **ans**, και ο αριθμός π, που συμβολίζεται με **pi**.

4.2 ΜΕΤΑΒΛΗΤΕΣ ΣΤΟ MATLAB

Μέσα από το MATLAB, μπορούμε πολύ εύκολα να κάνουμε μαθηματικές πράξεις μεταξύ αριθμών. Αν θελήσουμε να κάνουμε πράξεις με μεταβλητές, όμως, είναι υποχρεωτικό πρώτα να ορίσουμε τις μεταβλητές. Κάθε μεταβλητή μπορεί να πάρει μόνο μια τιμή. Σε περίπτωση που πάμε να της δώσουμε μια καινούρια τιμή με μια εντολή ανάθεσης, η παλιά τιμή αυτομάτως θα χαθεί. **Είναι σημαντικό να τονιστεί πως το MATLAB ξεχωρίζει τα πεζά με τα κεφαλαία στις μεταβλητές, δηλαδή οι NRe και nRe είναι διαφορετικές μεταβλητές.**

4.3 ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ

Με τη βοήθεια της συνάρτησης plot, παράγεται η γραφική παράσταση των στοιχείων μίας γραμμής ως προς τα στοιχεία μίας άλλης, με δεδομένο, όμως, ότι οι δύο γραμμές έχουν τον ίδιο αριθμό στοιχείων. Είναι καλό πως έχουμε τη δυνατότητα να επεξεργαστούμε το εκάστοτε γράφημα από τη γραμμή εργαλείων, εισάγοντας κείμενο και σχεδιάζοντας γραμμές μεγέθυνσης και σμίκρυνσης. Τη στιγμή που θέλουμε να κλείσουμε το παράθυρο των γραφημάτων, πατάμε την εντολή Close που βρίσκουμε στο μενού των αρχείων. Όταν χρειαστεί να σχηματίσουμε σύνθετες γραφικές παραστάσεις στο MATLAB, μπορούμε και πάλι να χρησιμοποιήσουμε τη συνάρτηση plot. Οι τιμές και τα ονόματα των δυο αξόνων διαλέγονται αυτόματα. Να αναφέρουμε πως για τρισδιάστατες γραφικές παραστάσεις χρησιμοποιείται η εντολή plot3.

4.4 ΠΙΝΑΚΕΣ

➤ ΟΡΙΣΜΟΣ ΠΙΝΑΚΑ

Πίνακας ονομάζονται οι δισδιάστατες σειρές, οι οποίες είναι διατεταγμένα σύνολα αριθμών και προσδιορίζονται από δύο δείκτες. Αυτές οι δισδιάστατες σειρές εισάγονται με τρόπο παρόμοιο της εισαγωγής των μονοδιάστατων σειρών. Μπορούμε είτε να γράψουμε όλες τις γραμμές σε μια σειρά και να τις χωρίζουμε με ελληνικά ερωτηματικά είτε κάθε γραμμή ξεχωριστά και να πατούμε Enter ανάμεσα στις διαδοχικές γραμμές. Τα στοιχεία των πινάκων συμβολίζονται με δύο δείκτες, από τους οποίους ο πρώτος δείχνει τη γραμμή και ο δεύτερος τη στήλη. Οι πίνακες που δημιουργούμε μπορεί να είναι τετραγωνικοί, αλλά μπορεί και να μην είναι.

➤ ΣΤΟΙΧΕΙΩΔΕΙΣ ΠΡΑΞΕΙΣ ΜΕ ΠΙΝΑΚΕΣ

Για να καταφέρουμε να κάνουμε πράξεις μεταξύ δυο πινάκων, είναι επιτακτική ανάγκη οι πίνακες να είναι ίδιας μορφής, δηλαδή ο αριθμός των γραμμών και των στηλών να είναι ίσος και στους δυο πίνακες. Έτσι, μπορούμε να κάνουμε προσθέσεις, αφαιρέσεις και πολλαπλασιασμούς μεταξύ των δυο πινάκων. Επιπροσθέτως, μας επιτρέπεται να φτιάξουμε πίνακες μηδενικούς, μοναδιαίους, με όλα τα στοιχεία τους να είναι 1 και πίνακες τυχαίων στοιχείων.

4.5 M-ΑΡΧΕΙΑ ΣΤΟ ΠΡΟΓΡΑΜΜΑ MATLAB

Στο MATLAB έχουμε δύο τρόπους, για να εισάγουμε δεδομένα. Ο ένας είναι με χρήση του editor του MATLAB, που είναι αυτός που είδαμε παραπάνω. Ο δεύτερος τρόπος είναι με τη χρήση του editor για M-αρχεία του MATLAB, όπου μας δίνεται η δυνατότητα να δημιουργήσουμε από απλές σειρές εντολών μέχρι και κώδικα προγραμματισμού. Τον τρόπο αυτό θα τον αναλύσουμε εκτενώς στη συνέχεια. Με βάση αυτόν τον τρόπο, έγιναν και τα προγράμματα για την περάτωση του πρακτικού μέρους της πτυχιακής εργασίας.

Ένα M-αρχείο είναι ένα αρχείο κειμένου που το αποθηκεύουμε γράφοντας το όνομα του αρχείου με επέκταση **.m** και περιέχει μια σειρά από

εντολές και συναρτήσεις, οι οποίες εκτελούνται από το επίπεδο εντολών του MATLAB γράφοντας απλά το αρχείο **χωρίς την επέκταση .m**. Ένα M-αρχείο μπορεί να περιέχει και άλλα M-αρχεία. Έτσι, το MATLAB εκτελεί κανονικά τις εντολές που έχει το αρχείο χωρίς να χρειάζεται εισαγωγή μεταβλητών αφού χρησιμοποιούνται οι μεταβλητές του χώρου εργασίας. Η επέκταση **.m** είναι πολύ σημαντική στην αποθήκευση ενός M-αρχείου. Αυτό συμβαίνει, γιατί δηλώνουμε πως το αρχείο θα αποθηκευτεί στο work και, κατά την εκτέλεση του M-αρχείου, το MATLAB εκτελεί αποκλειστικά και μόνο όσα βρίσκονται στο φάκελο work.

4.6 ΣΥΝΑΡΤΗΣΕΙΣ

Όταν σε ένα αρχείο τύπου M είναι δηλωμένη η λέξη function στην αρχή της πρώτης γραμμής, σημαίνει πως αναφερόμαστε σε αρχείο συνάρτησης. Κάθε συνάρτηση έχει την ικανότητα να παίρνει μεταβλητές από ένα πρόγραμμα, να τις επεξεργάζεται και να τις δίνει πάλι στο ίδιο πρόγραμμα. Τα αρχεία συναρτήσεων βοηθούν πολύ, στην περίπτωση που θέλουμε να εμπλέξουμε παραπάνω από μια συναρτήσεις μεταξύ τους χρησιμοποιώντας το MATLAB.

4.7 ΑΠΟΘΗΚΕΥΣΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ

Όταν έχουμε ολοκληρώσει ένα M-αρχείο και θέλουμε να το αποθηκεύσουμε, χρησιμοποιούμε την εντολή save ως εξής:

```
save filename.mat
```

Για να εισάγουμε ένα M-αρχείο που είναι αποθηκευμένο μέσα στο work, χρησιμοποιούμε την εντολή load ως εξής:

```
load filename
```

Αφού εισάγουμε το επιθυμητό M-αρχείο, έχουμε τη δυνατότητα ύστερα να το επεξεργαστούμε, όπως εμείς θέλουμε.

4.8 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ

Ύστερα από την αναφορά στη γλώσσα προγραμματισμού MATLAB και τις δυνατότητες που έχουμε μέσα από αυτό, θα ακολουθήσει μια λεπτομερής ανάλυση των προγραμμάτων που χρησιμοποιήθηκαν.

Αρχικά θα δούμε τη διαφορά που υπάρχει ανάμεσα στον DOGHT και στον DCT όσον αφορά την ενέργεια του σήματος. Η σχέση Parseval για το μετασχηματισμό DOGHT είναι:

$$\tilde{E} = \frac{1}{T} \sum_{n=0}^{N-1} |c_n|^2 = \frac{1}{T} \sum_{i=0}^{N-1} w_i \left| f\left(\frac{t_i}{T}\right) \right|^2$$

(Σχέση 3.12)

όπου \tilde{E} είναι η κατά προσέγγιση ενέργεια του σήματος. Ακριβώς παρακάτω είναι το πρόγραμμα που υπολογίζει την ενέργεια για τη μέθοδο του DOGHT:

```
N=8;
T=0.586;
n=0:1:(N-1);
S1=0;
tmin=input('dose timi gia to tmin:');
tmax=input('dose timi gia to tmax:');
C=[1.06 0 -0.18 0 0.10 0 -0.03 0];
for n=0:1:7
Cn1=abs(C(n+1)).^2;
S1=S1+Cn1;
end
E=(1/T)*S1
end
```

Στο πρόγραμμα αυτό ο αριθμός των δεδομένων δεν ζητείται από τον χρήστη, και η τιμή που έχουμε πάρει είναι 8 δεδομένα. Με δεδομένο επίσης τον συντελεστή κλιμάκωσης $T=0.586$ αλλά και τις τιμές των συντελεστών μετασχηματισμού μέσα από πίνακα, ζητείται από το χρήστη να δώσει μόνο τα

όρια της δειγματοληψίας. Στην πορεία, μέσα από ένα βρόχο επανάληψης υπολογίζεται ένα άθροισμα $S1$, το οποίο μας βοηθάει τελικά στον υπολογισμό της ενέργειας E . Τρέχοντας λοιπόν, το πρόγραμμα στο MATLAB παίρνουμε σαν αποτέλεσμα για την ενέργεια $E=1.9913$. Το αντίστοιχο παράδειγμα για τη μέθοδο του DCT από το οποίο υπολογίζεται η τιμή της ενέργειας του σήματος E είναι το ακόλουθο:

```

N=8;
T=0.586;
S1=0;
n=0:1:(N-1);
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);
for n=0:1:7
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:8
tk=tmin+k*Dt;
S=S+(2/(exp(tk)+exp(-tk)))*(cos((n*pi*(k+(1/2)))/N));
end
Cn=((2*a)/N)*S;
Cn1=abs(Cn).^2;
S1=S1+Cn1;
end
E=((N*Dt)/2)*S1

```

Στην περίπτωση αυτή, αφού υπολογίσει το πρόγραμμα τις τιμές του συντελεστή μετασχηματισμού C_n μέσα από τη FOR, στη συνέχεια τις προσθέτει σε μια μεταβλητή. Η ενέργεια του σήματος με βάση τον επόμενο τύπο: $E=((N*Dt)/2)*S1$, έχει την τιμή $E= 1.9447$.

Η σχέση Parseval είναι πολύ χρήσιμη, για να μπορέσουμε να προσδιορίσουμε τον αριθμό των σημείων δειγματοληψίας που απαιτούνται, για να προσεγγίσουμε καλύτερα την ενέργεια του σήματος, για τη σωστή επιλογή του συντελεστή κλιμάκωσης του χρόνου T , όπως και γενικότερα για την εκτίμηση του ρυθμού σύγκλισης του μετασχηματισμού. Η σχέση Parseval δείχνει πως στην περίπτωση του μετασχηματισμού DOGHT η ενέργεια του σήματος προσεγγίζεται από ένα αριθμητικό ολοκλήρωμα του τύπου Gauss-Hermite, ενώ στην περίπτωση του μετασχηματισμού DCT, όπως γνωρίζουμε ήδη από τη σχέση Parseval του DCT, η ενέργεια του σήματος προσεγγίζεται από ένα αριθμητικό ολοκλήρωμα του τύπου παραλληλογράμμου:

$$\tilde{E} = \frac{N \Delta t}{2} \sum_{n=0}^{N-1} |c_n|^2 = \Delta t \sum_{k=0}^{N-1} |f(t_k)|^2 \quad (\text{Σχέση 3.13})$$

Γνωρίζουμε, εξάλλου, πως το αριθμητικό ολοκλήρωμα του τύπου Gauss-Hermite συγκλίνει με πολύ μεγαλύτερη ταχύτητα από τον απλό κανόνα του παραλληλογράμμου στην περισσότερες περιπτώσεις. Διαπιστώνουμε πως ο DCT προσεγγίζει λιγότερο καλά την ενέργεια του σήματος. Ως προς την επιλογή του συντελεστή κλιμάκωσης του χρόνου T για το μετασχηματισμό DOGHT σε αναλογία με το μετασχηματισμό DCT, διαλέγουμε αρχικά ένα χρονικό παράθυρο $[-t_{\max}, t_{\max}]$ σε σχέση με την χρονική διάρκεια των σημάτων που μας ενδιαφέρουν. Για το συντελεστή κλιμάκωσης του χρόνου T υπάρχει η ακόλουθη:

$$T = \frac{t_{N-1}}{t_{\max}} \quad (\text{Σχέση 3.14})$$

όπου t_{N-1} είναι το μεγαλύτερο μηδενικό του πολυωνύμου Hermite σύμφωνα με τη θεωρία ολοκλήρωσης κατά Gauss-Hermite. Έτσι με την επιλογή αυτή το πρώτο σημείο δειγματοληψίας $(t_0 / T) = -t_{\max}$ και το τελευταίο $(t_{N-1} / T) = +t_{\max}$ ταυτίζονται με τα άκρα του υπολογιστικού παραθύρου $[-t_{\max}, t_{\max}]$. Στην πορεία,

για τα παρακάτω παραδείγματα θα χρησιμοποιήσουμε για τον συντελεστή κλιμάκωσης χρόνου την τιμή $T=0.586$. [7,8,9]

ΠΑΡΑΔΕΙΓΜΑ 1

Ως παράδειγμα θα υπολογίσουμε τον μετασχηματισμό DOGHT της συνάρτησης $f(t)=\text{sech}(t)$ στο χρονικό παράθυρο $t \in [-5,5]$ με $N=8$ σημεία. Τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες. Η αντίστροφα μετασχηματισμένη κατά DOGHT συνάρτηση δίνεται από την ακόλουθη σχέση στο πεδίο του χρόνου:

$$\tilde{f}(t_i) = \sum_{n=0}^{N-1} c_n h_n(t_i T) \quad (\text{Σχέση 3.15})$$

Μέσα από το πρόγραμμα αυτό έχουμε καταφέρει να υπολογίσουμε τις παραμέτρους C_n , t_n , και f_{in} για το διακριτό μετασχηματισμό συνημιτόνου DCT. Το πρόγραμμα έρχεται σε άμεση επαφή με το χρήστη αφού για κάθε βήμα του θέτει ένα ερώτημα στο οποίο είναι υποχρεωμένος ο χρήστης να απαντήσει για να συνεχίσει να τρέχει το πρόγραμμα. Έτσι, λοιπόν, στην αρχή του προγράμματος ζητείται από το χρήστη να δώσει τον αριθμό N δεδομένων στάθμης σήματος με τα οποία θέλει να εφαρμοστεί το συγκεκριμένο παράδειγμα. Παίρνουμε σαν δεδομένη την τιμή του συντελεστή κλιμάκωσης, μέσα από υπολογισμούς, $T=0.586$. Στη συνέχεια το n παίρνει όλες τις τιμές από 0 έως $N-1$ με βήμα 1. Το n είναι μια απλή μεταβλητή. Αμέσως μετά ζητούμε από το χρήστη να δώσει τα δυο όρια της δειγματοληψίας έτσι ώστε να είναι ομοιόμορφη και να έχει συμμετρικό διάστημα. Από τις τιμές των n και t_{min} , t_{max} και με δεδομένο ότι $T=0.586$ είναι εύκολο να υπολογίσουμε τις τιμές των C_n , t_n , και f_{in} . Έχει δημιουργηθεί και ένας πίνακας C ο οποίος ορίστηκε με μηδενικά και αποθηκεύω εκεί όλες τις νέες τιμές του συντελεστή μετασχηματισμού που υπολογίζω. Αν το $n=0$ τότε η τιμή του a είναι $a=1/\sqrt{2}$, ενώ για κάθε τιμή του n μεγαλύτερη του 0, $a=1$. Ο τύπος που μας δίνει τη δειγματοληψία στο συμμετρικό διάστημα είναι $t_k=t_{min}+k \cdot Dt$ με $Dt=(2 \cdot t_{max})/(N-1)$. Υπολογίζοντας μέσα στη FOR για όλες τις τιμές του N κάθε φορά την τιμή του t_k , και μέσα από την ακόλουθη σχέση:

$$S=S+(2/(\exp(tk)+\exp(-tk)))*(\cos((n*\pi*(k+(1/2)))/N));$$

μπορώ να υπολογίζω την τιμή του συντελεστή μετασχηματισμού C_n για το DCT, την τιμή των διαφόρων χρόνων δειγματοληψίας $t_n=t_i/T$ και την τιμή του δειγματοληφθέντος σήματος για κάθε t_n , που την ονομάζω fin .

```
function secht
N=input('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
tmin=input('dose timi gia to tmin:');
tmax=input('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(2/(exp(tk)+exp(-tk)))*cos((n*pi*(k+(1/2)))/N);
end
Cn=((2*a)/N)*S;
tn=tmin+n*Dt;
fin=2/(exp(tn)+exp(-tn));
end
```

Αμέσως από κάτω, στο επόμενο πρόγραμμα, αρχίζει η διαδικασία της αποσυμπίεσης.


```

function aposumpiesi
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
S1=0;
l=0;
C=[0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(2/(exp(tk)+exp(-tk)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end

```

S
end

Η διαδικασία της αποσυμπίεσης είναι παρόμοια με αυτή της συμπίεσης. Στην περίπτωση αυτή τα αποτελέσματα Cn μπαίνουν σε μια νέα FOR. Γίνεται και πάλι έλεγχος του n για να πάρει το a την τιμή που πρέπει και μέσα από την εξίσωση που ακολουθεί:

$$S=S+a*C(n+1)*(\cos((n*\pi*(l+1/2))/N))$$

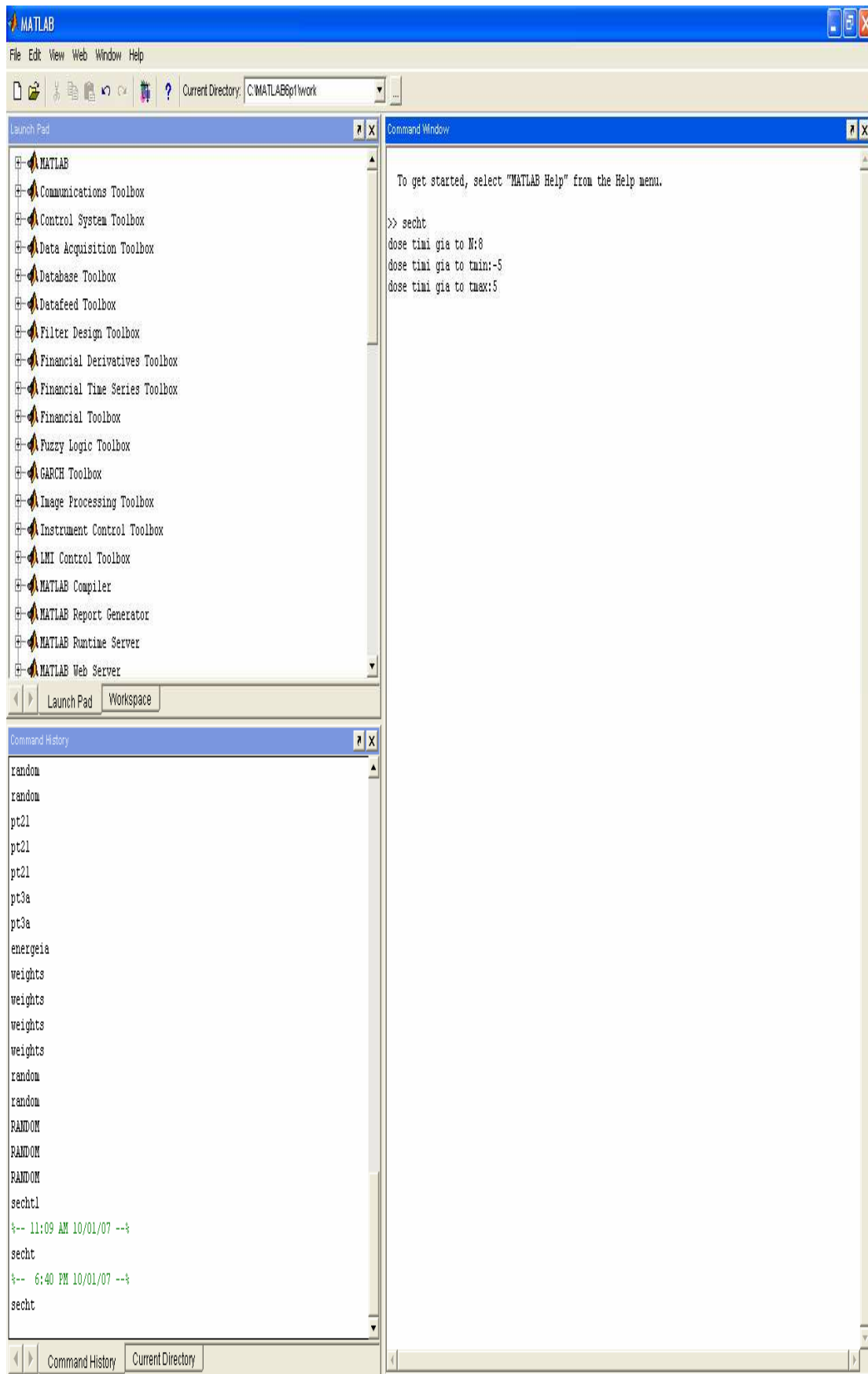
υπολογίζω κάθε τιμή για τον αντίστροφο μετασχηματισμό για το παράδειγμα αυτό.

