

**ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
Ι Δ Ρ Υ Μ Α**



ΠΕΛΟΠΟΝΝΗΣΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

<<ΜΕΛΕΤΗ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΝΕΛΙΚΤΙΚΩΝ ΚΩΔΙΚΩΝ>>

ΓΑΛΑΝΗ ΝΙΚΟΛΕΤΤΑ

A.M 2010039

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Κ.ΜΠΟΖΑΤΖΗΣ ΒΑΣΙΛΕΙΟΣ

ΣΠΑΡΤΗ ΦΕΒΡΟΥΑΡΙΟΣ 2017

ΕΥΧΑΡΙΣΤΙΕΣ

Αισθάνομαι την ανάγκη να επισημάνω την απέραντη ευγνωμοσύνη μου στους γονείς μου ,για όλα όσα μου έχουν προσφέρει στην διάρκεια των μαθητικών και φοιτητικών μου χρόνων και την αμέριστη υποστήριξη τους σε κάθε μου επιλογή.

Τέλος,θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ.Μποζατζή Βασίλειο για την εμπιστοσύνη και την αφιέρωση πολύτιμου χρόνου ,ώστε να ολοκληρωθεί η εργασία αυτή.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία μελετά τους συνελκτικούς κώδικες και τη σημασία τους. Και δίνει πρακτικά παραδείγματα κωδικοποίησης μέσω της εφαρμογής matlab.

Αρχικά στην εργασία παρουσιάζονται τα συστήματα υπολογιστών και η έννοια της συμπίεσης και της κωδικοποίησης. Γίνεται ιστορική αναδρομή στη συμπίεση και τις αρχές που τη διέπουν και παρουσιάζονται οι βασικοί τύποι αλγορίθμων που βοηθούν στη διαδικασία της συμπίεσης και παρουσιάζονται εφαρμογές αυτής. Στη συνέχεια παρουσιάζεται η κωδικοποίηση πηγής και εντροπίας.

Ακολουθεί η παρουσίαση της κωδικοποίησης Hoffman στην οποία γίνεται παρουσίαση του αλγόριθμου και παρουσιάζεται και εφαρμογή της συγκεκριμένης μορφής κωδικοποίησης.

Στη συνέχεια παρουσιάζεται η θεωρία των συνελκτικών κωδίκων, οι διάφοροι τύποι κωδικοποίησης και τα βασικά στοιχεία της θεωρίας.

Τέλος γίνεται παρουσίαση του προγράμματος που χρησιμοποιήσαμε για την εφαρμογή των κωδίκων και παρατίθενται παραδείγματα κωδικοποίησης.

Περιεχόμενα

Περίληψη.....	3
Σύστημα υπολογιστών.....	8
1.1. Ιστορική αναδρομή της συμπίεσης.....	8
1.2. Γενικές αρχές συμπίεσης.....	9
1.3. Διάκριση δύο τύπων αλγορίθμων συμπίεσης.....	10
1.3.1. Αλγόριθμοι συμπίεσης χωρίς απώλειες ή αντιστρεπτοί (<i>lossless compression</i>).....	10
1.3.2. Αλγόριθμοι συμπίεσης με απώλειες ή μη αντιστρεπτοί (<i>lossy compression</i>).....	10
1.4. Εφαρμογές.....	10
1.5. Κωδικοποίηση εντροπίας και κωδικοποίηση πηγής.....	11
1.5.1. Κωδικοποίηση εντροπίας.....	12
1.5.1.1. Περιορισμός των ακολουθιών επαναλαμβανόμενων χαρακτήρων.....	12
1.5.1.2. Στατιστική Κωδικοποίηση (<i>Statistical encoding</i>).....	12
1.5.1.3. Αντικατάσταση προτύπων.....	13
1.5.1.4. Κωδικοποίηση <i>Huffman</i>	13
1.5.1.5. Κωδικοποίηση πηγής.....	14
1.5.1.6. Κωδικοποίηση μετασχηματισμού (<i>transform encoding</i>).....	14
Κεφάλαιο 2 Κωδικοποίηση <i>Huffman</i>	16
2.1. Παρουσίαση του αλγορίθμου.....	16
2.2. Συμπίεση Δεδομένων: Μοντελοποίηση και Κωδικοποίηση.....	17
2.3. Εφαρμογή αλγορίθμου.....	18
Κεφάλαιο 3 ΚΩΔΙΚΟΠΟΙΗΣΗ ΚΑΝΑΛΙΟΥ.....	20
3.1. Τυχαιοποίηση - Διασπορά Ενέργειας.....	23
3.2. Εξωτερική Κωδικοποίηση.....	26
3.3. Εσωτερική Κωδικοποίηση.....	27
3.4. Συνελικτικός Κωδικοποιητής.....	29
ΚΕΦΑΛΑΙΟ 4ο ΘΕΩΡΙΑ ΚΩΔΙΚΩΝ ΣΥΝΕΛΙΚΤΙΚΟΙ ΚΩΔΙΚΕΣ.....	31
4.1. Κωδικοποίηση ελέγχου-διόρθωσης λαθών.....	33
4.2. Συνελικτικοί Κώδικες.....	36
4.2.1. Συναλυσώμενη Κωδικοποίηση.....	39
4.2.2. <i>Concatenation</i> Μέθοδοι.....	40
4.2.3. <i>Block Interleaving</i>	40
4.3. Βασικά Στοιχεία Θεωρίας.....	41

4.4. Συνάρτηση Μεταφοράς.....	42
4.4.1. Διάγραμμα Καταστάσεων (State Diagram).....	43
4.4.2. Δεντρικό Διάγραμμα (Tree Diagram).....	44
4.4.3. Διάγραμμα trellis(Trellis Diagram).....	47
4.5. Παράλληλα Συνδεδεμένα Συνελικτικοί Κώδικες (Parallel Concatenated Convolutional Codes-PCCC)-Turbo Κώδικες.....	48
ΕΜΠΕΙΡΙΚΗ ΔΙΕΡΕΥΝΗΣΗ.....	58
ΠΑΡΑΔΕΙΓΜΑΤΑ.....	65
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	69

ΕΙΣΑΓΩΓΗ

Οι συνελκτικοί κώδικες αποτελούν μια πολύ σημαντική κατηγορία κωδίκων, οι οποίοι επιτυγχάνουν αρκετά μεγάλα κέρδη στην κωδικοποίηση και η χρήση τους είναι ευρεία στα τηλεπικοινωνιακά συστήματα. Η ονομασία τους οφείλεται στον τρόπο με τον τρόπο που υλοποιείται η κωδικοποίηση της πληροφορίας, καθώς σχηματίζονται με συνέλιξη των ψηφίων της πληροφορίας με την κρουστική απόκρουση ενός καταχωρητή ολίσθησης.

Με την χρήση των συνελκτικών κωδίκων, ένα μπλοκ των k δυαδικών ψηφίων αντιστοιχίζεται σε ένα block των n δυαδικών ψηφίων. Κύριο χαρακτηριστικό γνώρισμα των συνελκτικών κωδίκων είναι ότι το νέο μπλοκ των n δυαδικών ψηφίων δεν εξαρτάται μόνο από τα τρέχοντα k δυαδικά ψηφία, αλλά και από προηγούμενα δυαδικά ψηφία. Αυτό σημαίνει ότι ο κωδικοποιητής πρέπει να έχει μνήμη και ως εκ τούτου μετατρέπεται σε μια Μηχανή Πεπερασμένων Καταστάσεων.

Ένας συνελκτικός κωδικοποιητής υλοποιείται με έναν απλό καταχωρητή ολίσθησης και modulo-2 αθροιστές. Ένας συνελκτικός κώδικας χαρακτηρίζεται από την τριάδα (n,k,L) .

Ο ακέραιος αριθμός k αποτελεί τον αριθμό των ψηφίων που εισέρχονται στον κωδικοποιητή, n είναι ο αριθμός των ψηφίων της εξόδου του κωδικοποιητή σε κάθε χρονικό βήμα και L ονομάζεται το μήκος περιορισμού και δηλώνει τον αριθμό των ομάδων των k ψηφίων που βρίσκονται κάθε στιγμή στον καταχωρητή.