l	Cn-DOGHT	Cn-DCT
0	1.06	0.3839
1	0.00	-0.0041
2	-0.18	-0.3945
3	0.00	-0.0035
4	0.10	0.1805
5	0.00	-0.0023
6	-0.03	-0.0686
7	0.00	0

l	tn=ti/T-DOGHT	tn=ti/T -DCT
0	-5.00	-5
1	-3.38	-3.5714
2	-1.97	-2.1429
3	-0.65	-0.7143
4	0.65	0.7143
5	1.97	2.1429
6	3.38	3.5714
7	5.00	5

l	$f_{in}=f(t_i/T)$ -DOGHT	$f_{in}=f(t_i/T)$ -DCT
0	0.01	0
1	0.07	0.0045
2	0.27	0.0516
3	0.82	0.5436
4	0.82	0.5436
5	0.27	0.0516
6	0.07	0.0045
7	0.01	0

l	$f_{fin}=f(t_i/T)$ -DOGHT	$f_{fin}=f(t_i/T)$ -DCT
0	0.01	0
1	0.07	0.0562
2	0.27	0.2315
3	0.82	0.7898
4	0.82	0.7898
5	0.27	0.2315
6	0.07	0.0562
7	0.01	0.0167



MATLAB

File Edit View Web Window Help

Current Directory: C:\MATLAB6p1\work

Launch Pad

- MATLAB
- Communications Toolbox
- Control System Toolbox
- Data Acquisition Toolbox
- Database Toolbox
- Datafeed Toolbox
- Filter Design Toolbox
- Financial Derivatives Toolbox
- Financial Time Series Toolbox
- Financial Toolbox
- Fuzzy Logic Toolbox
- GARCH Toolbox
- Image Processing Toolbox
- Instrument Control Toolbox
- LMI Control Toolbox
- MATLAB Compiler
- MATLAB Report Generator
- MATLAB Runtime Server
- MATLAB Web Server
- MPC Toolbox
- Mapping Toolbox
- Motorola DSP Developer's Kit
- Mu-Analysis and Synthesis Toolbox
- Neural Network Toolbox
- Optimization Toolbox
- Partial Differential Equation Toolbox

Workspace

Command Window

To get started, select "MATLAB Help" from the Help menu.

```
>> secht
dose timi gia to N:8
dose timi gia to tmin:-5
dose timi gia to tmax:5

Cn =

    0.3839

tn =

    -5

tend =

   -8.5324

fend =

  3.9395e-004

Cn =

   -0.0041

tn =

   -3.5714

tend =

   -6.0946

fend =

    0.0045

Cn =

   -0.3945

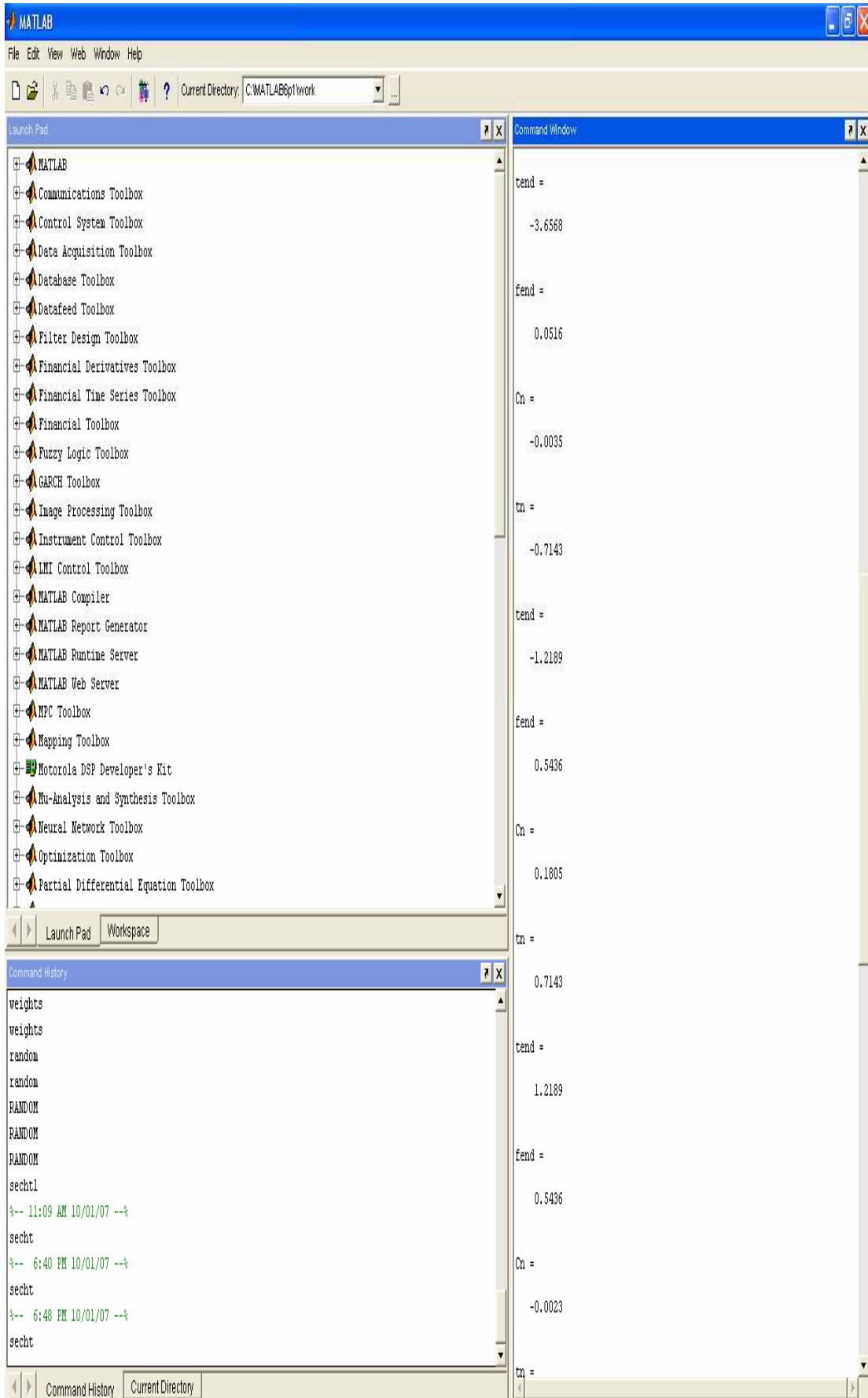
tn =

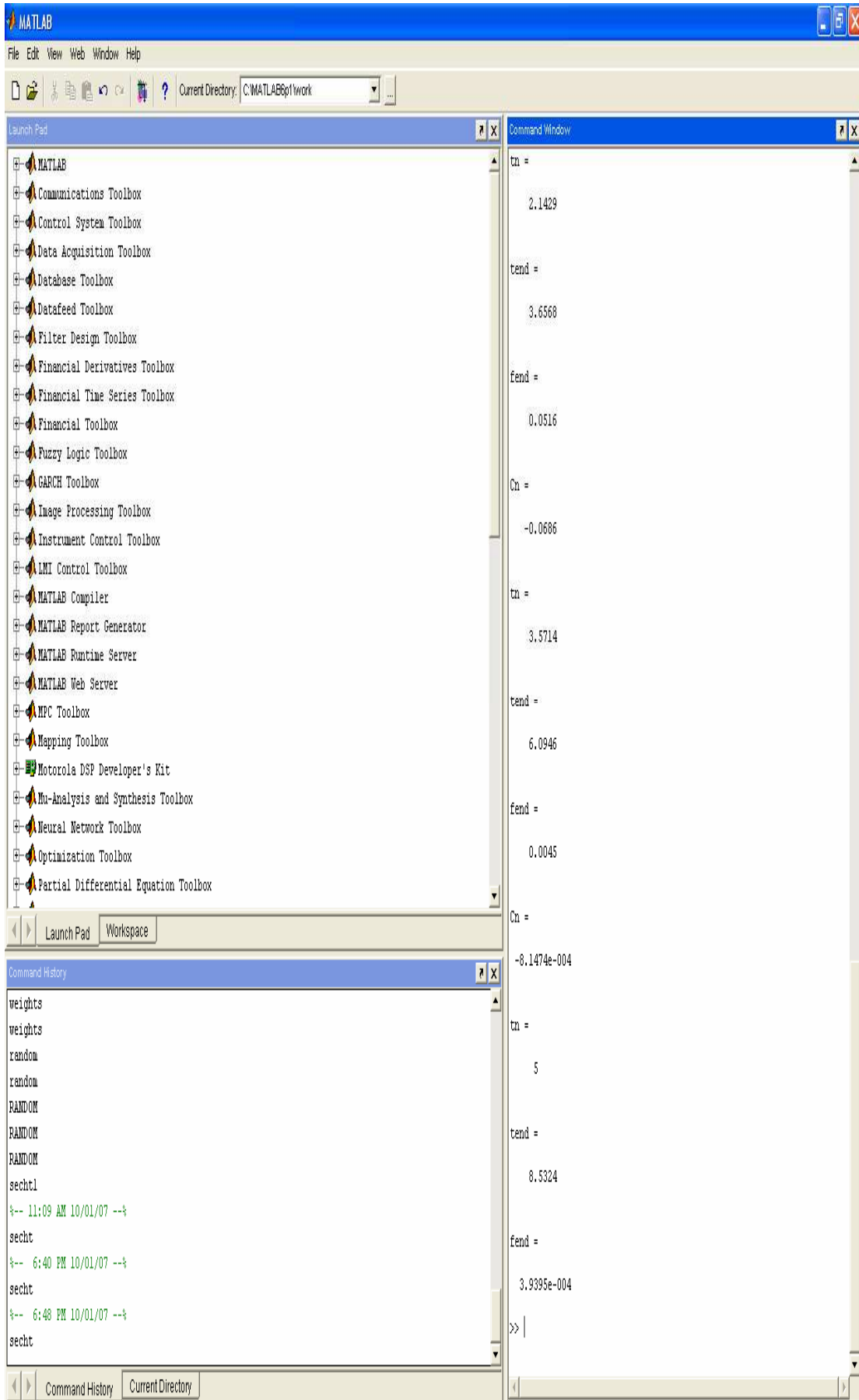
   -2.1429
```

Command History

```
weights
weights
random
random
RANDOM
RANDOM
RANDOM
secht1
%-- 11:09 AM 10/01/07 --%
secht
%-- 6:40 PM 10/01/07 --%
secht
%-- 6:48 PM 10/01/07 --%
secht
```

Command History Current Directory





Στο συγκεκριμένο παράδειγμα ο συντελεστής κλιμάκωσης του χρόνου, μέσα από υπολογισμούς, είναι $T=0.586$. Από τον πρώτο πίνακα που υπολογίζει τους συντελεστές μετασχηματισμού των DOGHT και DCT καθίσταται σαφές ότι ο αντίστροφος μετασχηματισμός DOGHT είναι ακριβής και ότι οι περιττοί συντελεστές του DOGHT είναι μηδενικοί εξαιτίας της άρτιας συμμετρίας της συνάρτησης υπερβολικής συντέμνουσας. Στον ίδιο πίνακα παρατηρούμε και τα αντίστοιχα αποτελέσματα για τον μετασχηματισμό DCT. Το σπουδαιότερο, όμως, στοιχείο είναι πως με τον DOGHT μπορούμε να πραγματοποιήσουμε μεγαλύτερη συγκέντρωση της ενέργειας του σήματος στους συντελεστές χαμηλότερης τάξης (χαμηλότερες «συχνότητες»). Κατά συνέπεια, ο DOGHT μας δίνει τη δυνατότητα να επιτύχουμε καλύτερη συμπίεση. Χαρακτηριστική είναι η περίπτωση των συντελεστών μηδενικής τάξης όπου έχουμε $c_0=0.39$ για τον DCT (συντελεστής DC) και $c_0=1.06$ για τον DOGHT.

ΠΑΡΑΔΕΙΓΜΑ 2

Ως παράδειγμα θα συγκρίνουμε τη συμπίεση του σήματος $f(t)=\exp(-|t|)$ στο χρονικό παράθυρο $t \in [-5,+5]$ με $N = 16$ σημεία από τον DCT και τον DOGHT με δευτεροβάθμια παρεμβολή. Ως μέτρο σύγκρισης αυτών των δύο μεθόδων θα χρησιμοποιήσουμε το κανονικοποιημένο rms σφάλμα υπολογισμένο στα σημεία της ομοιόμορφης δειγματοληψίας (PRD – Percent RMS Difference):

$$PRD = 100 * \left[\frac{\sum_{k=0}^{N-1} [f(t_k) - \tilde{f}(t_k)]^2}{\sum_{k=0}^{N-1} [f(t_k)]^2} \right]^{1/2}$$

Σε αυτό το πρόγραμμα υπολογίζουμε και πάλι τις παραμέτρους C_n , t_n , και fin για το διακριτό μετασχηματισμό συνημιτόνου DCT μέσα από μια άλλη όμως εξίσωση, η οποία αυτή τη φορά είναι εκθετική. Το πρόγραμμα βρίσκεται σε άμεση επαφή με το χρήστη αφού και σε αυτή την περίπτωση, σε κάθε βήμα ζητά από το χρήστη να δίνει διάφορες τιμές στις μεταβλητές για να

καταφέρει το πρόγραμμα να δώσει αποτελέσματα. Έτσι, λοιπόν, στην αρχή του προγράμματος ζητείται από το χρήστη να δώσει τον αριθμό N δεδομένων στάθμης σήματος με τα οποία θέλει να εφαρμοστεί το συγκεκριμένο παράδειγμα. Παίρνουμε σαν δεδομένη την τιμή του συντελεστή κλιμάκωσης που βγαίνει μετά από υπολογισμούς ίση με $T=0.586$. Στη συνέχεια το n που είναι μια απλή μεταβλητή παίρνει όλες τις τιμές από 0 έως N-1 με βήμα 1. Κατόπιν αυτού ζητείται από το χρήστη να δώσει τα δυο όρια της δειγματοληψίας έτσι ώστε να είναι ομοιόμορφη και να έχει συμμετρικό διάστημα. Από τις τιμές των n και tmin, tmax και με δεδομένο ότι $T=0.586$ είναι βατή η διαδικασία του υπολογισμού των τιμών των Cn, tn, και fin. Αν το n=0 τότε η τιμή του a είναι $a=1/(\sqrt{2})$, ενώ για κάθε τιμή του n μεγαλύτερη του 0, $a=1$. Ο τύπος που μας δίνει τη δειγματοληψία στο συμμετρικό διάστημα είναι $t_k=t_{min}+k \cdot Dt$ με $Dt=(2 \cdot t_{max})/(N-1)$. Υπολογίζοντας και πάλι μέσα στη FOR για όλες τις τιμές του N κάθε φορά την τιμή του tk και μέσα από την ακόλουθη σχέση:

$$S=S+(\exp(-\text{abs}(t_k))) \cdot (\cos((n \cdot \pi) \cdot (k+(1/2)))/N)$$

μπορώ να υπολογίζω την τιμή του συντελεστή μετασχηματισμού Cn για το DCT την τιμή των διαφόρων χρόνων δειγματοληψίας $t_n=t_i/T$ και την τιμή του δειγματοληφθέντος σήματος για κάθε tn, που την ονομάζω fin. Η διαφορά αυτού του προγράμματος από το πρώτο είναι ότι εδώ έχουμε εξίσωση εκθετικής μορφής.

```
function secht1
N=16;
T=0.586;
n=0:1:(N-1);
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);
for n=0:1:15
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
```

```

end
S=0;
for k=1:1:8
tk=tmin+k*Dt;
S=S+(exp(-abs(tk)))*(cos((n*pi*(k+(1/2)))/N));
end
Cn=((2*a)/N)*S;
tn=tmin+n*Dt;
fin=(exp(-abs(tn)))
end

```

Αμέσως μετά ακολουθεί το πρόγραμμα της αποσυμπίεσης το οποίο και σε αυτή την περίπτωση είναι παρόμοιο με αυτό της συμπίεσης. Εδώ τα αποτελέσματα C_n μπαίνουν σε μια FOR, ελέγχεται το n για να πάρει το a την τιμή που πρέπει και με βάση την επόμενη εξίσωση:

$$S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N))$$

υπολογίζω κάθε τιμή του αντίστροφου μετασχηματισμού.

```

function aposumpiesi1
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
S1=0;
l=0;
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end

```

```

S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end
for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S
end

```

Αφού υπολογίσαμε στα δυο προηγούμενα παραδείγματα τη συμπίεση και την αποσυμπίεση για την εξίσωση $fin=(\exp(-\text{abs}(tn)))$ τώρα με βάση το κανονικοποιημένο rms σφάλμα PRD υπολογισμένο στα σημεία της ομοιόμορφης δειγματοληψίας θα μπορέσουμε να συγκρίνουμε τις δυο μεθόδους για να δούμε και πάλι ποια είναι καλύτερη.

```

function PRD_DCT_1_1
N=input('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
S1=0;
l=0;
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

```

```

C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end

for n=0:1:(N-1)

```

```

tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1)))));
end

SUM1=0;
SUM2=0;
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end

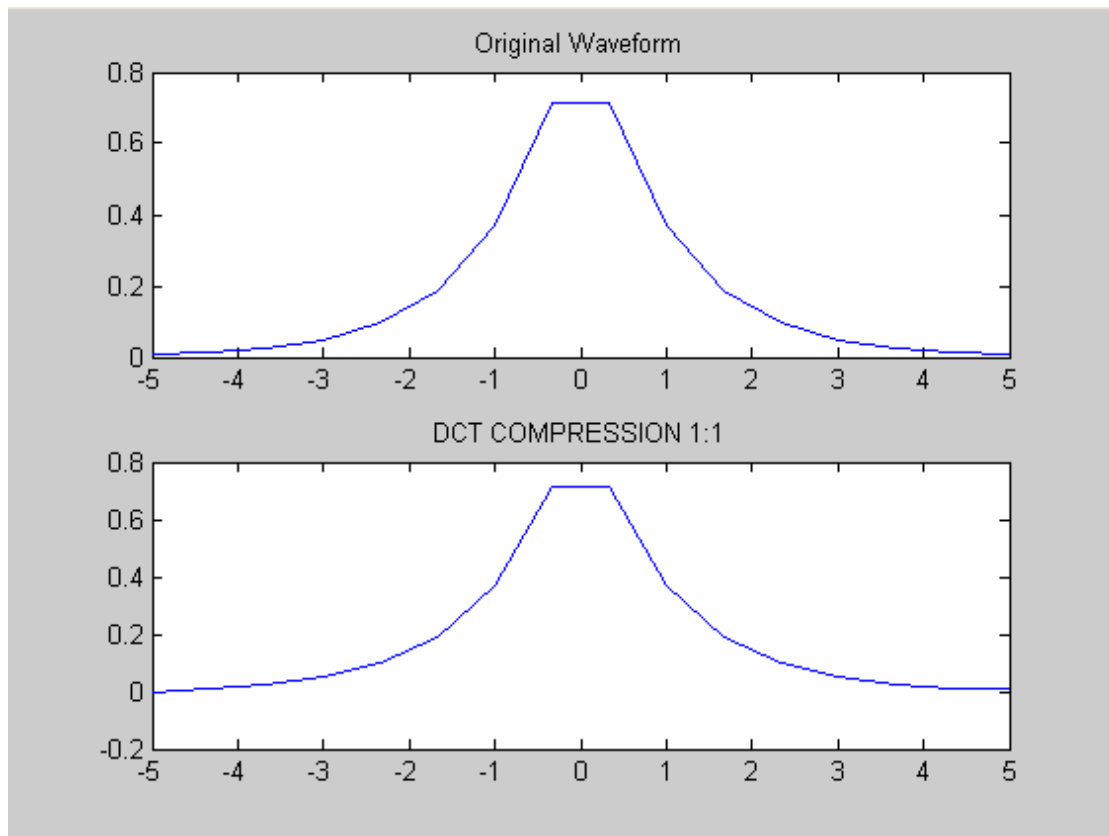
PRD=100*((SUM1/SUM2)^0.5)

```

```

subplot(2,1,1); plot(tn,fin);title('Original Waveform')
subplot(2,1,2); plot(tn,fout);title('DCT COMPRESSION 1:1')

```



```

function PRD_DCT_3_4_1
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
S1=0;
l=0;
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);

for n=0:1:(N-1)
tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1)))));
end

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end
for l=12:1:15
C(l+1)=0;
end