Ρυθμός ενός συνελκτικού κώδικα ορίζεται ως ο λόγος του αριθμού των ψηφίων στην είσοδο του κωδικοποιητή προς τον αριθμό των ψηφίων στην έξοδο του κωδικοποιητή. Με άλλα λόγια:

$$R=k/n$$

Επίσης ως ελεύθερη απόσταση ενός συνελκτικού κώδικα ορίζεται η ελάχιστη απόσταση μεταξύ δύο κωδίκων ακολουθιών.

Επειδή, ένας συνελκτικός κωδικοποιητής είναι μια διάταξη με μνήμη, είναι φανερό ότι οι n -άδες εξόδου του κωδικοποιητή δεν εξαρτώνται μόνο από τα k πιο πρόσφατα δυαδικά ψηφία στην είσοδο του κωδικοποιητή, αλλά επίσης και από τα $(L-1)k$ δυαδικά ψηφία που περιείχαν οι πρώτες $(L-1)k$ βαθμίδες του καταχωρητή πριν φθάσουν τα νέα k bits. Επομένως, προκύπτει ότι ο καταχωρητής ολίσθησης είναι μια μηχανή πεπερασμένων καταστάσεων με $2(L-1)k$ καταστάσεις.

Σε κάθε χρονική στιγμή, εισέρχονται στον καταχωρητή ολίσθησης k δυαδικά ψηφία πληροφορίας και εξέρχονται αυτού τα k δυαδικά ψηφία του τελευταίου σταδίου του καταχωρητή ολίσθησης. Από την στιγμή που τα k δυαδικά ψηφία εισέρχονται στον καταχωρητή ολίσθησης, υπολογίζονται n γραμμικοί συνδυασμοί των περιεχομένων του καταχωρητή ολίσθησης και αυτοί χρησιμοποιούνται για την παραγωγή της κωδικοποιημένης εξόδου.

Σύστημα υπολογιστών

Ο υπολογιστής δεν είναι κατασκευασμένος να εκτελεί κάποιες προκαθορισμένες εργασίες, αλλά είναι ένα σύγχρονο εργαλείο που έχει σχεδιαστεί με τέτοιο τρόπο ώστε να μας βοηθά στη διεκπεραίωση πολύπλοκων, χρονοβόρων, και συχνά βαρετών εργασιών. Τα κύρια χαρακτηριστικά των ηλεκτρονικών υπολογιστών είναι η μεγάλη ταχύτητα επεξεργασίας και αποθήκευσης δεδομένων και η αποδοτικότητά τους. Επίσης, μπορούν να εκτελέσουν οποιαδήποτε εργασία, από την απλή δημιουργία ενός εγγράφου μέχρι τη λύση πολύπλοκων μαθηματικών προβλημάτων και την αρχειοθέτηση χιλιάδων καταχωρίσεων, αρκεί να τους τροφοδοτήσουμε με τις κατάλληλες εντολές.

1.1. Ιστορική αναδρομή της συμπίεσης

Η συμπίεση αρχείων είναι ίσως η θεμελιώδης έκφραση της θεωρίας της πληροφορίας. Η θεωρία της πληροφορίας είναι ένας κλάδος των μαθηματικών που έχει τις ρίζες του στα τέλη του 1940. Χάρη στη δουλειά του Claude Shannon στο εργαστήριο Bell Labs, ο οποίος αναφερόταν σε διάφορα ερωτήματα για την πληροφορία συμπεριλαμβάνοντας διαφορετικούς τρόπους αποθήκευσης και επικοινωνίας μηνυμάτων.

Η συμπίεση δεδομένων μπαίνει στα χωράφια της θεωρίας της πληροφορίας εξαιτίας του συσχετισμού της με τον πλεονασμό. Η επιπλέον πληροφορία σε ένα μήνυμα χρησιμοποιεί επιπλέον bits για να κωδικοποιηθεί. Αν θα θέλαμε να ξεφορτωθούμε αυτή την επιπλέον πληροφορία θα έπρεπε να μειώναμε το μέγεθος του μηνύματος. Η θεωρία της πληροφορίας χρησιμοποιεί τον όρο της εντροπίας ως ένα μέτρο για το πόση πληροφορία κωδικοποιείται σε ένα μήνυμα. Η λέξη εντροπία πηγάζει από τη θερμοδυναμική και έχει παρόμοια σημασία. Όσο υψηλότερη είναι η εντροπία σε ένα μήνυμα τόσο περισσότερη πληροφορία περιέχει. Η εντροπία ενός συμβόλου προσδιορίζεται ως ο αρνητικός λογάριθμος της πιθανότητας.

Number of bits = - Log base 2 (probability)

Η εντροπία για ένα ολόκληρο μήνυμα είναι το σύνολο της εντροπίας του κάθε συμβόλου ατομικά. Η εντροπία συνδέεται με τη συμπίεση δεδομένων έτσι ώστε να αποφασιστεί πόσα bits πληροφορίας εμφανίζονται στο μήνυμα. Αν η πιθανότητα

εμφάνιση του χαρακτήρα 'e' σε αυτό το χειρόγραφο είναι 1/16 για παράδειγμα η πληροφορία που αναλογεί σε χαρακτήρες είναι 4 bits.

'eeee': $5 \cdot 4 = 20$ bits

Αν χρησιμοποιηθούν 8-bit ASCII, $5 \cdot 8 = 40$ bits χρησιμοποιούνται για να κωδικοποιηθεί το μήνυμα. Το πρόβλημα είναι ότι όταν υπολογίζουμε την εντροπία χρησιμοποιούμε έναν αριθμό που μας δίνει την πιθανότητα του δοθέντος συμβόλου. Αν αλλάξουμε το μοντέλο θα αλλάξει και η πιθανότητα. Ένα σύμβολο που έχει υψηλή πιθανότητα έχει λίγες πληροφορίες και θα χρειαστεί λιγότερα Bits για να κωδικοποιηθεί.

1.2. Γενικές αρχές συμπίεσης

Ορισμός συμπίεσης: Ως συμπίεση (compression) δεδομένων ορίζουμε την αφαίρεση της "πλεονάζουσας πληροφορίας" από αρχεία, ώστε να προκύπτει όσο το δυνατό πιο "χρήσιμη πληροφορία", έτσι ώστε αν γίνει αποσυμπίεση (decompression) να μας δώσει την αρχική ασυμπίεστη πληροφορία. Στη διαδικασία αυτή, υπάρχει περίπτωση είτε να μη φαίνονται τα αποτελέσματα της συμπίεσης είτε ακόμη και να αλλάζει σημαντικά το αρχικό αρχείο. Στόχος συμπίεσης: Είναι ο περιορισμός του μεγέθους που καταλαμβάνει ένα ποσό πληροφορίας εις βάρος βέβαια της διαθεσιμότητάς του, της υπολογιστικής ισχύος και πολύ συχνά και της ακρίβειας του περιεχομένου του. Αυτό σημαίνει, ότι οι διαδικασίες συμπίεσης και αποσυμπίεσης έχουν υπολογιστικό κόστος, που μπορεί να είναι τόσο μεγάλο που να απαιτεί ειδικό υλικό για να γίνει σε πραγματικό χρόνο. Από την άλλη πλευρά, η συμπιεσμένη μορφή της πληροφορίας δεν είναι άμεσα αξιοποιήσιμη. Πρέπει να προηγηθεί το στάδιο της αποσυμπίεσης για να αποκτήσει ξανά το σημασιολογικό της περιεχόμενο. Συνήθως μας απασχολεί η ταχύτητα αποσυμπίεσης και όχι τόσο αυτή της συμπίεσης. Στις περισσότερες εφαρμογές η συμπίεση γίνεται μια φορά στο στάδιο της κατασκευής και με χρήση ειδικού υλικού, ενώ η αποσυμπίεση γίνεται από τους χρήστες που έχουν στην διάθεσή τους υπολογιστές γενικής χρήσης.

1.3. Διάκριση δύο τύπων αλγορίθμων συμπίεσης

1.3.1. Αλγόριθμοι συμπίεσης χωρίς απώλειες ή αντιστρεπτοί (lossless compression)

Αυτό το είδος αλγορίθμων έχει το ιδιαίτερο χαρακτηριστικό ότι η διαδικασία συμπίεσης δεν αλλοιώνει καθόλου την πληροφορία. Δηλαδή, μετά την αποσυμπίεση, η πληροφορία επανέρχεται ακριβώς στη μορφή που είχε πριν. Συνήθως, αυτοί οι αλγόριθμοι εφαρμόζονται σε περιπτώσεις που δεν υπάρχει κανένα περιθώριο απωλειών. Για παράδειγμα, αν η πληροφορία που μεταφέρεται είναι ένα πρόγραμμα υπολογιστή, ένα και μόνο αλλοιωμένο bit μπορεί να είναι αρκετό να καταστήσει το πρόγραμμα άχρηστο.