```

```

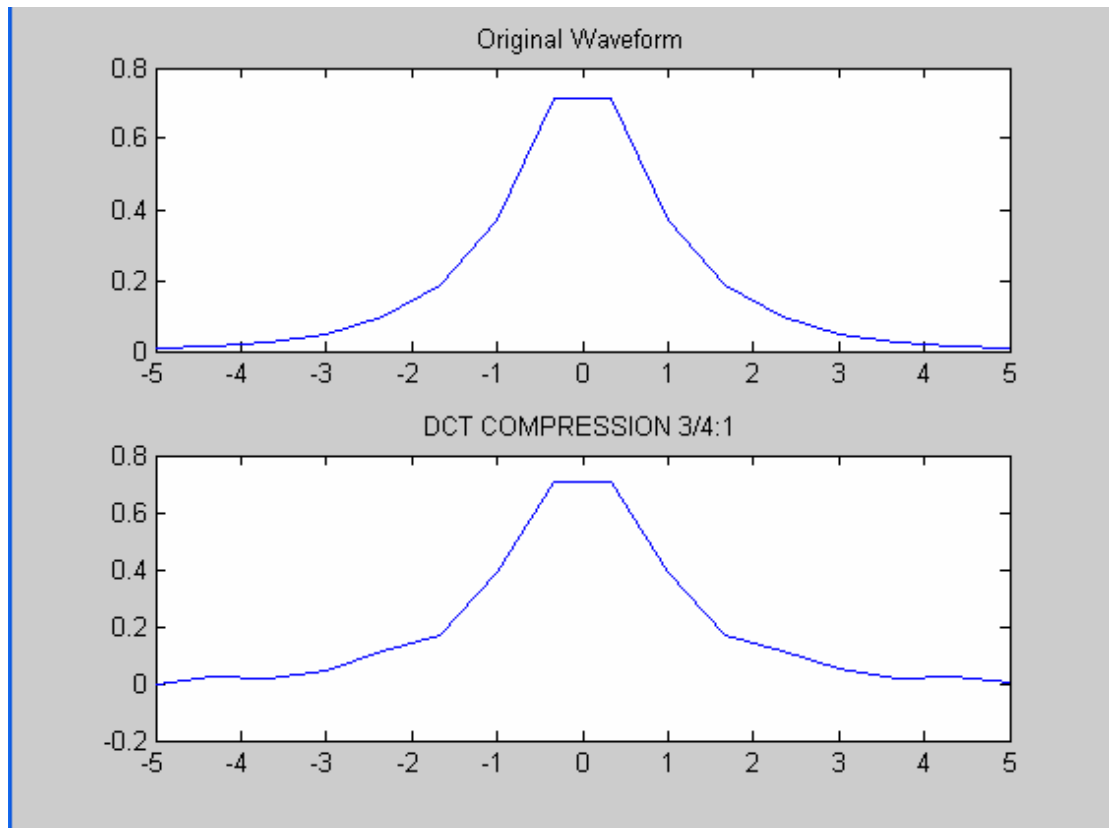
for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end

SUM1=0;
SUM2=0;
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end

PRD=100*((SUM1/SUM2)^0.5)

subplot(2,1,1); plot(tn,fin);title('Original Waveform')
subplot(2,1,2); plot(tn,fout);title('DCT COMPRESSION 3/4:1')

```



```

function PRD_DCT_2_1
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
S1=0;
l=0;
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);
for n=0:1:(N-1)
tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1)))));
end

```



```

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end

```

```

for l=7:1:15
C(l+1)=0;
end

```

```

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end
SUM1=0;
SUM2=0;

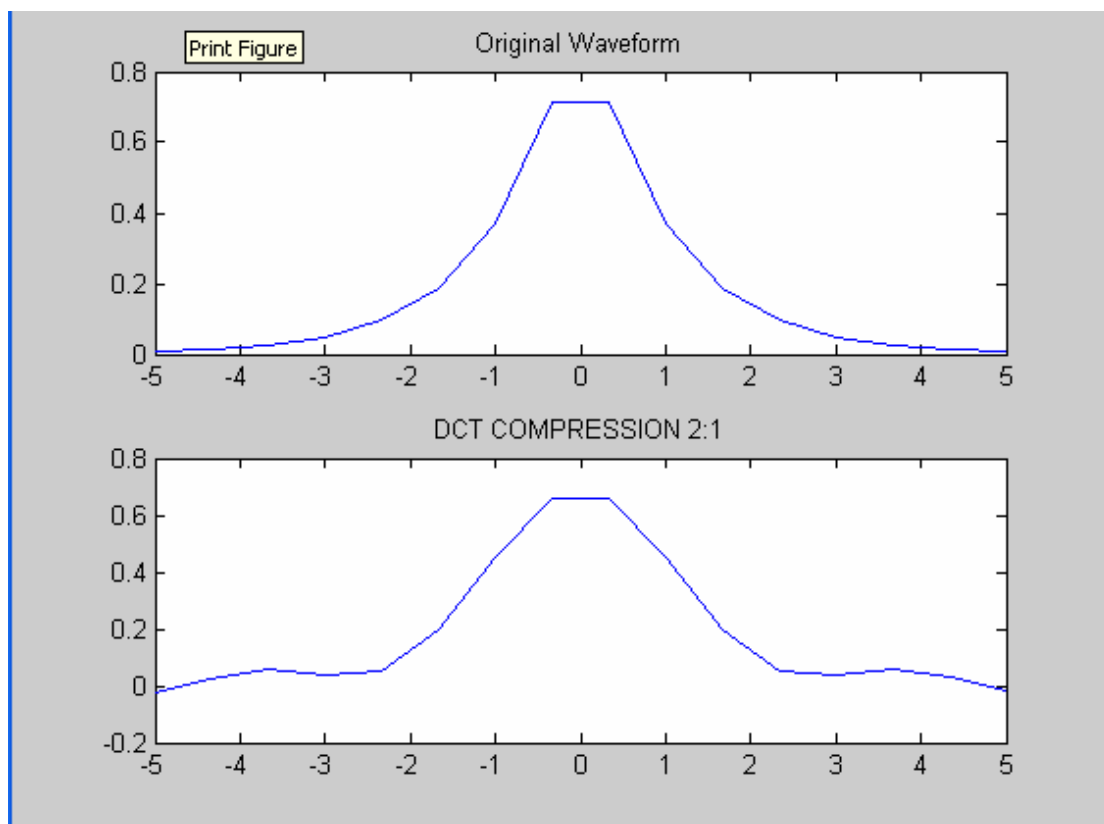
```

```
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end
```

```
PRD=100*((SUM1/SUM2)^0.5)
```

```
subplot(2,1,1); plot(tn,fin);title('Original Waveform')
```

```
subplot(2,1,2); plot(tn,fout);title('DCT COMPRESSION 2:1')
```



```

function PRD_DCT_4_1
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
S1=0;
l=0;
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
Dt=(2*tmax)/(N-1);

for n=0:1:(N-1)
tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1)))));
end

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end
for l=4:1:15
C(l+1)=0;
end

```

```

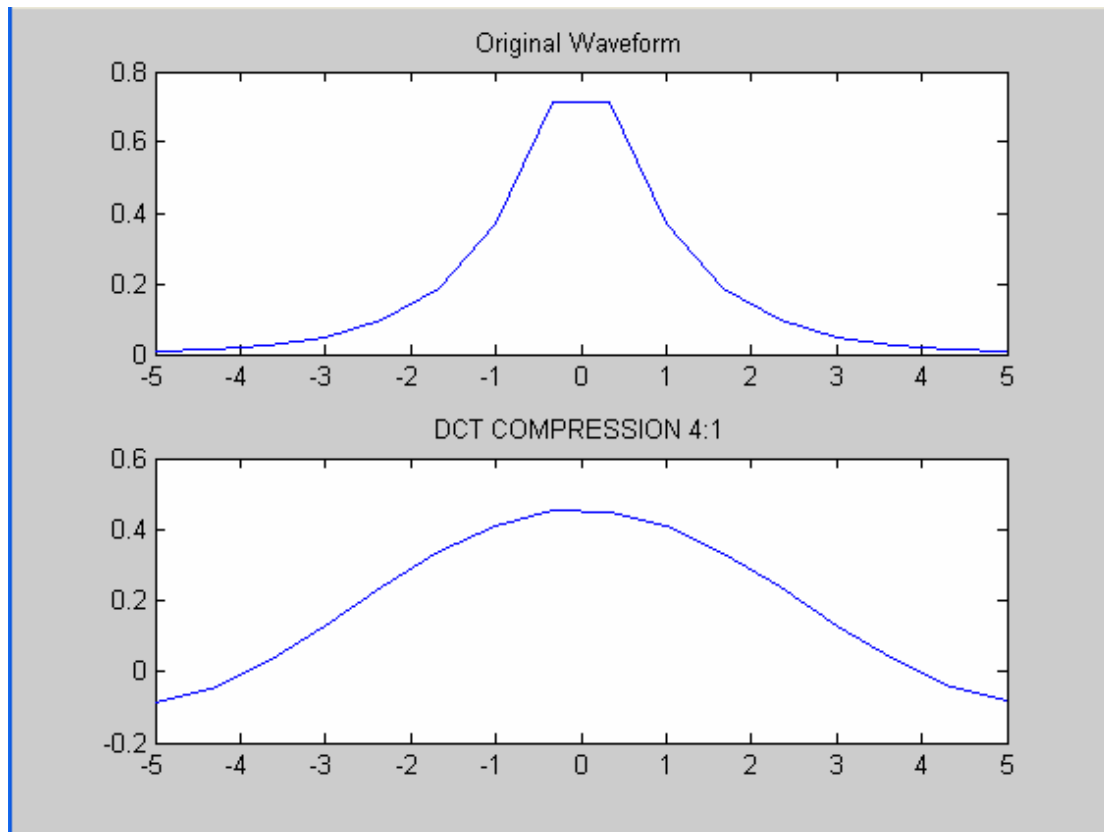
for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end

SUM1=0;
SUM2=0;
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end

PRD=100*((SUM1/SUM2)^0.5)

subplot(2,1,1); plot(tn,fin);title('Original Waveform');
subplot(2,1,2); plot(tn,fout);title('DCT COMPRESSION 4:1');

```



Τα παραπάνω 4 προγράμματα από τα οποία υπολογίσαμε το σφάλμα PRD έχουν δημιουργηθεί ακριβώς με τον ίδιο τρόπο. Εξαρχής, όπως σε όλα τα προγράμματα είναι καλό να αναφέρουμε πως το πρόγραμμα έχει άμεση επαφή με το χρήστη. Αυτό εύκολα το συμπεραίνουμε από το γεγονός ότι και πάλι ζητείται από το χρήστη να δίνει διάφορες τιμές στις μεταβλητές για να δώσει το πρόγραμμα αποτελέσματα. Έτσι, στην αρχή το πρόγραμμα θέλει από το χρήστη να δώσει τον αριθμό των δεδομένων στάθμης σήματος N με τα οποία επιθυμεί να εφαρμοστεί το συγκεκριμένο παράδειγμα. Η τιμή του συντελεστή κλιμάκωσης είναι δεδομένη και βγαίνει μετά από υπολογισμούς ίση με $T=0.586$. Στη συνέχεια το n που είναι μια απλή μεταβλητή παίρνει όλες τις τιμές από 0 έως $N-1$ με βήμα 1. Αμέσως μετά ορίζουμε τους πίνακες fin και $fout$, ο πρώτος μου δίνει το κανονικό σήμα συμπιεσμένο και ο δεύτερος το αποσυμπιεσμένο σήμα κάθε φορά, που θα μας βοηθήσουν να υπολογίσουμε το σφάλμα. Ο πίνακας $CoUt$ μου δίνει όλες τις τιμές του συντελεστή μετασχηματισμού από τις οποίες στην πορεία θα υπολογίσω την αποσυμπίεση. Κατόπιν αυτού ο χρήστης πρέπει να δώσει τα δυο όρια της

δειγματοληψίας έτσι ώστε να είναι ομοιόμορφη και να έχει συμμετρικό διάστημα. Μετά από αυτά ελέγχεται το n για να δούμε ποια τιμή θα πάρει. Αν το $n=0$ τότε η τιμή του a είναι $a=1/(\sqrt{2})$, ενώ για κάθε τιμή του n μεγαλύτερη του 0, $a=1$. Ο τύπος που μας δίνει τη δειγματοληψία στο συμμετρικό διάστημα είναι $t_k=t_{\min}+k\Delta t$ με $\Delta t=(2*t_{\max})/(N-1)$. Υπολογίζοντας και πάλι μέσα στη FOR για όλες τις τιμές του N κάθε φορά την τιμή του t_k και μέσα από την ακόλουθη σχέση:

$$S=S+(\exp(-\text{abs}(t_k))) * (\cos((n*\pi*(k+(1/2)))/N))$$

μπορώ και υπολογίζω τον πίνακα της αποσυμπίεσης. Ύστερα από τον πίνακα αυτόν και από τον πίνακα της συμπίεσης πάω να βρω το σφάλμα για κάθε μια από τις 4 περιπτώσεις. Ορίζω δυο αθροίσματα στην αρχή να είναι μηδέν και μέσα από μια FOR τα υπολογίζω με βάση βέβαια και εδώ τους πίνακες f_{in} και f_{out} . Τέλος βρίσκω πόσο είναι το PRD από την ακόλουθη σχέση:

$$PRD=100*((SUM1/SUM2)^{0.5})$$

και βγάζω σε μορφή σχεδιαγράμματος για όλες τις περιπτώσεις το κανονικό σήμα σε σύγκριση με αυτό της αποσυμπίεσης. Παρατηρούμε από τις κυματομορφές ότι η μέθοδος του DOGHT είναι πολύ καλή για σήματα που έχουν κυματομορφή σχήματος καμπάνας γιατί βλέπουμε πως εκεί που χάνει η μέθοδος του DOGHT είναι μόνο στα άκρα της κυματομορφής.

Είναι σημαντικό να πούμε και τον τρόπο με τον οποίο διαμορφώνεται η κάθε μια κατηγορία συμπίεσης. Όταν μιλάμε για συμπίεση 2:1 εννοούμε πως για τους πρώτους 8 συντελεστές μετασχηματισμού κρατάμε την τιμή που έχουν ενώ τους άλλους 8 τους μηδενίζουμε. Αντίστοιχα στη συμπίεση 4:1 κρατάμε τους 4 πρώτους συντελεστές και μηδενίζουμε τους υπόλοιπους. Όμοια συμβαίνει και για τη συμπίεση (3/4):1.

l	Cn –DOGHT	Cn -DCT
0	0.259	0.2588
1	0.000	-0.0013
2	-0.273	-0.2732
3	0.000	-0.0012
4	0.149	0.1487
5	0.000	-0.0011
6	-0.084	-0.0840
7	0.000	0
8	0.049	0.0483
9	0.000	0
10	-0.029	-0.0296
11	0.000	0
12	0.017	0.0165
13	0.000	0
14	-0.008	-0.0078
15	0.000	0

l	$tn=ti/T$ -DOGHT	$tn=ti/T$ -DCT
0	-5.000	-5
1	-4.126	-4.3333
2	-3.388	-3.6667
3	-2.715	-3
4	-2.081	-2.3333
5	-1.472	-1.6667
6	-0.878	-1
7	-0.292	-0.3333
8	0.292	0.3333
9	0.878	1
10	1.472	1.6667
11	2.081	2.3333
12	2.715	3
13	3.388	3.6667
14	4.126	4.3333
15	5.000	5

l	$f_{in}=f(t_i/T)$ -DOGHT	$f_{in}=f(t_i/T)$ -DCT
0	0.007	0
1	0.013	0.0131
2	0.026	0.0256
3	0.050	0.0498
4	0.097	0.0970
5	0.189	0.1889
6	0.368	0.3679
7	0.717	0.7165
8	0.717	0.7165
9	0.368	0.3679
10	0.189	0.1889
11	0.097	0.0970
12	0.050	0.0498
13	0.026	0.0256
14	0.013	0.0131
15	0.007	0

l	ffin=f(ti/T)-DOGHT	ffin=f(ti/T)-DCT
0	0.007	0
1	0.013	0.0131
2	0.026	0.0256
3	0.050	0.0498
4	0.097	0.0970
5	0.189	0.1889
6	0.368	0.3679
7	0.717	0.7165
8	0.717	0.7165
9	0.368	0.3679
10	0.189	0.1889
11	0.097	0.0970
12	0.050	0.0498
13	0.026	0.0256
14	0.013	0.0131
15	0.007	0.0067

Πρέπει να αναφέρουμε πως οι τιμές στους παραπάνω πίνακες αντιπροσωπεύουν τη συμπίεση 1:1.

N	PRD-DOGHT (%)	PRD-DCT (%)
16	3.57	0.64
12	7.54	4.44
8	10.54	14.29
4	16.75	43.37

Από τον πίνακα του ποσοστιαίου RMS σφάλματος φαίνεται εύκολα πως ενώ ο DCT για συμπίεσεις μικρής μορφής όπως 1:1 και (3/4):1 είναι καλύτερος αφού μας δίνει μικρότερο σφάλμα από τον DOGHT, εντούτοις παρατηρούμε πως για μεγαλύτερες συμπίεσεις όπως 2:1 ή 4:1, ο DOGHT είναι πολύ καλύτερος από τον DCT. Αυτό συμβαίνει γιατί το ποσοστό σφάλματος για τον DCT ανεβαίνει κατακόρυφα κάτι το οποίο δε ισχύει για τον

DOGHT. Η διαφορά αυτή είναι φανερή στην περίπτωση της συμπίεσης 4:1 όπου έχουμε για τον DOGHT 16.75% σφάλμα ενώ για τον DCT έχουμε σφάλμα της τάξης του 43.37%.

ΠΑΡΑΔΕΙΓΜΑ 3

Ως παράδειγμα 3, θα συγκρίνουμε τη συμπίεση του σήματος $f(t)=\exp(-|t|)+0.1*(\text{random} - 0.5)$ στο χρονικό παράθυρο $t \in [-5,+5]$ με $N = 16$ σημεία από τον DCT και τον DOGHT με δευτεροβάθμια παρεμβολή. Όπου random είναι τυχαίοι αριθμοί στο διάστημα $t \in [0,1]$.

Σε αυτό το τρίτο πρόγραμμα υπολογίζουμε για ακόμη μια φορά τις παραμέτρους C_n , t_n , και f_{in} για το διακριτό μετασχηματισμό συνημιτόνου DCT μέσα από μια καινούρια εξίσωση. Όπως στα προηγούμενα δυο παραδείγματα έτσι και εδώ ο χρήστης είναι σε άμεση επαφή με το πρόγραμμα. Αυτό φαίνεται από το γεγονός ότι ο χρήστης είναι αυτός που δίνει τις διάφορες τιμές στις μεταβλητές για να είναι σε θέση το πρόγραμμα να δώσει αποτελέσματα. Με τον τρόπο αυτό, στην εκκίνηση του προγράμματος ο χρήστης πρέπει να δώσει τον αριθμό N δεδομένων στάθμης σήματος με τα οποία θέλει να εφαρμοστεί το παράδειγμα αυτό. Θα πάρουμε δεδομένη την τιμή του συντελεστή κλιμάκωσης $T=0.586$. Ακολούθως, η τιμή του n που είναι μια απλή μεταβλητή θα πάρει όλες τις τιμές από 0 έως $N-1$ με βήμα 1. Ύστερα ο χρήστης θα είναι αυτός που θα δώσει τα δυο όρια της δειγματοληψίας έτσι ώστε να είναι ομοιόμορφη και να έχει συμμετρικό διάστημα. Γνωρίζοντας τις τιμές των n και t_{min} , t_{max} και με δεδομένο ότι $T=0.586$ δε θα δυσκολευτούμε να υπολογίσουμε τις τιμές των C_n , t_n , και f_{in} . Όταν το $n=0$, η τιμή του a είναι $a=1/(\sqrt{2})$, ενώ όταν η τιμή του n είναι μεγαλύτερη του 0, τότε $a=1$. Ο τύπος που μας δίνει τη δειγματοληψία στο συμμετρικό διάστημα είναι $t_k=t_{min}+k*D_t$ με $D_t=(2*t_{max})/(N-1)$. Υπολογίζοντας και εδώ, μέσα από τη FOR, κάθε φορά την τιμή του t_k και μέσα από την ακόλουθη σχέση:

$$S=S+ (\exp (-\text{abs} (t_k)) + (0.1*(\text{random}-0.5))) * (\cos ((n*\pi*(k+ (1/2)))/N));$$

μπορώ να υπολογίζω την τιμή του συντελεστή μετασχηματισμού C_n για το DCT την τιμή των διαφόρων χρόνων δειγματοληψίας $t_n=t_i/T$ και την τιμή του δειγματοληφθέντος σήματος για κάθε t_n , που την ονομάζω f_{in} . Η διαφορά

αυτού του προγράμματος από τα δυο πρώτα είναι ότι εδώ θα πρέπει να γίνεται έλεγχος της τιμής που θα δίνουμε στο random. Το random είναι κάποια **τυχαία** νούμερα ανάμεσα στο 0 και στο 1. Σε περίπτωση που δώσουμε τιμή για το random μεγαλύτερη από το όριο αυτό θα ζητηθεί από το χρήστη να δώσει ένα άλλο νούμερο μέσα στο όριο. Με το παράδειγμα αυτό φαίνεται η διαφορά των δυο μεθόδων ακόμα και όταν δίνουμε τυχαία σήματα. Ακολουθεί το πρόγραμμα της συμπίεσης και αμέσως από κάτω βρίσκεται και το πρόγραμμα της αποσυμπίεσης.

```
function random
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
random=input('dose timi gia to random:');
Dt=(2*tmax)/(N-1);
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
if 0<random
if random<1
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
else random=input('dose timi megaliteri apo 0 kai mikroteri tou 1 gia to
random:')
end
end
end
```

```

Cn=((2*a)/N)*S
tn=tmin+n*Dt;
fend=(exp(-abs(tn)))+0.1*(random-0.5);
end