1.3.2. Αλγόριθμοι συμπίεσης με απώλειες ή μη αντιστρεπτοί (lossy compression)

Σε αυτόν τον τύπο συμπίεσης, η πληροφορία μπορεί να περιγράψει μια φωτογραφία. Είναι δυνατόν να επιτύχουμε καλύτερη συμπίεση κάνοντας μερικές υποχωρήσεις όσον αφορά στην πιστότητα του συμπιεσμένου σήματος. Είναι φανερό ότι σε τέτοιες περιπτώσεις το σημασιολογικό περιεχόμενο ουσιαστικά δεν μεταβάλλεται αλλά υπεισέρχεται η έννοια της μείωσης της ποιότητας. Το ψηφιακό σήμα ως ακολουθία bits σαφώς και μεταβάλλεται.

1.4. Εφαρμογές

Από τα γνωστά αρχεία που δεν επιδέχονται απώλεια των δεδομένων είναι τα αρχεία κειμένου και τα αρχεία δεδομένων Η/Υ. Αρχεία που επιδέχονται απώλειες των δεδομένων τους, με αντίτιμο τη μείωση της ποιότητάς τους, είναι τα αρχεία εικόνων, γραφικών, βίντεο, ήχου, animation κ.ά. Υπάρχουν απωλεστικές μέθοδοι συμπίεσης - αποσυμπίεσης, όπου η απώλεια της πληροφορίας δεν είναι εμφανής σε χαμηλές συμπίεσεις. Τις περισσότερες φορές, όταν στέλνουμε αρχεία εικόνας ή ήχου σε πραγματικό χρόνο (real-time), π.χ. τηλεδιάσκεψη, τηλεσυνεργασία, ζωντανή μετάδοση εκδηλώσεων μέσω του Internet, αλληλεπιδραστική επικοινωνία, που απαιτούν μετάδοση μεγάλου όγκου πληροφορίας σε πολύ μικρό χρόνο προσφεύγουμε σε τεχνικές μείωσης της χρωματικής πληροφορίας και της ανάλυσής τους, ώστε να φτάνουν πιο γρήγορα στον προορισμό τους. Γιατί περισσότερο μας ενδιαφέρει ο χρόνος μετάδοσης (να φτάσει η πληροφορία στο δέκτη μέσα σε

συγκεκριμένο χρονικό διάστημα) παρά η ποιότητα της εικόνας και του ήχου. Συνεπώς η συμπίεση είναι εξαιρετικά κρίσιμη και αναγκαία. Το χαρακτηριστικό αυτό είναι πολύ σημαντικό για την ιατρική εικόνα ειδικά, όπου απώλεια στην εικόνα μπορεί να σημαίνει απώλεια σημαντικής ιατρικής πληροφορίας. Η αξιοπιστία μιας εφαρμογής τηλε-διάγνωσης στηρίζεται σε μεγάλο βαθμό στη διασφάλιση της ιατρικής διάγνωσης. Οι μη αντιστρέψιμοι αλγόριθμοι πετυχαίνουν μεγάλους λόγους συμπίεσης με αντάλλαγμα απώλειες στην εικόνα. Οι απώλειες αυτές συνήθως δεν είναι ορατές με το μάτι. Όμως, η επιλογή αλγόριθμου συμπίεσης ιατρικών εικόνων είναι ένα αρκετά περίπλοκο ζήτημα δεδομένων των εξής γεγονότων:

1. Υπάρχει χαμηλή ανοχή σε απώλεια πληροφορίας λόγω της διαγνωστικής ποιότητας που απαιτείται για τις εικόνες. 2. Λόγω του ότι γίνεται εκλεκτική απεικόνιση μίας εικόνας με χρήση λιγότερων αποχρώσεων του γκρι, η πληροφορία που μπορεί να γίνει ορατή στο μάτι ποικίλει ανάλογα με τον αλγόριθμο απεικόνισης και επεξεργασίας. Συνεπώς, υπάρχει μεγάλη ευαισθησία στην επιλογή πληροφορίας που θα χαθεί με τη συμπίεση.
2. Επιπρόσθετα, οι χρήσιμες πληροφορίες σε κάθε είδος εικόνας είναι διαφορετικές. Άλλη είναι η πληροφορία που βλέπει ο γιατρός σε μία ακτινογραφία θώρακα και άλλη σε ένα υπερηχογράφημα. Συνεπώς, ανάλογα με το είδος της εικόνας διαφέρει και η πληροφορία που πρέπει να διατηρηθεί. Ένας αλγόριθμος συμπίεσης μπορεί να λαμβάνει υπόψη του ποιες πληροφορίες πρέπει να διατηρήσει αναλλοίωτες και πού μπορεί ενδεχομένως να έχει απώλειες ανάλογα με το είδος της εικόνας. 4. Τέλος, το τελικό μέγεθος της εικόνας εξαρτάται από το είδος της εικόνας. Συνεπώς, έχει διαφορετικές απαιτήσεις συμπίεσης.

1.5. Κωδικοποίηση εντροπίας και κωδικοποίηση πηγής

Μια απλοποιημένη ταξινόμηση των τεχνικών συμπίεσης είναι η εξής: κωδικοποίηση εντροπίας (entropy encoding) και κωδικοποίηση πηγής (source encoding)¹.

¹Στην ορολογία της ψηφιακής επεξεργασίας σήματος, ο όρος “κωδικοποίηση” περιλαμβάνει και την έννοια της συμπίεσης.

1.5.1. Κωδικοποίηση εντροπίας

Η κωδικοποίηση εντροπίας αναφέρεται σε τεχνικές, οι οποίες δεν λαμβάνουν υπ' όψη τους το είδος της πληροφορίας που πρόκειται να συμπιεστεί. Με άλλα λόγια, αυτές οι τεχνικές αντιμετωπίζουν την πληροφορία ως μια απλή ακολουθία από bits. Γι' αυτό το λόγο, η κωδικοποίηση εντροπίας μπορεί να εφαρμοστεί ανεξάρτητα από το είδος της πληροφορίας. Επιπλέον, οι τεχνικές κωδικοποίησης εντροπίας προσφέρουν κωδικοποίηση χωρίς απώλειες. Μπορούμε να αντικαθιστούμε κάθε ακολουθία 10 διαδοχικών μηδενικών που βρίσκουμε με έναν ειδικό χαρακτήρα ακολουθούμενο από τον αριθμό 10. Με αυτόν τον τρόπο, μειώνουμε το μήκος της ακολουθίας χωρίς να κάνουμε καμία υπόθεση για τη σημασία των μηδενικών, αλλά και χωρίς να αλλοιώνεται το σήμα. Οι τεχνικές κωδικοποίησης εντροπίας διαχωρίζονται σε δύο βασικές κατηγορίες:

1.5.1.1. Περιορισμός των ακολουθιών επαναλαμβανόμενων χαρακτήρων

Αυτή η μέθοδος κωδικοποίησης εντροπίας είναι από τις παλαιότερες και πιο απλές που χρησιμοποιούνται. Η ιδέα είναι ότι σε μια τυχαία ακολουθία από bits είναι πιθανό να εμφανιστούν κάποια τμήματα που αποτελούνται από κάποιο επαναλαμβανόμενο χαρακτήρα². Αυτά τα τμήματα μπορούν να αντικατασταθούν από αυτό το χαρακτήρα, έναν ειδικό χαρακτήρα που ονομάζεται σημαία και το πλήθος των επαναλήψεων του χαρακτήρα. Η κωδικοποίηση αυτή έχει την παρακάτω σημασία: Κάθε φορά που συναντάται η σημαία, ο χαρακτήρας που προηγείται αυτής πρέπει να επαναληφθεί όσες φορές υποδεικνύει ο αριθμός που ακολουθεί τη σημαία. Αυτή η μορφή που περιγράψαμε μπορεί να γίνει πιο αποδοτική, αν έχουμε συχνά εμφανιζόμενες ακολουθίες μηδενικών. Σ' αυτές τις περιπτώσεις απαιτείται απλώς μια σημαία (που θα σημαίνει “ επαναλαμβανόμενα μηδενικά” και ο αριθμός των επαναλήψεων. Και στις δύο περιπτώσεις, το μήκος των ακολουθιών πρέπει να είναι τέτοιο, ώστε να υπάρχει ουσιαστικό όφελος από αυτήν την αντικατάσταση.