```

Έχοντας τα δυο προγράμματα το ένα κάτω από το άλλο όπως και τα αποτελέσματά τους έχουμε τη δυνατότητα να κάνουμε μια καλή σύγκριση ανάμεσα τους.

```

function aporandom
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
S1=0;
l=0;
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
random=input('dose timi gia to random:');
Dt=(2*tmax)/(N-1);
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end

S=0;
for k=1:1:N
if 0<random
if random<1
tk=tmin+k*Dt;
S=S+(exp(-abs(tk)))+(0.1*(random-0.5))*(cos((n*pi*(k+(1/2)))/N));

```

```

else random=input('dose timi megaliteri apo 0 kai mikroteri tou 1 gia to
random:')
end
end
end
tn=tmin+n*Dt
C(n+1)=((2*a)/N)*S;
end

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));

end
fin=S;

end

```

Στο προηγούμενο παράδειγμα η εξίσωση που εξετάσαμε ήταν εκθετικής μορφής. Στο συγκεκριμένο παράδειγμα έχουμε και πάλι την εκθετική μορφή της προηγούμενης εξίσωσης αλλά εδώ υπάρχει και το γινόμενο: $0.1*(\text{random}-0.5)$. Από αυτό το γινόμενο, και κυρίως από το `random`, δίνονται τα τυχαία σήματα. Ανάλογα με την τιμή που δίνουμε για το `random` παίρνουμε κάθε φορά και ένα διαφορετικό σήμα.

```

function PRD_DCT_RANDOM_1_1
N=input ('dose timi gia to N:');
T=0.586;
n=0:1:(N-1);
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
random=input('dose timi gia to random:');
Dt=(2*tmax)/(N-1);
S=0;
for k=1:1:N
if 0<random
if random<1
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
else random=input('dose timi megaliteri apo 0 kai mikroteri tou 1 gia to
random:')
end
end
end
for n=0:1:(N-1)
tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1))))+0.1*(random-0.5);
end

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;

```

```

for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end

SUM1=0;
SUM2=0;
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end
PRD=100*((SUM1/SUM2)^0.5)

subplot(2,1,1); plot(tn,fin);title('Original Waveform')
subplot(2,1,2); plot(tn,fout);title('DCT RANDOM COMPRESSION 1:1')

```



```

function PRD_DCT_RANDOM_3_4_1
T=0.586;
N=input ('dose timi gia to N:');
n=0:1:(N-1);
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
random=input('dose timi gia to random:');
Dt=(2*tmax)/(N-1);
S=0;
for k=1:1:N
if 0<random
if random<1
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
else random=input('dose timi megaliteri apo 0 kai mikroteri tou 1 gia to
random:');
end
end
end

for n=0:1:(N-1)
tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1)))+(0.1*(random-0.5)));
end

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end

```

```

S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end

```

```

for l=11:1:15
C(l+1)=0;
end

```

```

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end

```

```

SUM1=0;
SUM2=0;
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end

```

```

PRD=100*((SUM1/SUM2)^0.5)
subplot(2,1,1); plot(tn,fin);title('Original Waveform')
subplot(2,1,2); plot(tn,fout);title('DCT RANDOM COMPRESSION 3:4:1')

function PRD_DCT_RANDOM_2_1
T=0.586;
N=input ('dose timi gia to N:');
n=0:1:(N-1);
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input ('dose timi gia to tmin:');
tmax=input ('dose timi gia to tmax:');
random=input('dose timi gia to random:');
Dt=(2*tmax)/(N-1);
S=0;
for k=1:1:N
if 0<random
if random<1
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
else random=input('dose timi megaliteri apo 0 kai mikroteri tou 1 gia to
random:')
end
end
end

for n=0:1:(N-1)
tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1)))+(0.1*(random-0.5)));

end

```

```

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end

```

```

for l=7:1:15
C(l+1)=0;
end

```

```

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end

```

```

SUM1=0;
SUM2=0;
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end

PRD=100*((SUM1/SUM2)^0.5)

subplot(2,1,1); plot(tn,fin);title('Original Waveform')
subplot(2,1,2); plot(tn,fout);title('DCT RANDOM COMPRESSION 2:1')

function PRD_DCT_RANDOM_4_1
T=0.586;
N=input('dose timi gia to N:');
n=0:1:(N-1);
fin=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
fout=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
tmin=input('dose timi gia to tmin:');
tmax=input('dose timi gia to tmax:');
random=input('dose timi gia to random:');
Dt=(2*tmax)/(N-1);
S=0;
for k=1:1:N
if 0<random
if random<1
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N)));
else random=input('dose timi megaliteri apo 0 kai mikroteri tou 1 gia to
random:')
end
end
end
end

```

```

for n=0:1:(N-1)
tn(n+1)=tmin+n*Dt;
fin(n+1)=(exp(-abs(tn(n+1)))+(0.1*(random-0.5)));

```

```

end

```

```

for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=0;
for k=1:1:N
tk=tmin+k*Dt;
S=S+(exp(-abs(tk))+(0.1*(random-0.5)))*(cos((n*pi*(k+(1/2)))/N));
end
C(n+1)=((2*a)/N)*S;
end

```

```

for l=4:1:15
C(l+1)=0;
end

```

```

for l=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end

```

```

S=S+a*C(n+1)*(cos((n*pi*(l+1/2))/N));
end
S;
fout(l+1)=S;
end

SUM1=0;
SUM2=0;
for k=0:1:(N-1)
SUM1=SUM1+((fin(k+1)-fout(k+1))^2);
SUM2=SUM2+(fin(k+1)^2);
end

PRD=100*((SUM1/SUM2)^0.5)

subplot(2,1,1); plot(tn,fin);title('Original Waveform')
subplot(2,1,2); plot(tn,fout);title('DCT RANDOM COMPRESSION 4:1')

```

Τα παραπάνω 4 προγράμματα από τα οποία υπολογίζουμε το σφάλμα PRD είναι όμοια μεταξύ τους. Καταρχάς, να σημειωθεί ότι το πρόγραμμα έχει άμεση επαφή με το χρήστη. Αυτό το διαπιστώνουμε από το γεγονός ότι και πάλι ζητείται από το χρήστη να δίνει διάφορες τιμές στις μεταβλητές για να πάρει αποτελέσματα. Έτσι, στην αρχή το πρόγραμμα θέλει από το χρήστη να δώσει τον αριθμό των δεδομένων στάθμης σήματος N με τα οποία επιθυμεί να εφαρμοστεί το συγκεκριμένο παράδειγμα. Η τιμή του συντελεστή κλιμάκωσης είναι μετά από υπολογισμούς ίση με $T=0.586$. Στη συνέχεια το n το οποίο είναι μια απλή μεταβλητή παίρνει όλες τις τιμές από 0 έως N-1 με βήμα 1. Αμέσως μετά ορίζουμε τους πίνακες fin και fout, ο πρώτος μου δίνει το κανονικό σήμα συμπιεσμένο και ο δεύτερος το αποσυμπιεσμένο σήμα κάθε φορά, που θα μας βοηθήσουν να υπολογίσουμε το σφάλμα. Ο πίνακας Cout μου δίνει όλες τις τιμές του συντελεστή μετασχηματισμού από τις οποίες στην πορεία θα βγάλω τον πίνακα της αποσυμπίεσης. Κατόπιν αυτού ο χρήστης πρέπει να δώσει τα δυο όρια της δειγματοληψίας έτσι ώστε να είναι ομοιόμορφη και να έχει συμμετρικό διάστημα. Μετά από αυτά ελέγχεται το n

το οποίο αν είναι $n=0$ τότε η τιμή του a είναι $a=1/\sqrt{2}$, ενώ για κάθε τιμή του n μεγαλύτερη του 0, $a=1$. Ο τύπος που μας δίνει τη δειγματοληψία στο συμμετρικό διάστημα είναι $t_k=t_{\min}+k \cdot Dt$ με $Dt=(2 \cdot t_{\max})/(N-1)$. Υπολογίζοντας και πάλι μέσα στη FOR για όλες τις τιμές του N κάθε φορά την τιμή του t_k και μέσα από την ακόλουθη σχέση:

$$S=S+(\exp(-\text{abs}(t_k))+0.1 \cdot (\text{random}-0.5)) \cdot (\cos((n \cdot \pi \cdot (k+(1/2)))/N)));$$

υπολογίζω τον πίνακα της αποσυμπίεσης. Ύστερα από τον πίνακα αυτόν και από τον πίνακα της συμπίεσης πάω να βρω το σφάλμα για κάθε μια από τις 4 περιπτώσεις. Ορίζω δυο αθροίσματα στην αρχή να είναι μηδέν και μέσα από μια FOR τα υπολογίζω με βάση βέβαια και εδώ τους πίνακες `fin` και `fout`. Τέλος βρίσκω πόσο είναι το PRD από την ακόλουθη σχέση:

$$PRD=100 \cdot ((SUM1/SUM2)^{0.5})$$

Επίσης είναι ανάγκη να ειπωθεί και ο τρόπος με τον οποίο διαμορφώνεται η κάθε κατηγορία συμπίεσης. Στη συμπίεση 2:1 τους πρώτους 8 συντελεστές μετασχηματισμού τους κρατάμε με την τιμή που έχουν ενώ τους άλλους 8 τους μηδενίζουμε. Με τον ίδιο τρόπο στη συμπίεση 4:1 κρατάμε τους 4 πρώτους συντελεστές και μηδενίζουμε τους υπόλοιπους. Όμοια συμβαίνει και για τη συμπίεση (3/4):1. Είναι σαφές λοιπόν ότι στο παράδειγμα αυτό τα δεδομένα είναι ίδια με το προηγούμενο παράδειγμα. Μόνο που στο παράδειγμα αυτό επειδή έχουμε **τυχαία** σήματα για αυτό πρέπει να αναφερθεί πως η τιμή που δίνεται στο `random` είναι 0.5.

l	Cn -DOGHT	Cn -DCT
0	0.259	0.2588
1	0.00	-0.0013
2	-0.273	-0.2732
3	-0.00	-0.0012
4	0.149	0.1487
5	-0.00	-0.0011
6	-0.084	-0.0840
7	0.00	0
8	0.049	0.0483
9	0.00	0
10	-0.029	-0.0296
11	0.00	0
12	0.017	0.0165
13	0.00	0
14	-0.008	-0.0078
15	0.00	0

l	$tn=ti/T$ -DOGHT	$tn=ti/T$ -DCT
0	-5	-5
1	-4.3333	-4.3333
2	-3.6667	-3.6667
3	-3	-3
4	-2.3333	-2.3333
5	-1.6667	-1.6667
6	-1	-1
7	-0.3333	-0.3333
8	0.3333	0.3333
9	1	1
10	1.6667	1.6667
11	2.3333	2.3333
12	3	3
13	3.6667	3.6667
14	4.3333	4.3333
15	5	5

l	fin=f(ti/T)-DOGHT	fin=f(ti/T)-DCT
0	0.007	0.0067
1	-0.013	0.0131
2	0.026	0.0256
3	0.050	0.0498
4	0.097	0.0970
5	0.189	0.1889
6	0.368	0.3679
7	0.717	0.7165
8	0.717	0.7165
9	0.368	0.3679
10	0.189	0.1889
11	0.097	0.0970
12	0.050	0.0498
13	0.026	0.0256
14	0.013	0.0131
15	0.007	0.0067

l	ffin=f(ti/T)-DOGHT	ffin=f(ti/T)-DCT
0	0.007	0.0067
1	-0.013	0.0131
2	0.026	0.0256
3	0.050	0.0498
4	0.097	0.0970
5	0.189	0.1889
6	0.368	0.3679
7	0.717	0.7165
8	0.717	0.7165
9	0.368	0.3679
10	0.189	0.1889
11	0.097	0.0970
12	0.050	0.0498
13	0.026	0.0256
14	0.013	0.0131
15	0.007	0.0102

Πρέπει να γνωρίζουμε πως οι τιμές στους παραπάνω πίνακες είναι για την περίπτωση της συμπίεσης 1:1. Πρέπει να τονιστεί πως για τα αποτελέσματα της στάθμης σήματος και του αντίστροφου μετασχηματισμού η τιμή του random είναι ίση με random=0.5. Φαίνεται από τα τελικά αποτελέσματα πως για την περίπτωση της συμπίεσης 1:1 οι τιμές του συμπιεσμένου και αποσυμπιεσμένου σήματος είναι κοινές. Αυτό συμβαίνει γιατί με τη μέθοδο του DCT σε συμπίεση 1:1 έχω σχεδόν μηδενικό σφάλμα κάτι το οποίο δεν ισχύει για τον DOGHT.

N	PRD-DOGHT (%)	PRD-DCT (%)
16	3.57	0.64
12	7.54	4.44
8	10.54	14.29
4	16.75	43.37

Από τα αποτελέσματα του πίνακα όσον αφορά το PRD φαίνεται και σε αυτό το παράδειγμα πως για συμπίεσεις μικρής μορφής δηλαδή 1:1 και (3/4):1 ο DCT είναι καλύτερος. Εκεί όμως που κερδίζει η μέθοδος του DOGHT είναι σε συμπίεσεις μεγαλύτερης μορφής, δηλαδή 2:1, 4:1, όπου τα ποσοστά του σφάλματος για τον DCT ανεβαίνουν κατακόρυφα σε αντίθεση με αυτόν.

Συνοψίζοντας, λοιπόν, με βάση όλα τα απλά παραδείγματα που έχουν εκτεθεί παραπάνω, καταλήγουμε εύκολα στο συμπέρασμα ότι ο μετασχηματισμός DOGHT διαθέτει καλύτερης ποιότητας συμπίεση σε σύγκριση με το μετασχηματισμό DCT όσον αφορά μία μεγάλη κατηγορία σημάτων, η οποία περιλαμβάνει τόσο ντετερμινιστικά όσο και τυχαία σήματα. Έτσι, με τον τρόπο αυτό είναι δυνατή η πρόβλεψη πως η χρήση του μετασχηματισμού DOGHT θα έχει ως αποτέλεσμα μια σημαντική αύξηση των λόγων συμπίεσης των σημάτων ήχου και μάλιστα διαθέτοντας και υπολογιστική πολυπλοκότητα, η οποία θυμίζει την πολυπλοκότητα του μετασχηματισμού DCT. [7,8,9]

4.9 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΗΧΟΥ

Η λεπτομερής ανάλυση που έγινε προηγουμένως αφορούσε προγράμματα με συγκεκριμένη μορφή εξίσωσης. Στη συνέχεια ακολουθεί ανάλυση προγραμμάτων τα οποία έχουν σαν είσοδο έναν ήχο. Έτσι λοιπόν το σήμα εισόδου τώρα θα είναι ένα αρχείο ήχου σε αντίθεση με πριν που ήταν μια συγκεκριμένη εξίσωση. Το πρόγραμμα που ακολουθεί έχει σαν είσοδο τον ήχο <<chimes>>.

Στο πρόγραμμα που ακολουθεί θα φανεί η ποιοτική διαφορά ενός ήχου κανονικής μορφής σε σχέση με ένα συμπιεσμένο ήχο για τον DCT μετασχηματισμό. Αρχικά ζητείται από το χρήστη να δώσει το δείγμα των αριθμών στους οποίους θα εφαρμοστεί το πρόγραμμα. Αυτό γίνεται μέσα από την τιμή που θα δώσουμε στη μεταβλητή N. Αντίστοιχα η τιμή που θα δώσει ο χρήστης στη μεταβλητή r μας βοηθάει να καταλάβουμε τι μορφή συμπίεσης ζητά ο χρήστης αφού για να το δούμε αυτό κάνουμε διαίρεση N/r. Στη συνέχεια δίνουμε εντολή να ακουστεί στο πρόγραμμα ο κανονικός ήχος. Ο ήχος που θα ακουστεί είναι επίσης επιλογή του χρήστη. Στο συγκεκριμένο παράδειγμα είναι ο ήχος <<chimes>>. Ύστερα βρίσκουμε το μήκος του αρχείου ήχου με την εντολή size και το χωρίζουμε σε N μέρη έτσι ώστε να το

χωρίσουμε σε όσο το δυνατόν μικρότερα μέρη μπορούμε για να κάνουμε τη συμπίεση. Μετά μέσα από τη FOR και αφού ελέγξουμε την τιμή του n υπολογίζουμε τον συντελεστή μετασχηματισμού Cn για το DCT. Με τη βοήθεια του συντελεστή μετασχηματισμού παίρνω αποτέλεσμα και για την έξοδο fout. Τέλος υπάρχει ακόμη μια FOR που δίνει την τιμή του σφάλματος PRD για κάθε είδος συμπίεσης. Πρώτα βγάζει το κάθε ένα σφάλμα χωριστά και ύστερα βγάζει το μέσο όρο τους. Μετά από όλη τη προηγούμενη διαδικασία ακούμε το συμπιεσμένο ήχο και βλέπουμε το αποτέλεσμα και σε γραφική παράσταση. Η συμπίεση που έγινε για να φαίνεται και η διαφορά ποιότητας είναι 1:1, (3/4):1, 2:1, 4:1, 8:1 και 16:1.

```
function chimes
N=input('dose timi gia to N:');
r=input('dose timi gia to r:');
Cn=[];
at=[];
audio=[];

[fin, f] = wavread('chimes.wav');
disp('Original')
wavplay(fin,f)

b=size(fin,1);
c=fix(b/N);

for j=1:c

    at=fin((N*(j-1)+1):j*N);

    for n=0:1:(N-1)
        if n==0
            a=1/(sqrt(2));
        elseif n>0
            a=1;
        end
        Cn(n+1)=0;
        for k=1:1:r
            Cn(n+1)=Cn(n+1)+(at(k))*(cos((n*pi*(k+(1/2)))/N));
        end
        Cn(n+1)=((2*a)/N)*Cn(n+1);
    end
for x=0:1:(N-1)
    S=0;
    for n=0:1:(N-1)
```

```

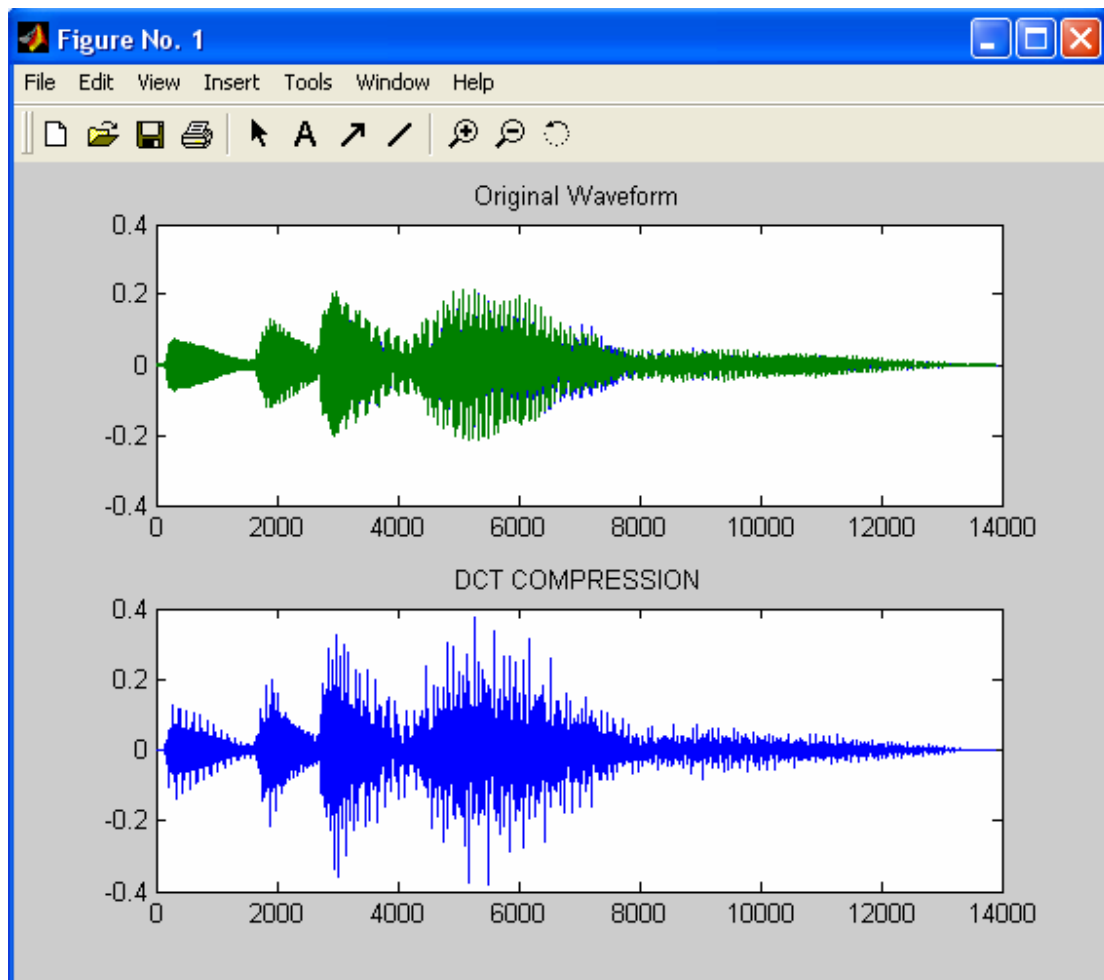
    if n==0
        a=1/(sqrt(2));
    elseif n>0
        a=1;
    end
    S=S+a*Cn(n+1)*(cos((n*pi*(x+1/2))/N));
end
fout(x+1)=S;
end
audio(1,(N*(j-1)+1):j*N)=fout;
end
disp('Compression Factor 2')
wavplay(audio, f)
subplot(2,1,1);plot(fin);title('Original Waveform')
subplot(2,1,2);plot(audio);title('DCT COMPRESSION')

```

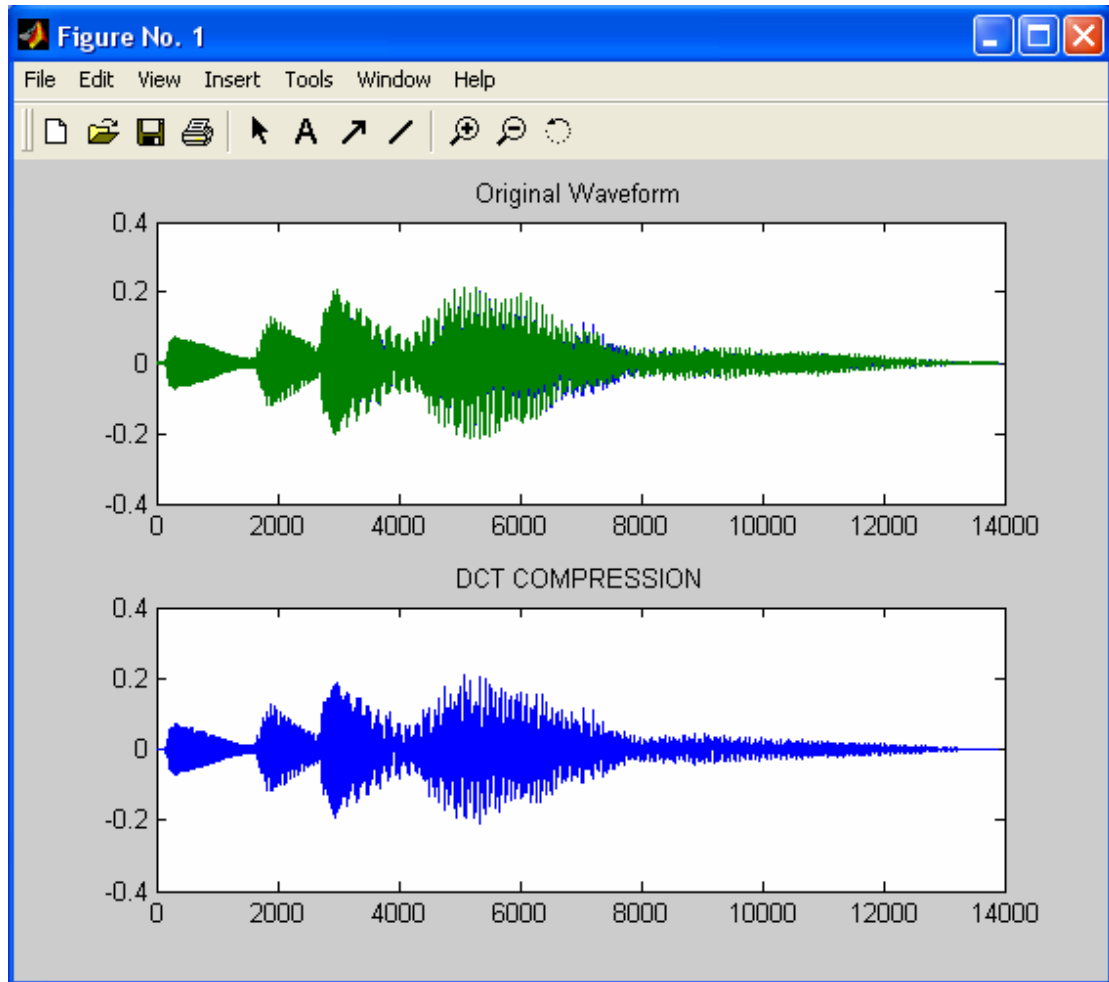
```

>> chimes
dose timi gia to N:16
dose timi gia to r:16
Original
Compression Factor 2

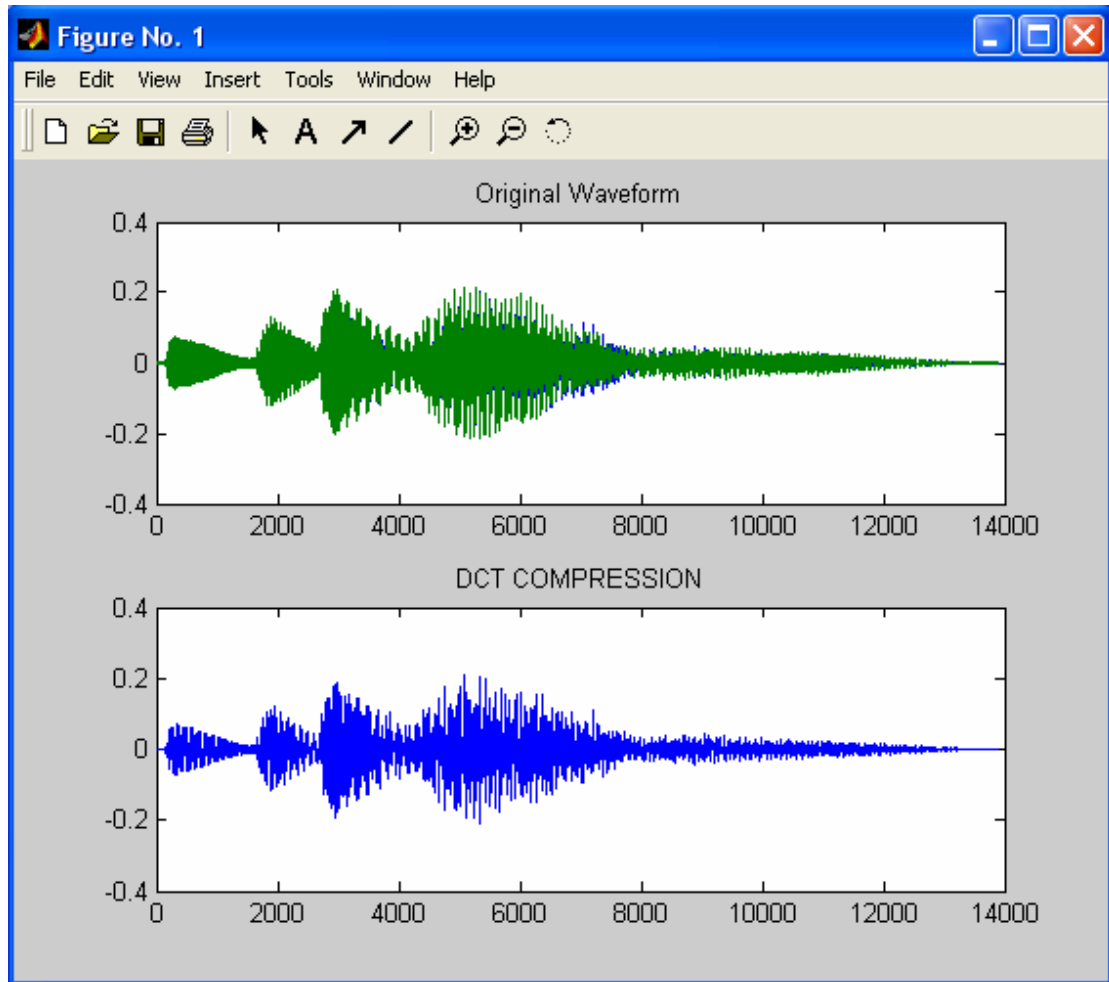
```



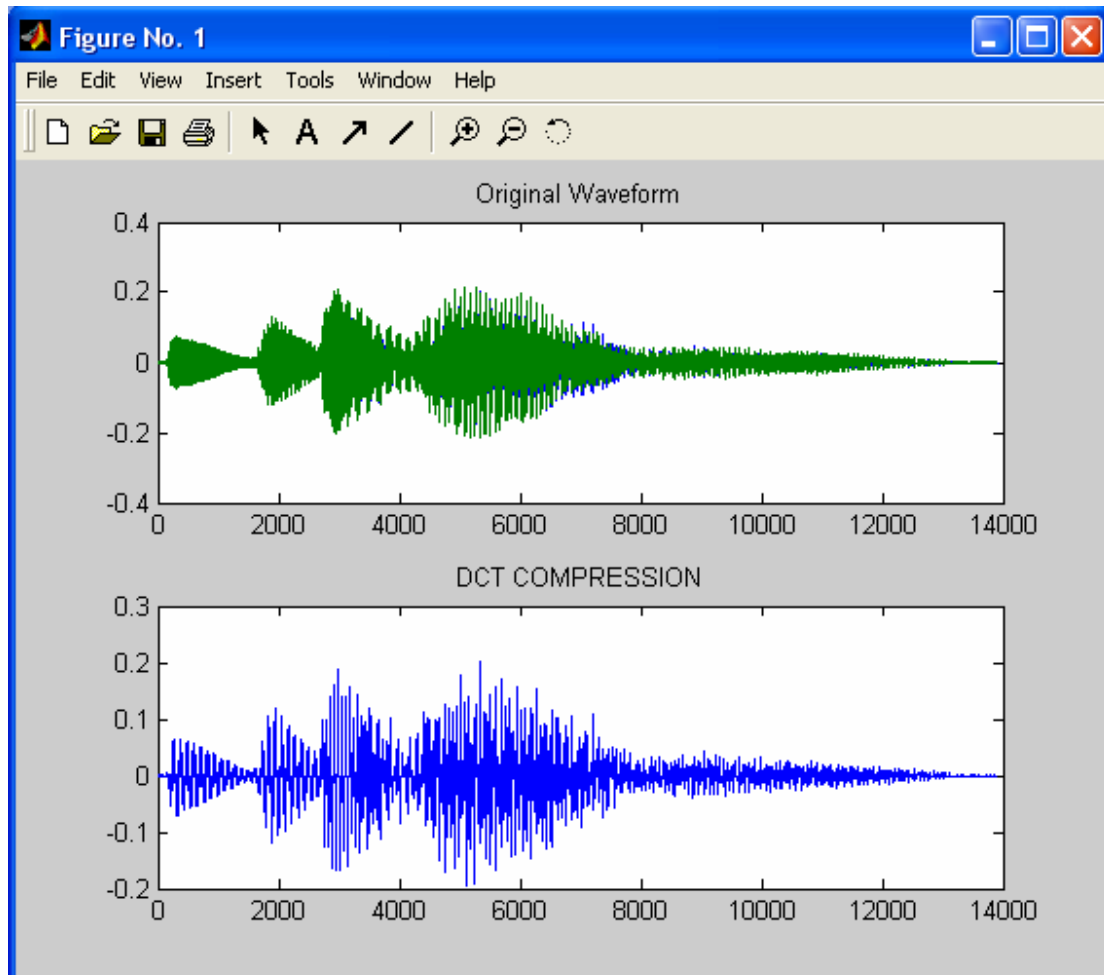
```
>> chimes
dose timi gia to N:16
dose timi gia to r:12
Original
Compression Factor 2
```



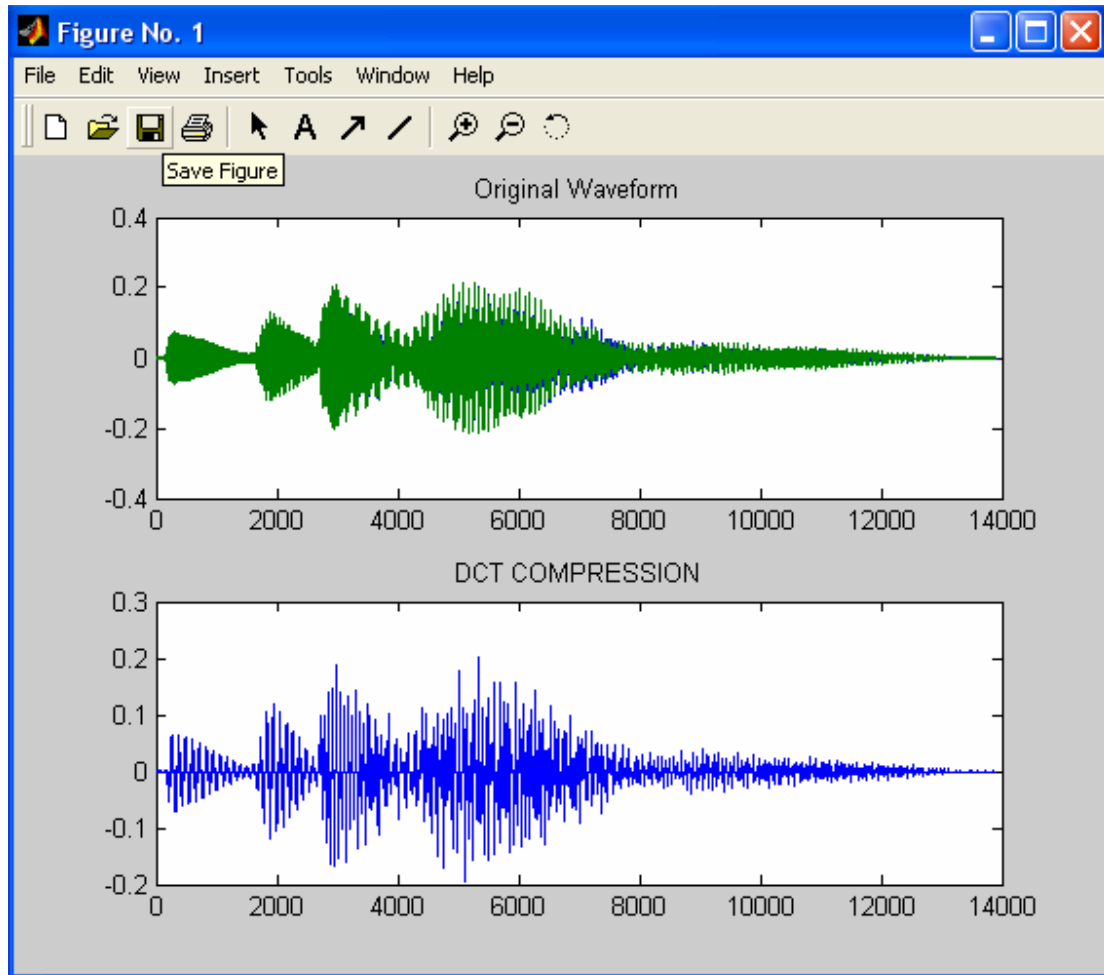

```
>> chimes
dose timi gia to N:16
dose timi gia to r:8
Original
Compression Factor 2
```



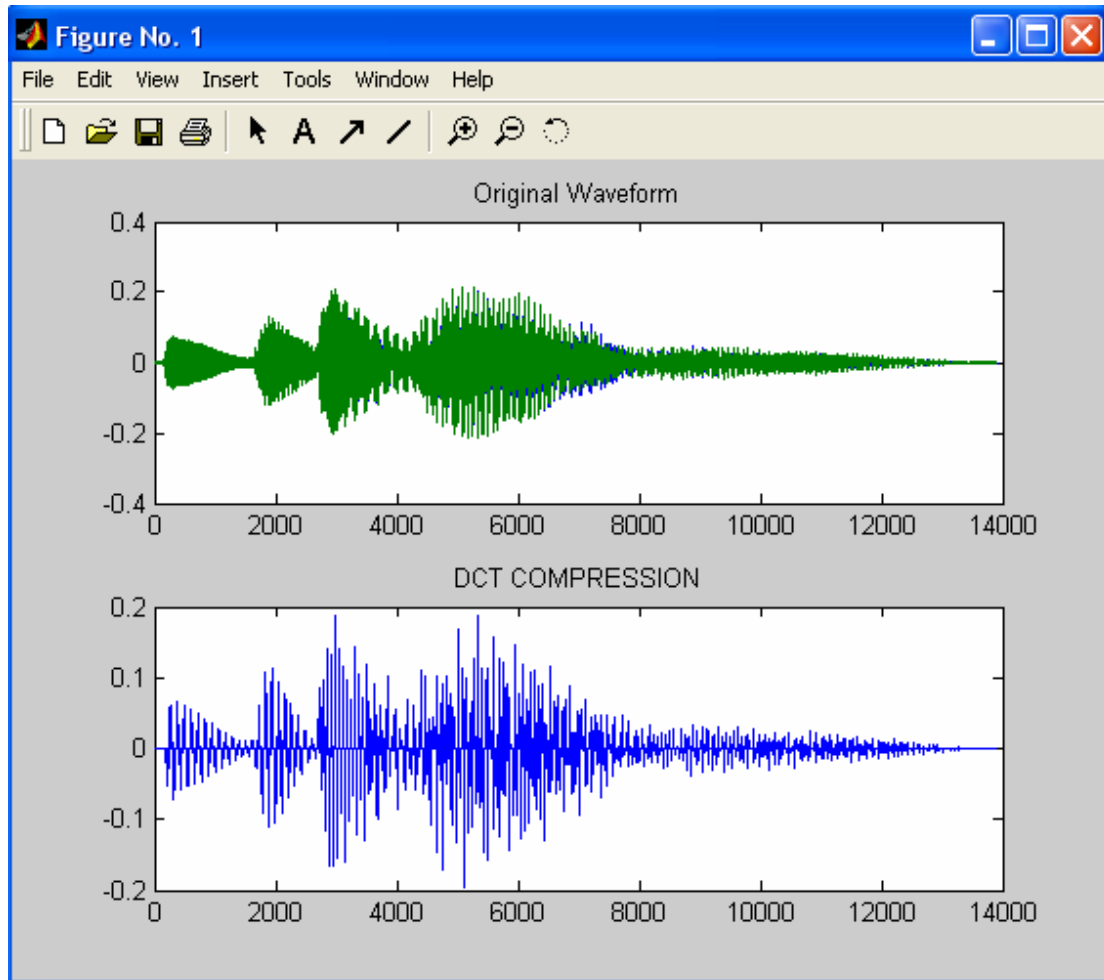
```
>> chimes
dose timi gia to N:16
dose timi gia to r:4
Original
Compression Factor 2
```



```
>> chimes
dose timi gia to N:16
dose timi gia to r:2
Original
Compression Factor 2
```



```
>> chimes
dose timi gia to N:16
dose timi gia to r:1
Original
Compression Factor 2
```



Μέσα από αυτό το παράδειγμα του ήχου παρατηρούμε εύκολα ακουστικά και από τις γραφικές παραστάσεις πως όσο μεγαλύτερη συμπίεση ζητάμε τόσο περισσότερο μειώνεται η ποιότητα του ήχου. Σε κάθε μια από τις γραφικές παραστάσεις φαίνεται πως στα άκρα αρχίζει το σήμα να χάνει ποιότητα και αυτό γιατί τα πάνω και κάτω όρια της γραφικής από το ένα ποσοστό συμπίεσης στο επόμενο αυξάνονται μεταξύ τους. Σε όλες τις γραφικές παραστάσεις το σήμα που είναι με πράσινο χρώμα είναι το κανονικό ενώ αυτό με το μπλε χρώμα είναι το συμπιεσμένο. Έτσι λοιπόν βλέπουμε ότι

σε όλες τις γραφικές παραστάσεις το κανονικό σήμα είναι συνεχώς το ίδιο σε αντίθεση με το συμπιεσμένο που ενώ στην αρχή, στη συμπίεση 1:1, είναι έντονο το μπλε χρώμα, στη συμπίεση 16:1 το έντονο μπλε έχει μειωθεί αισθητά.

Τα ίδια αποτελέσματα παρατηρούμε και στο επόμενο παράδειγμα που αφορά έναν άλλο ήχο. Σε αυτό το παράδειγμα θα χρησιμοποιήσουμε τον ήχο <<notify>>.