1.5.1.2. Στατιστική Κωδικοποίηση (Statistical encoding)

Η στατιστική κωδικοποίηση είναι μια μέθοδος που χρησιμοποιείται πολύ συχνά. Η βασική αρχή αυτής της τεχνικής βρίσκεται στον εντοπισμό των πιο συχνά 2Υποθέτουμε ότι η ακολουθία από bits που αποτελεί την πληροφορία ομαδοποιείται σε χαρακτήρες ή οκτάδες από bits. Οι χαρακτήρες αποτελούν το ελάχιστο ποσό πληροφορίας. Υπάρχουν βέβαια και κάποιοι αλγόριθμοι συμπίεσης που λειτουργούν με άλλες ομαδοποιήσεις.

εμφανιζόμενων ακολουθιών χαρακτήρων και στην κωδικοποίησή τους με λιγότερα bits. Δηλαδή οι σπάνια εμφανιζόμενες ακολουθίες θα έχουν μεγαλύτερους κωδικούς, ενώ οι συχνές μικρότερους. Είναι φανερό ότι η μέθοδος απαιτεί την ύπαρξη λεξικού, όπου αποθηκεύονται οι ακολουθίες που αντιστοιχούν σε κάθε κωδικό για να μπορεί να γίνει η αποσυμπίεση. Καθοριστικής σημασίας για την επίδοση του αλγορίθμου είναι η στατιστική επεξεργασία των δεδομένων, για την ανεύρεση των ακολουθιών που θα κωδικοποιηθούν με μικρούς κωδικούς. Στην απλούστερη περίπτωση, το λεξικό είναι σταθερό, ενώ στην πιο σύνθετη το βρίσκουμε κάθε φορά που γίνεται η συμπίεση κάποιας ποσότητας δεδομένων. Η στατιστική κωδικοποίηση παίρνει δύο μορφές: αντικατάσταση προτύπων (pattern substitution) και κωδικοποίηση Huffman (Huffman encoding).

1.5.1.3. Αντικατάσταση προτύπων

Η μέθοδος της αντικατάστασης προτύπων χρησιμοποιείται αποκλειστικά για κείμενα. Συχνά εμφανιζόμενα πρότυπα (ακολουθίες χαρακτήρων, λέξεις) αντικαθιστώνται με λίγους χαρακτήρες. Για παράδειγμα, θα μπορούσαμε να κωδικοποιήσουμε αυτές τις σημειώσεις αντικαθιστώντας τη λέξη “πολυμέσα” με τους χαρακτήρες “*π”. Σε μια τέτοια περίπτωση, το λεξικό προκύπτει από ανάλυση του κειμένου, ενώ κάποιες λέξεις είναι εκ των προτέρων γνωστό ότι θα εμφανιστούν σίγουρα.

1.5.1.4. Κωδικοποίηση Huffman

Η κωδικοποίηση Huffman αποτελεί μια γενίκευση της στατιστικής κωδικοποίησης. Για κάποιο ρεύμα δεδομένων³ υπολογίζεται η συχνότητα εμφάνισης κάθε χαρακτήρα. Από αυτή τη συχνότητα, ο αλγόριθμος του Huffman υπολογίζει το ελάχιστο μήκος κωδικού που πρέπει να δοθεί σε κάθε χαρακτήρα και πραγματοποιεί τη βέλτιστη ανάθεση κωδικών. Αυτοί οι κωδικοί αποθηκεύονται στο λεξικό. Η μέθοδος του Huffman χρησιμοποιείται στη συμπίεση ακίνητης και κινούμενης εικόνας. Ανάλογα με τις λεπτομέρειες της υλοποίησης, για κάθε εικόνα ή ομάδα εικόνων δημιουργείται ένα νέο λεξικό. Στην περίπτωση της κινούμενης εικόνας, το λεξικό μπορεί να επαναδημιουργείται για κάθε πλαίσιο ή σειρά πλαισίων. Σε κάθε περίπτωση, η διαδικασία συμπίεσης πρέπει να αποθηκεύει το λεξικό για να είναι δυνατή η αποσυμπίεση.