Στο πρόγραμμα αυτό ζητείται πρώτα από το χρήστη να δώσει τον αριθμό των δειγμάτων που θα χρησιμοποιηθούν. Αυτό πετυχαίνεται με την τιμή που θα δώσουμε στη μεταβλητή N. Όμοια η τιμή που θα δώσει ο χρήστης στη μεταβλητή r μας βοηθάει να καταλάβουμε τι μορφή συμπίεσης ζητά ο χρήστης. Αυτό γίνεται κάνοντας διαίρεση N/r. Ύστερα δίνεται εντολή από το πρόγραμμα να ακουστεί ο κανονικός ήχος. Ο ήχος που θα ακουστεί είναι επίσης επιλογή του χρήστη. Στο συγκεκριμένο παράδειγμα είναι ο ήχος <<notify>>. Μετά βρίσκουμε το μήκος του αρχείου ήχου με την εντολή size και το χωρίζουμε σε N μέρη έτσι ώστε να το χωρίσουμε σε όσο το δυνατόν μικρότερα μέρη μπορούμε για να κάνουμε τη συμπίεση. Στη συνέχεια μέσα από τη FOR υπολογίζουμε τον συντελεστή μετασχηματισμού Cn για το DCT αφού βεβαίως πρώτα ελέγξουμε την τιμή του n. Με τη βοήθεια του συντελεστή μετασχηματισμού έχουμε αποτέλεσμα και για την έξοδο fout. Τέλος υπάρχει ακόμη μια FOR που βγάζει σαν αποτέλεσμα την τιμή του σφάλματος PRD για κάθε είδος συμπίεσης. Πρώτα βγάζει το κάθε ένα σφάλμα χωριστά και ύστερα το μέσο όρο τους. Μετά από όλη τη προηγούμενη διαδικασία ακούμε το συμπιεσμένο ήχο και βλέπουμε το αποτέλεσμα και σε γραφική παράσταση. Η συμπίεση που έγινε για να φαίνεται και η διαφορά ποιότητας είναι 1:1, (3/4):1, 2:1, 4:1, 8:1 και 16:1.

```
function notify
N=input('dose timi gia to N:');
r=input('dose timi gia to r:');
Cn=[];
at=[];
audio=[];

[fin, f] = wavread('notify.wav');
disp('Original')
wavplay(fin,f)
```

```

b=size(fin,1);
c=fix(b/N);

for j=1:c

    at=fin((N*(j-1)+1):j*N);

    for n=0:1:(N-1)
        if n==0
            a=1/(sqrt(2));
        elseif n>0
            a=1;
        end
        Cn(n+1)=0;
        for k=1:1:r
            Cn(n+1)=Cn(n+1)+(at(k))*(cos((n*pi*(k+(1/2)))/N));
        end
        Cn(n+1)=((2*a)/N)*Cn(n+1);
    end

    for x=0:1:(N-1)
        S=0;
        for n=0:1:(N-1)
            if n==0
                a=1/(sqrt(2));
            elseif n>0
                a=1;
            end
            S=S+a*Cn(n+1)*(cos((n*pi*(x+1/2))/N));
        end
        fout(x+1)=S;
    end

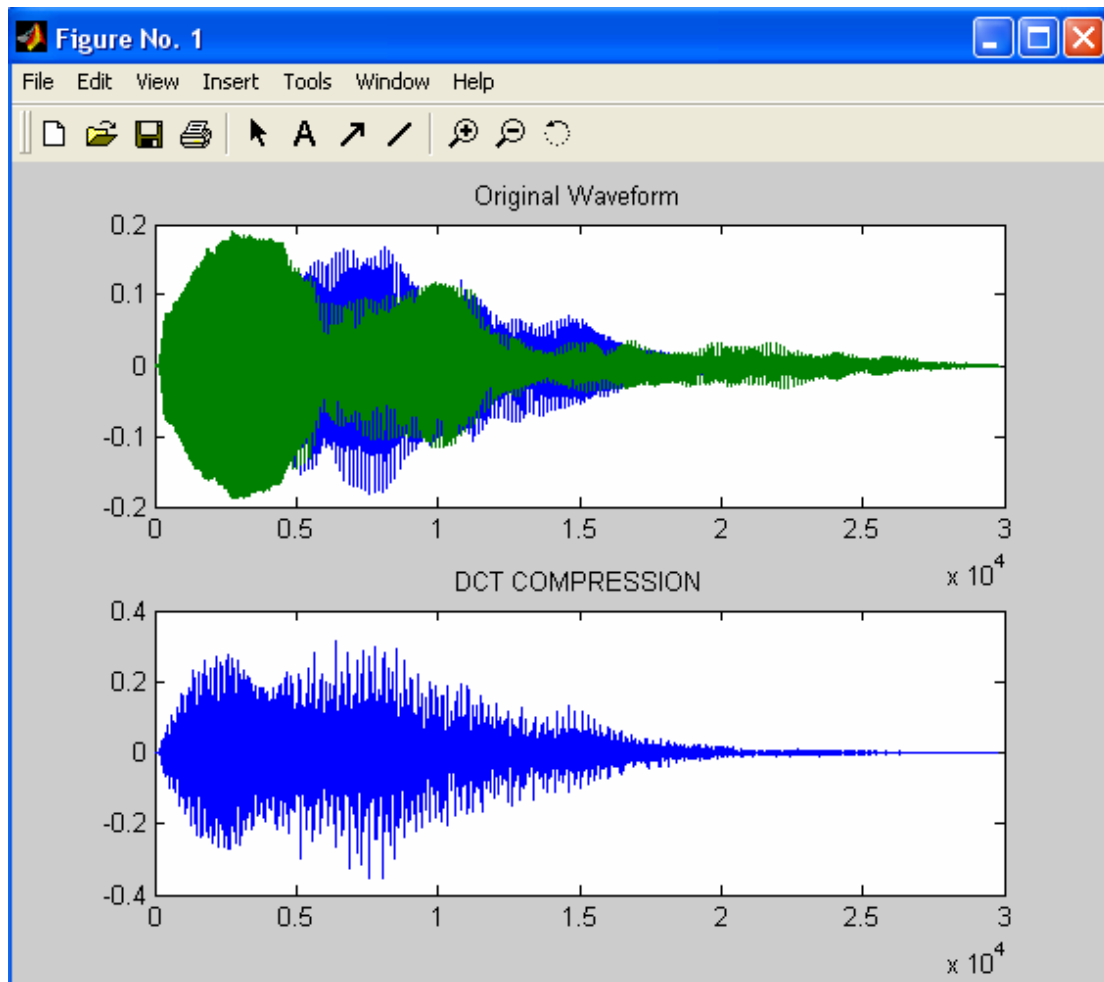
    audio(1,(N*(j-1)+1):j*N)=fout;
end

disp('Compression Factor 2')
wavplay(audio, f)

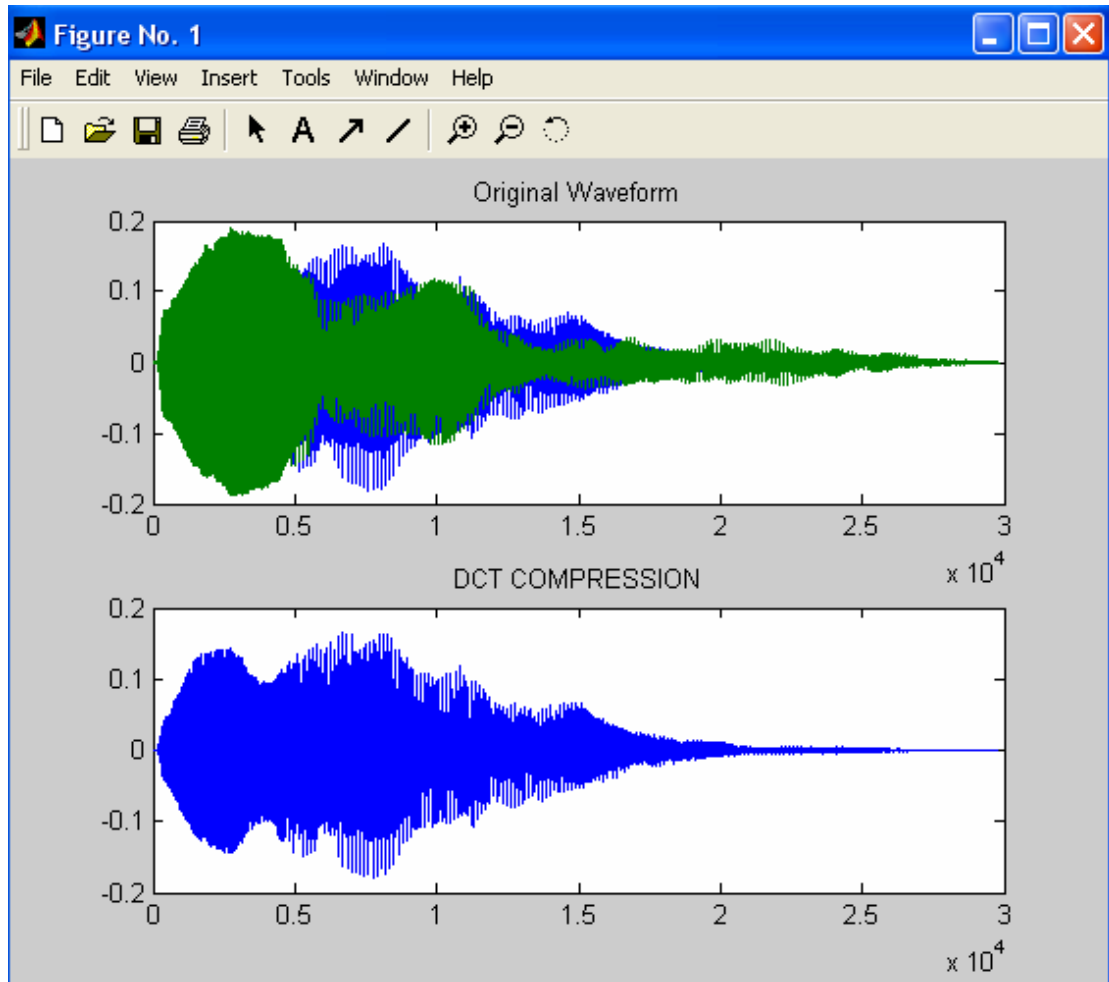
subplot(2,1,1);plot(fin);title('Original Waveform')
subplot(2,1,2);plot(audio);title('DCT COMPRESSION')

```

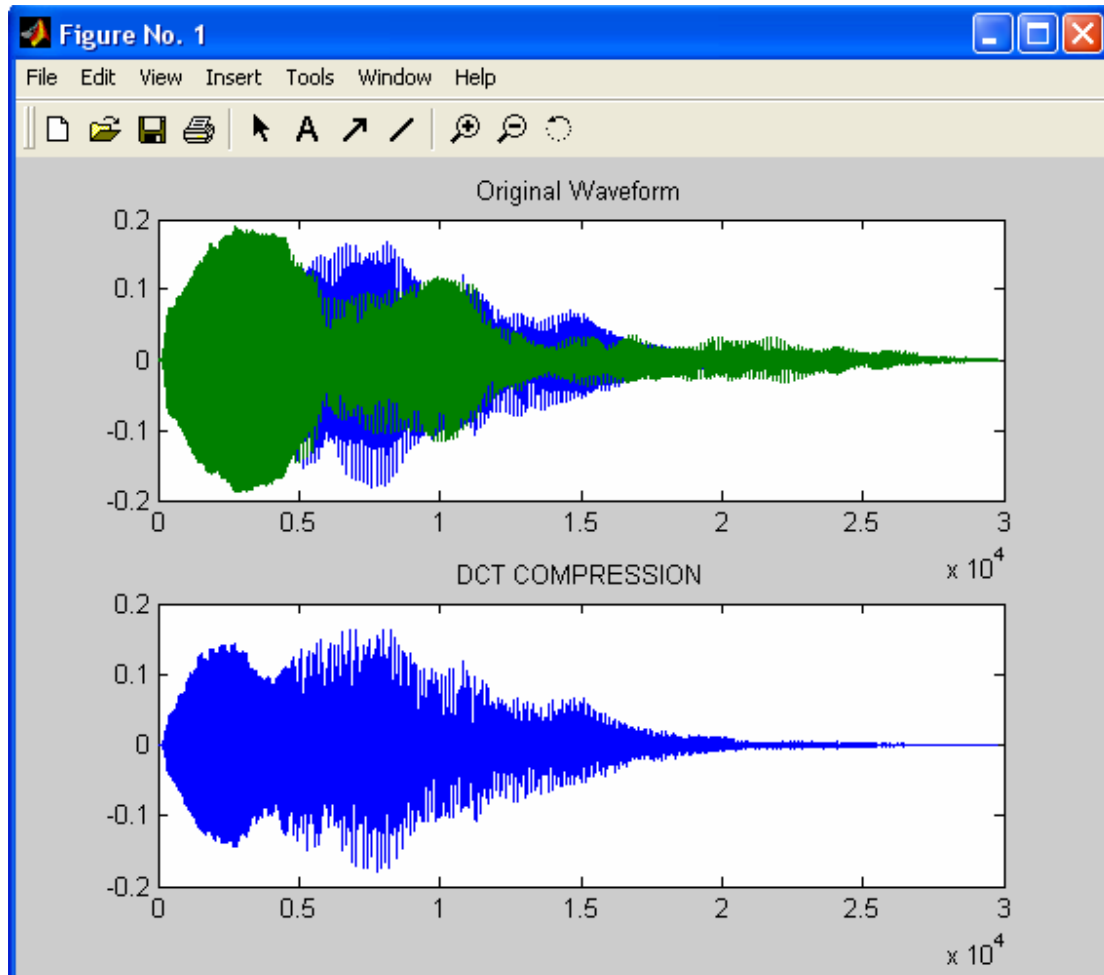
```
>> notify
dose timi gia to N:16
dose timi gia to r:16
Original
Compression Factor 2
```



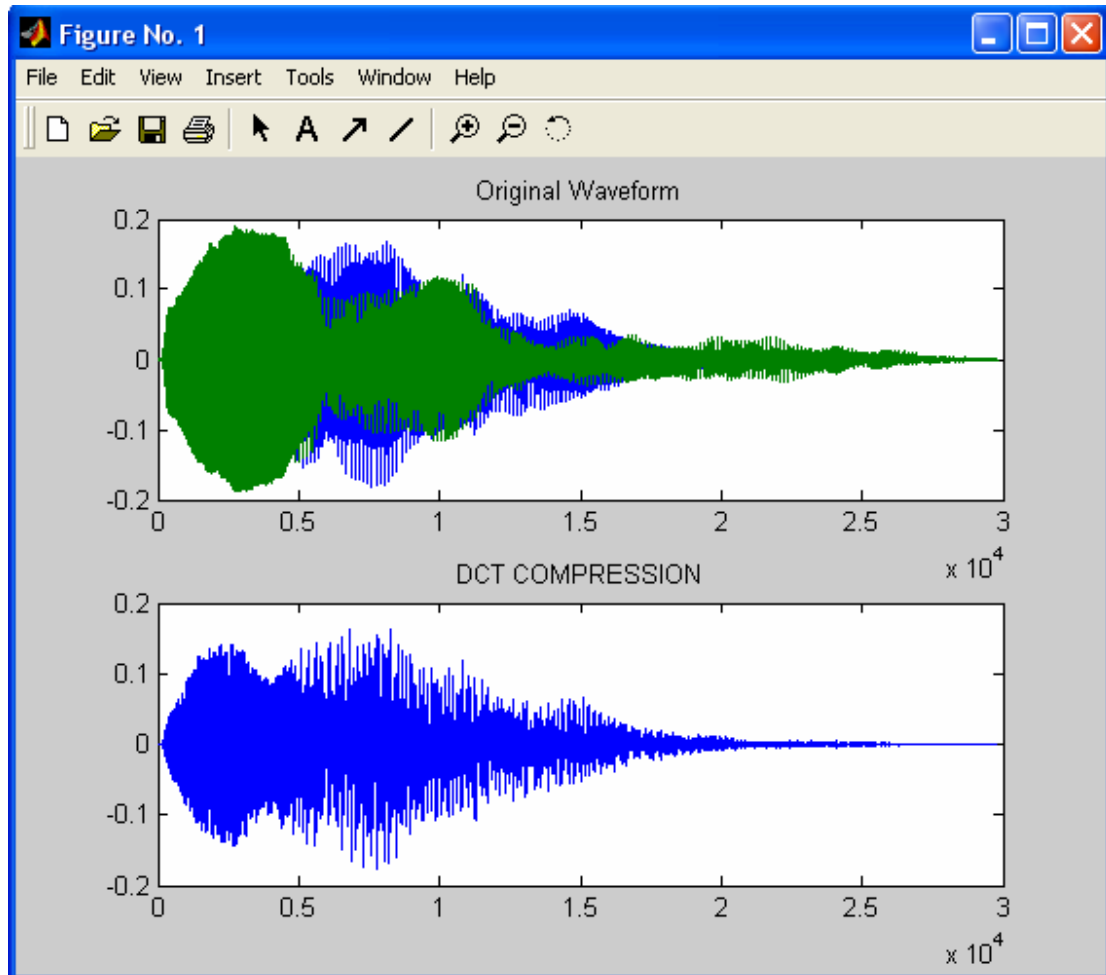
```
>> notify
dose timi gia to N:16
dose timi gia to r:12
Original
Compression Factor 2
```



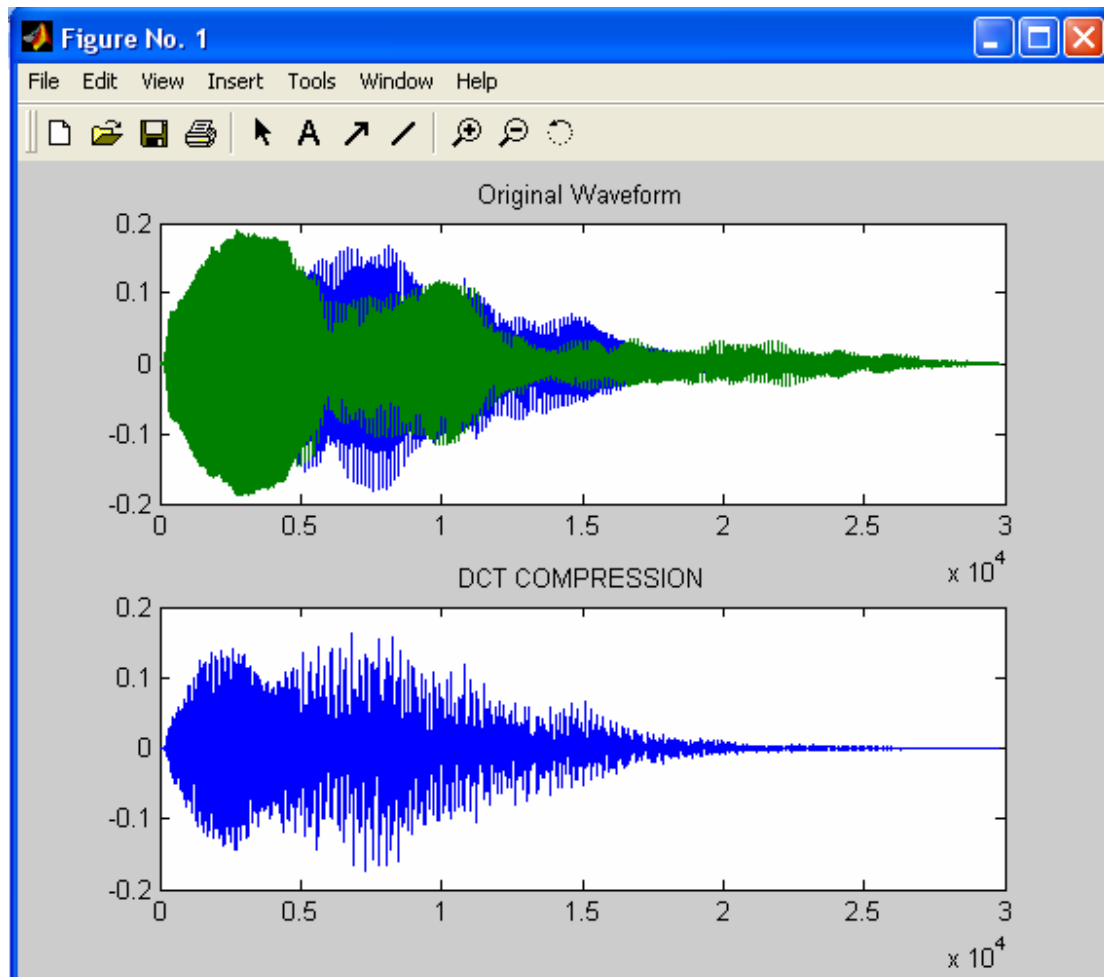
>> notify
dose timi gia to N:16
dose timi gia to r:8
Original
Compression Factor 2



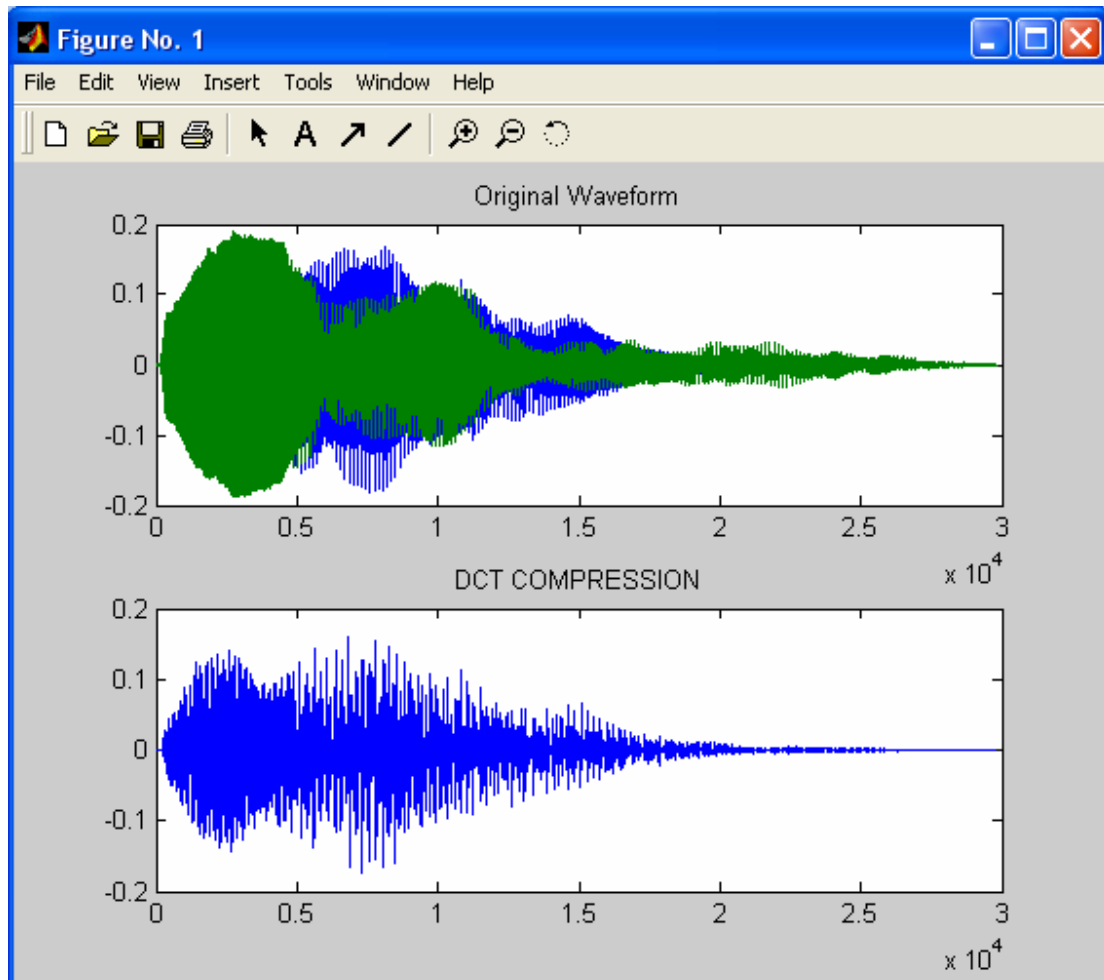
```
.>> notify  
dose timi gia to N:16  
dose timi gia to r:4  
Original  
Compression Factor 2
```



```
>> notify
dose timi gia to N:16
dose timi gia to r:2
Original
Compression Factor 2
```



>> notify
dose timi gia to N:16
dose timi gia to r:1
Original
Compression Factor 2



Όπως στο προηγούμενο παράδειγμα ήχου έτσι και σε αυτό παρατηρούμε εύκολα ακουστικά και από τις γραφικές παραστάσεις πως όσο μεγαλύτερη συμπίεση ζητάμε τόσο περισσότερο μειώνεται η ποιότητα του ήχου. Σε κάθε μια από τις γραφικές παραστάσεις φαίνεται πως στα άκρα αρχίζει το σήμα να χάνει ποιότητα. Σε όλες τις γραφικές παραστάσεις το σήμα που είναι με πράσινο χρώμα είναι το κανονικό ενώ αυτό με το μπλε χρώμα είναι το συμπιεσμένο. Έτσι λοιπόν βλέπουμε ότι σε όλες τις γραφικές παραστάσεις το κανονικό σήμα είναι το ίδιο. Αντίθετα το συμπιεσμένο σήμα

στην αρχή, δηλαδή στη συμπίεση 1:1, είναι έντονο το μπλε χρώμα αλλά όσο μεγαλώνει η συμπίεση με κορύφωση τη συμπίεση 16:1 το έντονο μπλε χρώμα έχει μειωθεί αισθητά.

Μόλις παραπάνω είδαμε τις διαφορές που υπάρχουν στα δυο προηγούμενα παραδείγματα που αφορούσαν έναν απλό ήχο μέσα από τις γραφικές παραστάσεις και από τα αποτελέσματα για το σφάλμα PRD. Στη συνέχεια θα αναλύσουμε ένα πρόγραμμα το οποίο ασχολείται με ένα ηχητικό κείμενο. Μέσα από το πρόγραμμα αυτό θα φανεί ότι και σε αυτή την περίπτωση ισχύουν τα ίδια δεδομένα με πριν που μιλήσαμε για έναν απλό ήχο.

Στο πρόγραμμα αυτό ζητείται αρχικά από το χρήστη να δώσει τιμές για τις μεταβλητές N και r που αφορούν τον αριθμό των δειγμάτων που θα χρησιμοποιηθούν και την κατανόηση της μορφής συμπίεσης που ζητά ο χρήστης αντίστοιχα. Αυτό πετυχαίνεται με τη διαίρεση N/r . Ύστερα δίνεται εντολή στο πρόγραμμα να ακουστεί το ηχητικό κείμενο. Το ηχητικό κείμενο που ακούγεται κάθε φορά είναι επιλογή του χρήστη. Στο συγκεκριμένο παράδειγμα είναι το ηχητικό κείμενο <<christos>>. Ύστερα βρίσκουμε το μήκος του αρχείου ήχου με την εντολή `size` και το χωρίζουμε σε N μέρη έτσι ώστε να το χωρίσουμε σε όσο το δυνατόν μικρότερα μέρη μπορούμε για να κάνουμε την καλύτερη δυνατή συμπίεση. Στη συνέχεια μέσα από τη FOR υπολογίζουμε τον συντελεστή μετασχηματισμού C_n για το DCT αφού βεβαίως πρώτα ελέγξουμε την τιμή του n . Το n σε αυτό το πρόγραμμα εάν πάρει την τιμή 0 τότε το a έχει την τιμή $(1/\sqrt{2})$. Αν το n όμως έχει τιμή μεγαλύτερη του μηδενός τότε το a θα έχει την τιμή 1. Με τη βοήθεια του συντελεστή μετασχηματισμού θα έχουμε αποτέλεσμα και για το `fout` που είναι η έξοδος. Τέλος υπάρχει ακόμη μια FOR που βγάζει σαν αποτέλεσμα την τιμή του σφάλματος PRD για κάθε είδος συμπίεσης. Πρώτα βγάζει το κάθε ένα σφάλμα χωριστά και ύστερα υπολογίζει το μέσο όρο όλων των σφαλμάτων. Μετά από όλη τη προηγούμενη διαδικασία ακούμε το συμπίεσμένο ήχο και βλέπουμε το αποτέλεσμα και σε γραφική παράσταση. Η συμπίεση που έγινε για να φαίνεται και η διαφορά ποιότητας είναι 1:1, (3/4):1, 2:1, 4:1, 8:1 και 16:1.

```

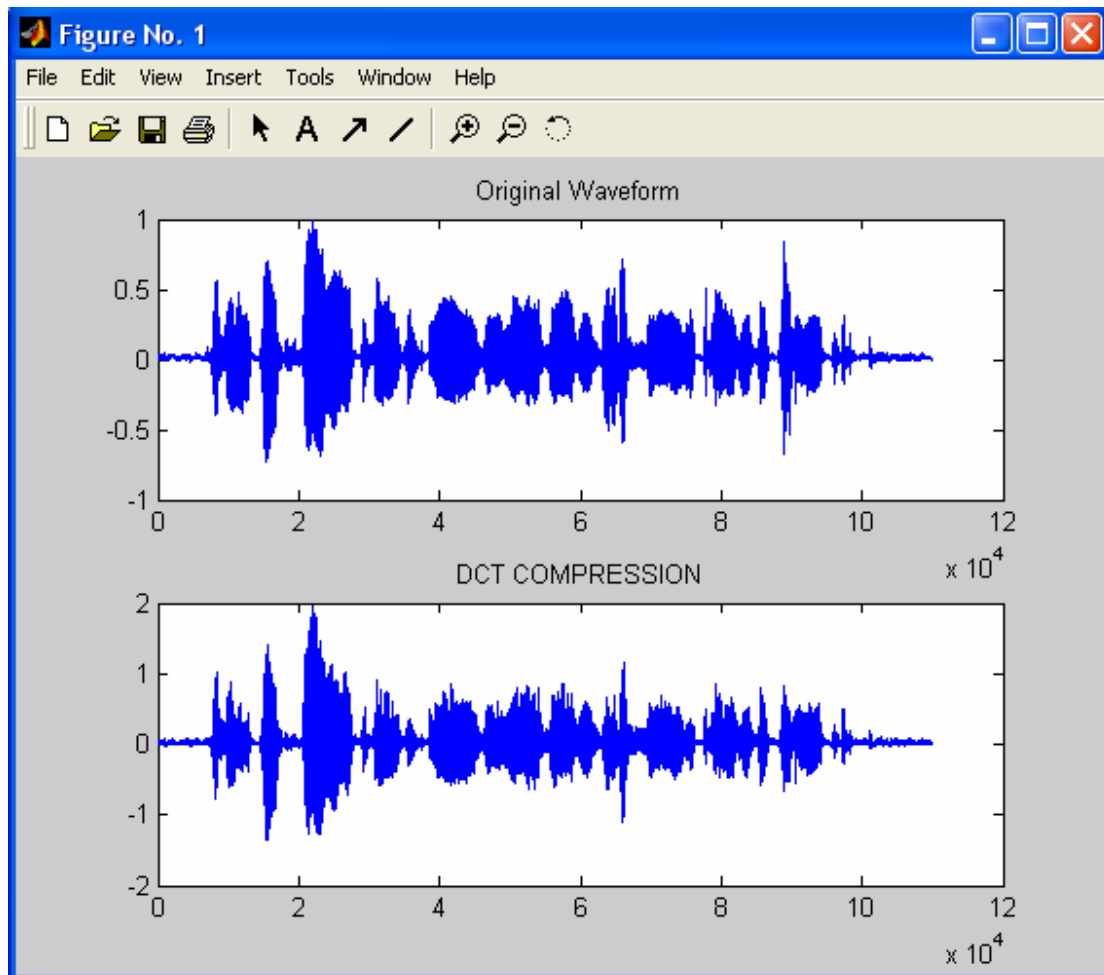
function christos
N=input ('dose timi gia to N:');
r=input ('dose timi gia to r:');
Cn=[];
at=[];
audio=[];
[fin, f] = wavread('christos.wav');
disp('Original')
wavplay(fin,f)
b=size(fin,1);
c=fix(b/N);
for j=1:c
at=fin((N*(j-1)+1):j*N);
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
Cn(n+1)=0;
for k=1:1:r
Cn(n+1)=Cn(n+1)+(at(k))*(cos((n*pi*(k+(1/2)))/N));
end
Cn(n+1)=((2*a)/N)*Cn(n+1);
end
for x=0:1:(N-1)
S=0;
for n=0:1:(N-1)
if n==0
a=1/(sqrt(2));
elseif n>0
a=1;
end
S=S+a*Cn(n+1)*(cos((n*pi*(x+1/2))/N));
end
fout(x+1)=S;
end
audio(1,(N*(j-1)+1):j*N)=fout;
end

disp('Compression Factor 2')
wavplay(audio, f)

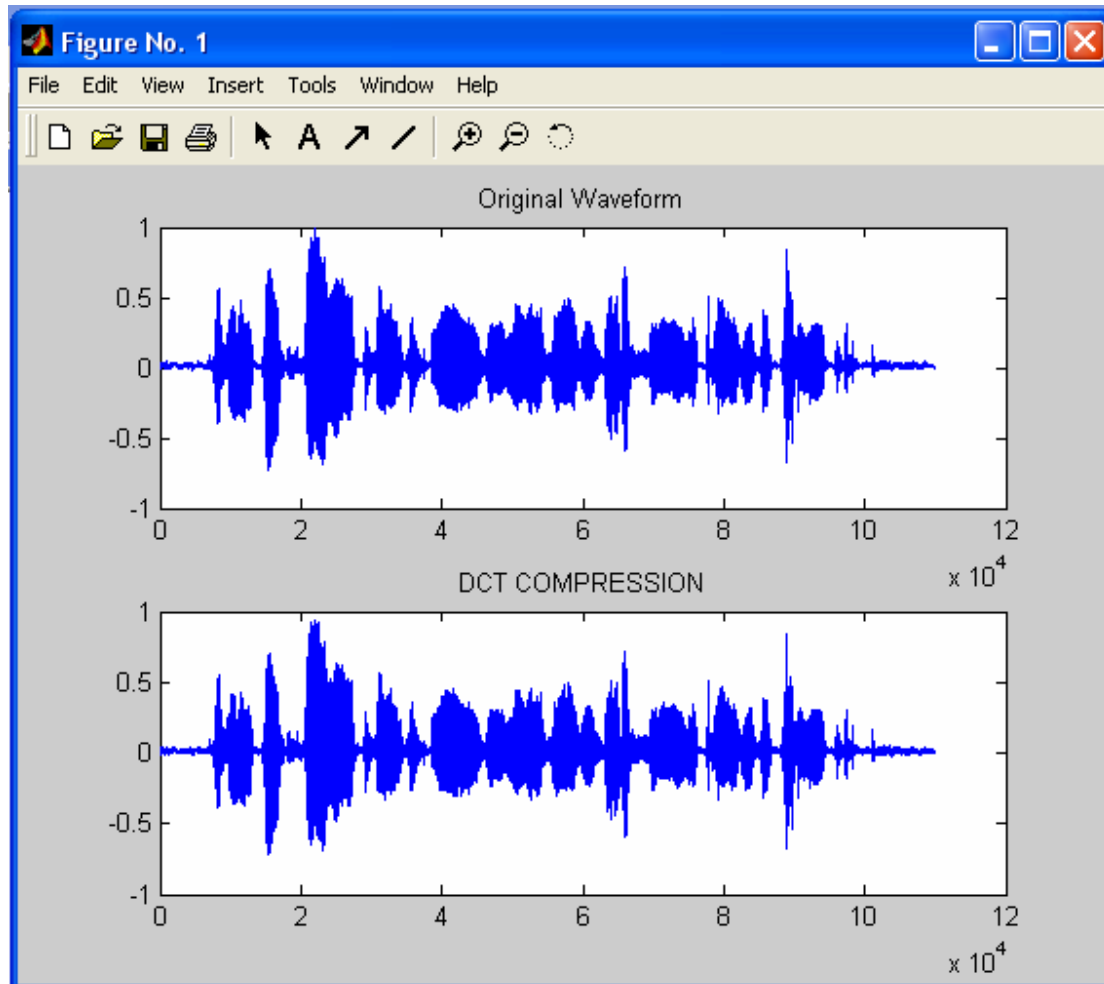
subplot(2,1,1);plot(fin);title('Original Waveform')
subplot(2,1,2);plot(audio);title('DCT COMPRESSION')

```

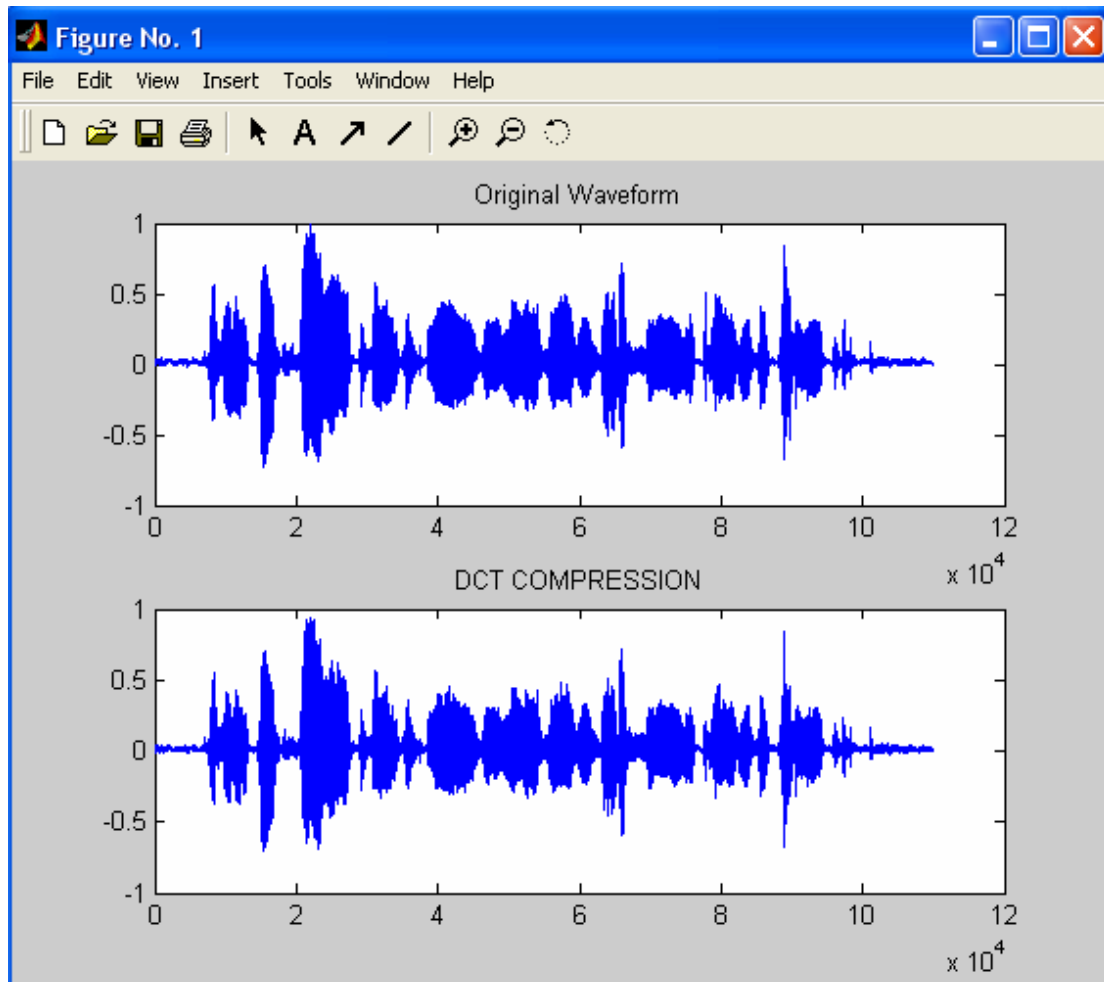
```
>> christos
dose timi gia to N:16
dose timi gia to r:16
Original
Compression Factor 2
```



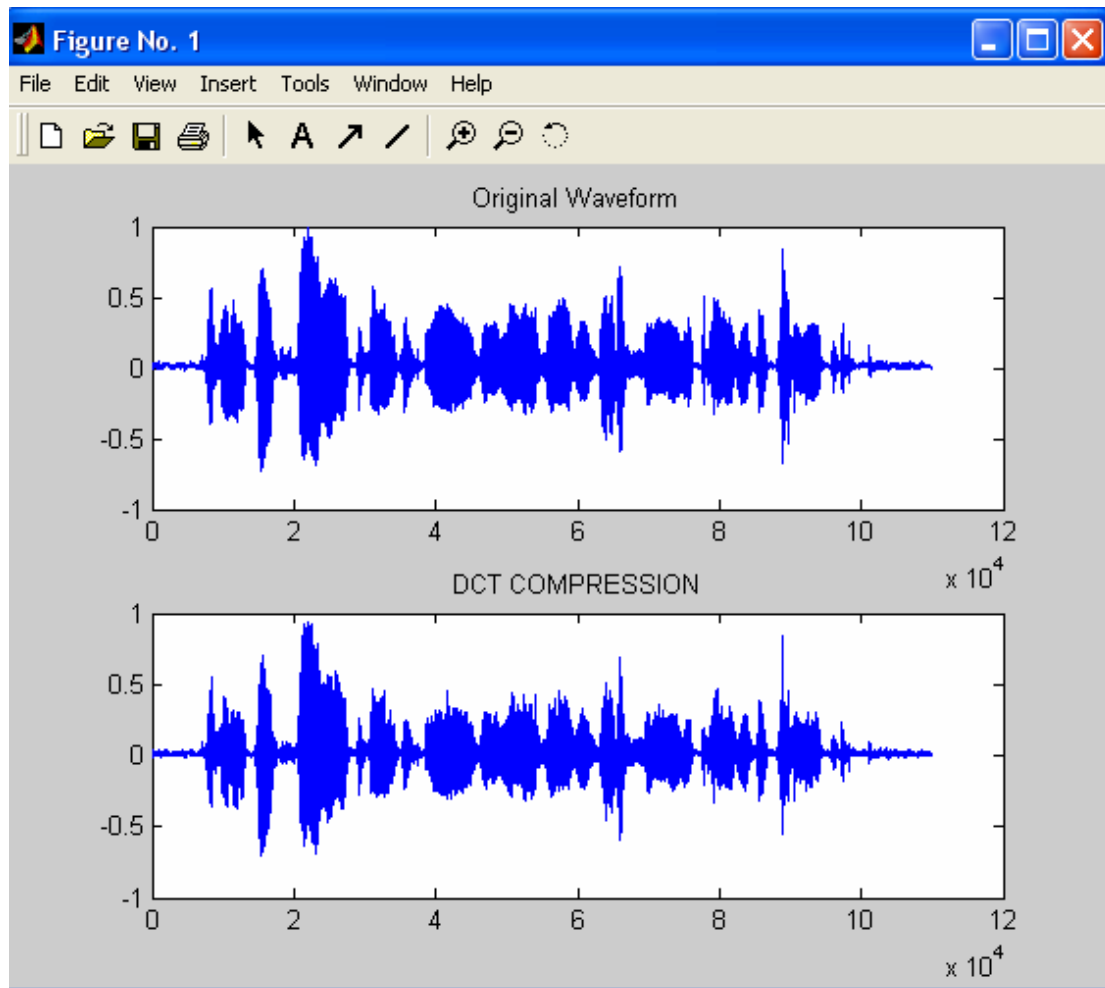
```
>> christos
dost = timi_gia_to_N:16
dost = timi_gia_to_r:12
Original
Compression Factor 2
```



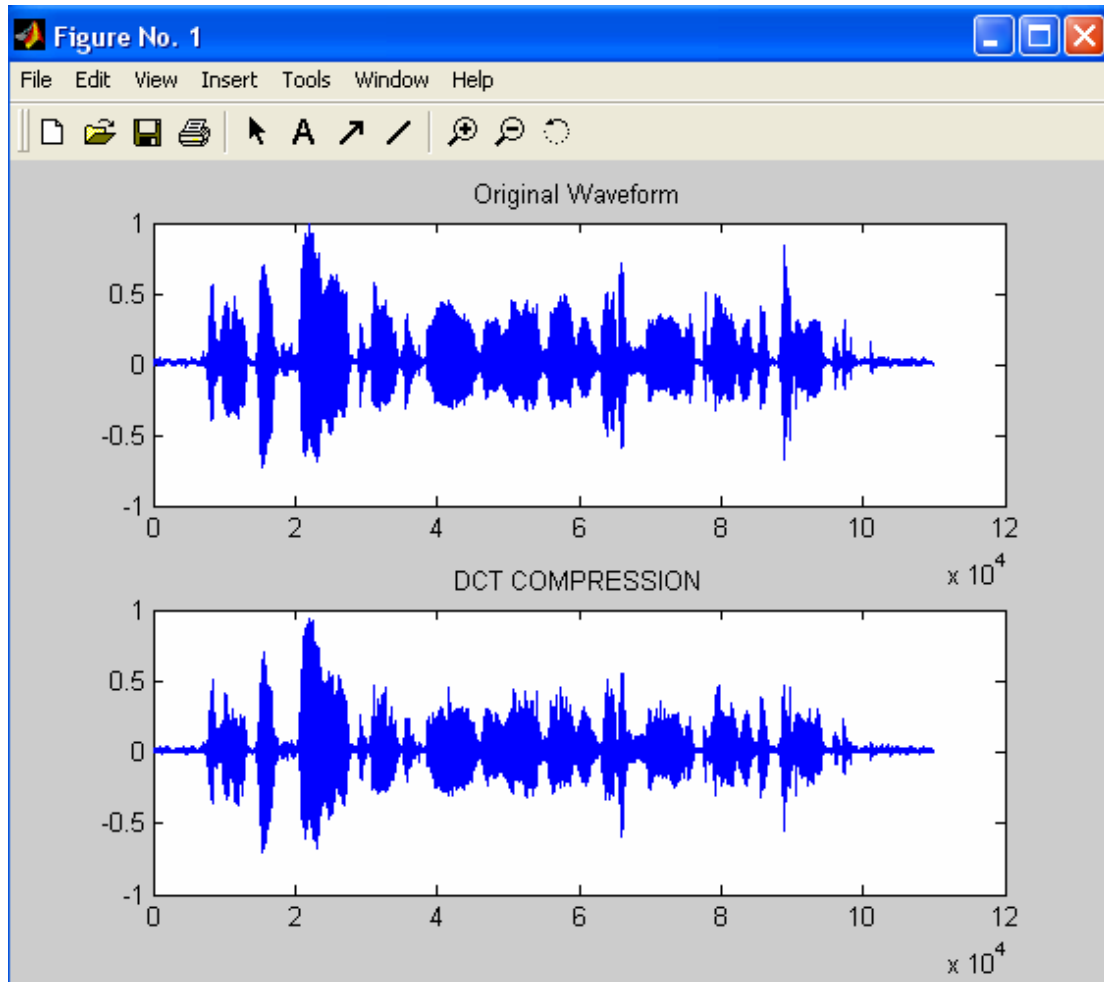

```
>> christos
dose timi gia to N:16
dose timi gia to r:8
Original
Compression Factor 2
```



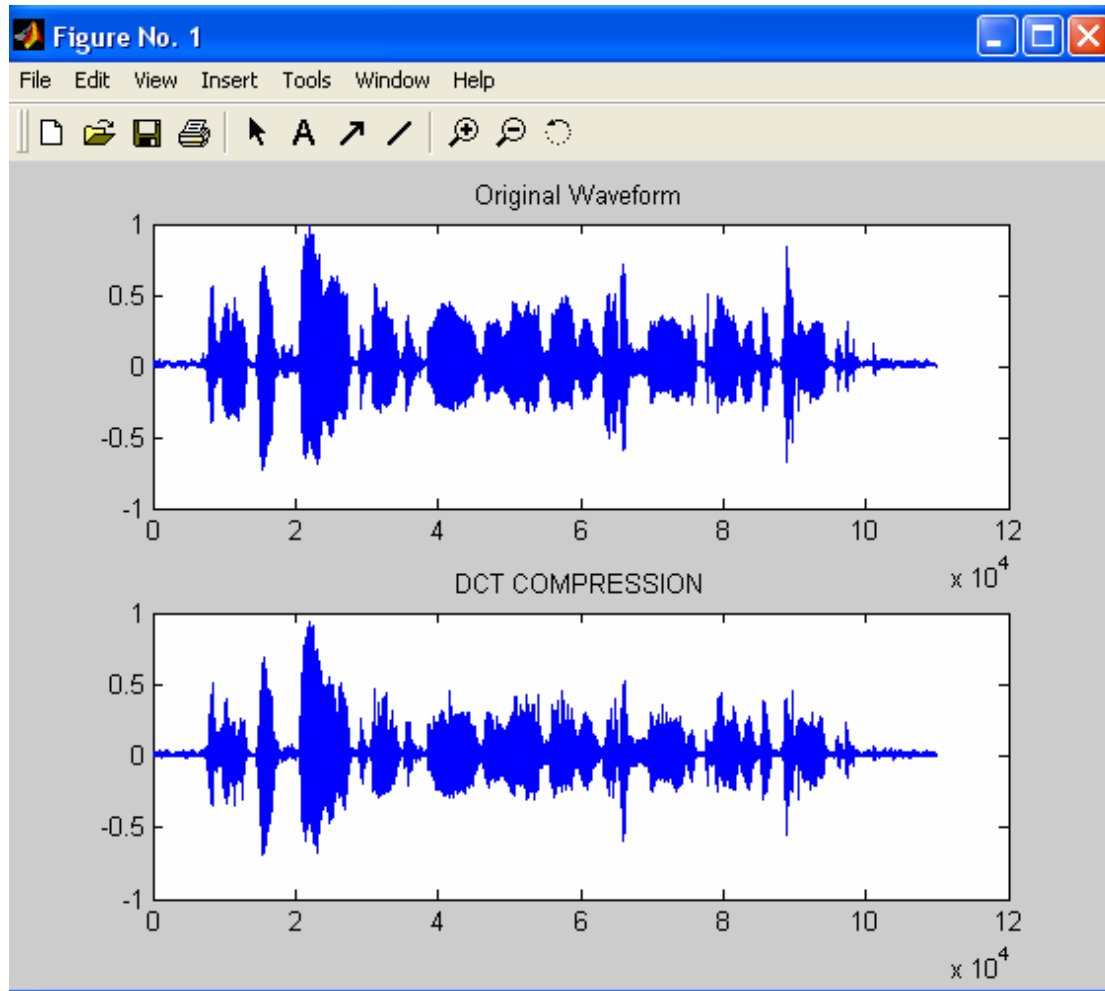
```
>> christos
dose timi gia to N:16
dose timi gia to r:4
Original
Compression Factor 2
```



```
>> christos
dose timi gia to N:16
dose timi gia to r:2
Original
Compression Factor 2
```



```
>> christos
dose timi gia to N:16
dose timi gia to r:1
Original
Compression Factor 2
```



Επειδή ασχολούμαστε με τη μέθοδο του DCT, θα δούμε πως, όσο μεγαλύτερη συμπίεση ζητάμε από το πρόγραμμα, τόσο χειρότερα αποτελέσματα έχουμε στη γραφική παράσταση που αφορά στη συμπίεση και κυρίως στη στιγμή που ακούγεται το κάθε συμπιεσμένο ηχητικό κείμενο χωριστά, αμέσως μετά από το κανονικό ηχητικό κείμενο.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Έχοντας, λοιπόν, ολοκληρώσει την αναλυτική παρουσίαση των δυο μεθόδων συμπίεσης τόσο σε περιπτώσεις εξισώσεων όσο και σε περιπτώσεις παραδειγμάτων ήχου, μπορούμε πλέον να διαμορφώσουμε πληρέστερη εικόνα σχετικά με τη χρήση των μετασχηματισμών DOGHT και DCT. Η εικόνα αυτή, όπως έγινε σαφές από το περιεχόμενο της εργασίας θα προκύψει με βάση τα αρχεία ήχου wav. Είναι φυσικό ότι στις περιπτώσεις των εξισώσεων είναι πιθανό για τη μορφή που προκύπτει να μην αφορά στον ήχο, αλλά αυτό δεν αποτέλεσε θέμα αυτής της εργασίας. Θα μπορούσε, όμως, να αποτελέσει το θέμα κάποιας μεταγενέστερης προσπάθειας.

Είναι, λοιπόν, σημαντική η αναφορά της χρήσης του προτύπου wav, κυρίως για την απλή αποθήκευση των pcm ηχητικών κυματομορφών, έχοντας ανάγκη μεγάλης χωρητικότητας ψηφιακής μνήμης. Το μέγεθος ενός ασυμπίεστου αρχείου μουσικής (wav) είναι 1.4Mbits/sec. Επομένως, για ένα τραγούδι διάρκειας 4 λεπτών χρειάζονται 42 Mbytes. Κατά συνέπεια, η μετάδοση του παραπάνω ασυμπίεστου αρχείου μουσικής .wav μέσω Internet απαιτεί 1h40min. Είναι φανερό ότι ο χρόνος μετάδοσης του ήχου μέσω του Internet είναι πολύ μεγάλος.

Το 1997 εμφανίστηκαν στο διαδίκτυο κωδικοποιημένα MP3, τα οποία χρησιμοποιούν το μετασχηματισμό συνημιτόνου DCT. Στην περίπτωση αυτή που χρησιμοποιούμε τον DCT μετασχηματισμό μπορούμε να συμπίεσουμε ένα αρχείο έως 10 φορές με ανεπαίσθητη μείωση της ποιότητάς του. Έτσι, το τραγούδι, το οποίο στο προηγούμενο παράδειγμα είχε διάρκεια τέσσερα λεπτά και χρειαζόταν 42Mbytes, τώρα συμπιέζεται 11 φορές και αποθηκεύει μνήμη 4 Mbytes. Επίσης, η αποστολή του από το Internet είναι περίπου 9 λεπτά.

Ο αλγόριθμος DOGHT μπορεί να χρησιμοποιηθεί στη θέση του DCT και η μορφή, στην οποία δίνεται, είναι εξίσου απλή και αναλυτική με αυτή του DCT. Ακόμη, κατά τη χρήση του αλγορίθμου DOGHT, ισχύουν ορισμένες μαθηματικές ιδιότητες, όπως η σχέση Parseval, οι οποίες παρουσιάζουν πολλές ομοιότητες με αυτές του μετασχηματισμού DCT. Οι μαθηματικές

ιδιότητες αυτές αποτελούν σημαντικότερο παράγοντα, χάρη στον οποίο καθίσταται πολύ ευκολότερη η εφαρμογή του μετασχηματισμού DOGHT.

Η σχέση Parseval στον DOGHT προσεγγίζει την ενέργεια του σήματος μέσα από ένα αριθμητικό ολοκλήρωμα του τύπου Gauss-Hermite, ενώ στον DCT η ενέργεια του σήματος προσεγγίζεται μέσα από ένα αριθμητικό ολοκλήρωμα του τύπου παραλληλογράμμου:

$$\tilde{E} = \frac{N \Delta t}{2} \sum_{n=0}^{N-1} |c_n|^2 = \Delta t \sum_{k=0}^{N-1} |f(t_k)|^2 \quad (\text{Σχέση 3.13})$$

Από τις εξισώσεις που αναγράφονται παραπάνω παρατηρούμε πως το αριθμητικό ολοκλήρωμα του τύπου Gauss-Hermite συγκλίνει με πολύ μεγαλύτερη ταχύτητα από τον απλό κανόνα του παραλληλογράμμου στις περισσότερες περιπτώσεις. Έτσι, λοιπόν, καταλαβαίνουμε ότι ο μετασχηματισμός DCT δεν προσεγγίζει τόσο καλά την ενέργεια του σήματος όσο ο DOGHT.

Επιπλέον, από το σύνολο των παραδειγμάτων συμπίεσης που προαναφέρθηκαν μέσα στην εργασία, προκύπτει πως ο συντελεστής κλιμάκωσης του χρόνου που υπολογίζεται από τον παρακάτω τύπο είναι $T=0.586$.

$$T = \frac{t_{N-1}}{t_{\max}}$$

Έγινε, ακόμη, σαφές πως ο αντίστροφος μετασχηματισμός DOGHT είναι ακριβής. Επίσης οι περιτοί συντελεστές του DOGHT είναι μηδενικοί εξαιτίας της άρτιας συμμετρίας που έχει η συνάρτηση. Από τον ίδιο πίνακα βλέπουμε και τα αντίστοιχα αποτελέσματα του μετασχηματισμού DCT. Το σπουδαιότερο στοιχείο που πρέπει να τονίσουμε είναι πως με το μετασχηματισμό DOGHT έχουμε την ικανότητα να συγκεντρώσουμε περισσότερη ενέργεια σήματος στους συντελεστές χαμηλότερης τάξης (χαμηλότερες «συχνότητες»). Σαν αποτέλεσμα με τον DOGHT επιτυγχάνουμε

καλύτερη συμπίεση. Χαρακτηριστικό παράδειγμα είναι οι συντελεστές μηδενικής τάξης όπου έχουμε $c_0=0.39$ για τον DCT και $c_0=1.06$ για τον DOGHT.

Επιπροσθέτως, τα αποτελέσματα των κυματομορφών μας δείχνουν ότι η μέθοδος του DOGHT είναι πολύ καλή για σήματα που έχουν κυματομορφή σχήματος καμπάνας, γιατί παρατηρούμε ότι η μέθοδος του DOGHT χάνει μόνο στα άκρα της κυματομορφής. Η κυματομορφή αυτή μπορεί να περιλαμβάνει τόσο ντετερμινιστικά όσο και τυχαία σήματα. Στις υπόλοιπες περιπτώσεις αποδείχθηκε μέσα από τα παραδείγματα ότι ο μετασχηματισμός DCT είναι μάλλον προτιμότερος από τους χρήστες.

Επίσης, όσον αφορά στο ποσοστιαίο RMS σφάλμα, ο μετασχηματισμός DCT για συμπίεσεις μικρής μορφής είναι καλύτερος, επειδή το σφάλμα σε αυτή την περίπτωση είναι σαφώς μικρότερο από τον DOGHT. Διαπιστώνουμε, όμως, εύκολα πως, όταν πρόκειται για μεγαλύτερες συμπίεσεις, ο DOGHT είναι πολύ καλύτερος από τον DCT. Αυτό ισχύει, διότι το ποσοστό σφάλματος για τον DCT ανεβαίνει κατακόρυφα, κάτι το οποίο δε συμβαίνει στον DOGHT.

Τέλος, από τα παραδείγματα που περιέχουν ήχο συμπεραίνουμε τόσο από το άκουσμα όσο και από τις γραφικές παραστάσεις πως, όσο μεγαλύτερη συμπίεση ζητάμε, τόσο περισσότερο μειώνεται η ποιότητα του ήχου. Σε κάθε μια από τις γραφικές παραστάσεις φαίνεται πως στα άκρα το σήμα χάνει ποιότητα και αυτό, γιατί τα πάνω και κάτω όρια της κάθε γραφικής παράστασης από το ένα ποσοστό συμπίεσης στο άλλο αυξάνονται μεταξύ τους.

Με βάση, λοιπόν, όλα τα παραπάνω γίνεται πλέον σαφές ότι η χρησιμοποίηση μεγαλύτερων λόγων συμπίεσης σε εφαρμογές ήχου και αλγορίθμων σχετικά απλών και όχι πολύπλοκων κατά τη διάρκεια υπολογισμών παραμένει ακόμη και σήμερα εξαιρετικά αναγκαία.

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία έχει βασικό της αντικείμενο τη μελέτη των περιπτώσεων συμπίεσης ηχητικών σημάτων είτε μέσω παραδειγμάτων εξισώσεων είτε μέσω παραδειγμάτων ήχου. Αρχικά, λοιπόν, ορίζεται ο ήχος μαζί με τα μεγέθη που πρέπει να προσέχουμε κατά τον έλεγχο της ποιότητας του ήχου καθώς και τα προβλήματα που δημιουργεί.

Κατόπιν, αναλύεται λεπτομερώς το θεώρημα δειγματοληψίας που αποτελεί βασικό παράγοντα για τη συμπίεση του ήχου και ο κβαντισμός σημάτων σε συνδυασμό με το σφάλμα κβαντισμού. Ακολούθως, περιγράφονται οι μετατροπές από αναλογικό σε ψηφιακό σήμα και αντιστρόφως και, φυσικά, η ψηφιοποίηση. Τέλος, παρουσιάζονται οι δυο βασικότερες κατηγορίες προτύπων ήχου, το πρότυπο wave (wav) και το πρότυπο motion picture experts group (mpeg). Στο τελευταίο κομμάτι του θεωρητικού τμήματος της εργασίας αναλύεται η λειτουργία των μετασχηματισμών DOST και DCT και πραγματοποιείται η παρουσίαση του MATLAB ως γλώσσας προγραμματισμού.

Αφού τελειώσει το θεωρητικό αυτό μέρος, ακολουθεί το πειραματικό που βασίζεται αποκλειστικά και μόνο σε αλγορίθμους. Στόχος των αλγορίθμων είναι να κατανοήσουμε ποιος από τους δυο μετασχηματισμούς είναι καλύτερος ανά περίπτωση, με ξεχωριστό κάθε φορά υπολογισμό συμπίεσης και αποσυμπίεσης. Εκτός από αυτά, φαίνονται και τα αποτελέσματα από ορισμένες παραμέτρους, όπως ο συντελεστής μετασχηματισμού C_n , καθώς και οι τιμές σφάλματος που μπορεί να προκύψουν από τις παραμέτρους αυτές. Από τους πίνακες, λοιπόν, των παραδειγμάτων αυτών μπορούμε να καταλήξουμε στο συμπέρασμα πως ο μετασχηματισμός DOST είναι ιδανικότερος στη συμπίεση σημάτων που έχουν κυματομορφή καμπάνας στην οποία η πληροφορία συγκεντρώνεται στη μέση του σήματος, ακόμη και όταν η κυματομορφή περιλαμβάνει ντετερμινιστικά ή τυχαία σήματα, ενώ στις υπόλοιπες περιπτώσεις θα πρέπει μάλλον να προτιμηθεί από τους χρήστες ο μετασχηματισμός DCT.

ΒΙΒΛΙΟΓΡΑΦΙΑ

I. ΒΟΗΘΗΜΑΤΑ

1. Μιχάλης Σπάσος, «Ηλεκτρονικές Μετρήσεις και Οργανολογία-Θεωρία και Πειράματα», εκδ. Αϊβάζης, Θεσσαλονίκη 2002.
2. Herbert Taub – Donald L. Schilling, «Αρχές Τηλεπικοινωνιακών Συστημάτων», εκδ. Τζιόλα (3^η Βελτιωμένη Ελληνική Έκδοση) Θεσσαλονίκη 2005.
3. Simon Haykin, «Συστήματα Επικοινωνίας», εκδ. Παπασωτηρίου, Αθήνα 1995.
4. Γεώργιος Γεωργανόπουλος, «Εισαγωγή στη Θεωρητική Επεξεργασία των Ψηφιακών και Αναλογικών Σημάτων και Συστημάτων», εκδ. Αφοί Κυριακίδη, Θεσσαλονίκη 2003.
5. Γεώργιος Παπανικολάου, «Ηλεκτρακουστική», εκδ. University Studio Press, Θεσσαλονίκη 1985.
6. Χαράλαμπος Δημούλας, «Τεχνικές Επεξεργασίας και Διαχείρισης Οπτικοακουστικού Περιεχομένου στη Μελέτη Βιοακουστικών Φαινομένων», Διδακτορική Διατριβή, Θεσσαλονίκη 2006.
7. Παύλος Λαζαρίδης, “Signal compression method using the discrete orthogonal Gauss-Hermite transform “, European Patent, EP1 406 212, Θεσσαλονίκη 2004.
8. Παύλος Λαζαρίδης, “Signal compression method using the discrete orthogonal Gauss-Hermite transform“, Greek Patent, GR20020100423, Θεσσαλονίκη 2002.
9. Παύλος Λαζαρίδης, Guy Debarge & Philippe Gallion, “Signal Compression using the Discrete Orthogonal Gauss-Hermite Transform”, submitted IEEE, 2005.
10. Παύλος Λαζαρίδης, Guy Debarge & Philippe Gallion, “Discrete orthogonal Gauss-Hermite transform for optical pulse propagation analysis”, J. Opt. Soc. Am. B, Vol. 20 (2003), No. 7, p. 1508-1513.

11. Δημήτριος Αργυρόπουλος & Βασίλειος Φλωρίδης, «Συμπύεση ήχου με χρήση προηγμένων μεθόδων επεξεργασίας σήματος», Διδακτορική διατριβή ΑΠΘ, Θεσσαλονίκη 2003.
12. Στάυρος Πανάς, «Ψηφιακή Επεξεργασία Σημάτων», εκδ. University Studio Press, Θεσσαλονίκη, 1987.
13. Στάυρος Πανάς, «Επεξεργασία Αναλογικών Σημάτων», εκδ. Υπηρεσία Δημοσιευμάτων, Θεσσαλονίκη, 1996.
14. Ιωάννης Σαρρής «Ισοστάθμιση της ακουστικής απόκρισης κλειστών χώρων με χρήση τεχνικών ψηφιακής επεξεργασίας σήματος», Διδακτορική διατριβή ΕΜΠ, Αθήνα 2005

II. ΙΣΤΟΣΕΛΙΔΕΣ

1. <http://www.lib.ece.ntua.gr/> , 12/10/2007
2. <http://shmmmy.ntua.gr/> , 1/11/2007
3. <http://www.sulaco.org/mp3/gpsycho/quality.html>.