³ Ως ρεύμα (stream) ορίζουμε μια άπειρη ακολουθία στοιχείων.

1.5.1.5. Κωδικοποίηση πηγής

Η διαφορά αυτής της τεχνικής είναι ότι οι μετασχηματισμοί τους οποίους υφίσταται το αρχικό σήμα εξαρτώνται άμεσα από τον τύπο του. Για παράδειγμα, ο λόγος χαρακτηρίζεται από συχνά διαστήματα σιωπής που μπορούν να περιγραφούν με πιο αποτελεσματικό τρόπο. Δηλαδή, οι μετασχηματισμοί του σήματος κάνουν χρήση των ιδιαίτερων σημασιολογικών χαρακτηριστικών που μεταφέρει το σήμα. Γενικά, αυτές οι τεχνικές μπορούν να παράγουν μεγαλύτερα ποσοστά συμπίεσης σε σχέση με την κωδικοποίηση εντροπίας. Μειονεκτούν όμως στη σταθερότητα, γιατί το ποσοστό συμπίεσης που επιτυγχάνουν διαφοροποιείται ανάλογα με το αντικείμενο που συμπιέζεται. Πάντως, η κωδικοποίηση πηγής μπορεί να λειτουργήσει και με απώλειες και χωρίς απώλειες.

Οι τεχνικές κωδικοποίησης πηγής διακρίνονται σε τρεις τύπους: Κωδικοποίηση μετασχηματισμού (transform encoding), Διαφορική ή προβλεπτική κωδικοποίηση (differential or predictive encoding) και Διανυσματική κβαντοποίηση (vector quantization).

1.5.1.6. Κωδικοποίηση μετασχηματισμού (transform encoding)

Η κωδικοποίηση μετασχηματισμού είναι ο πρώτος τύπος κωδικοποίησης πηγής που εξετάζουμε. Όπως έχουμε εξηγήσει, η κωδικοποίηση πηγής λαμβάνει υπ' όψη και τις ιδιότητες του σήματος που πρόκειται να συμπεστεί. Η κωδικοποίηση μετασχηματισμού χρησιμοποιείται συνήθως στη συμπίεση εικόνων. Η βασική της αρχή είναι η εξής: το σήμα υφίσταται ένα μαθηματικό μετασχηματισμό από το αρχικό πεδίο του χρόνου ή του χώρου σε ένα αφηρημένο πεδίο το οποίο είναι πιο κατάλληλο για συμπίεση.

Αυτή η διαδικασία είναι αντιστρεπτή, δηλαδή υπάρχει ο αντίστροφος μετασχηματισμός που θα επαναφέρει το σήμα στην αρχική του μορφή. Ένας τέτοιος μετασχηματισμός είναι ο μετασχηματισμός Fourier. Μέσω του μετασχηματισμού Fourier μια συνάρτηση του χρόνου $f(t)$ μπορεί να μετασχηματιστεί σε μια $g(\lambda)$ στο πεδίο των συχνοτήτων. Η νέα αυτή συνάρτηση παρέχει το πλάτος (ή συντελεστή) g των συχνοτήτων λ που απαρτίζουν την αρχική συνάρτηση.

Στην περίπτωση των εικόνων χρησιμοποιείται μια ειδική μορφή του μετασχηματισμού Fourier, ο διακριτός συνημιτονικός μετασχηματισμός Fourier, και το σημαντικό σημείο που εκμεταλλευόμαστε είναι το εξής: Στη φασματική (στο πεδίο των συχνοτήτων) αναπαράσταση των εικόνων, οι συχνότητες περιγράφουν πόσο γρήγορα μεταβάλλονται τα χρώματα και η απόλυτη φωτεινότητα. Εκτός από τον μετασχηματισμό Fourier υπάρχουν και άλλοι, όπως οι μετασχηματισμοί των Hadamar, Haar και των Karhunen-Loeve. Ανάλογα με τις ιδιότητες του τύπου της πληροφορίας που θέλουμε να συμπίεσουμε, επιλέγουμε και τον καταλληλότερο μετασχηματισμό. Αφού επιλεγεί και εκτελεστεί ο μετασχηματισμός, βρίσκουμε τους πιο σημαντικούς από τους συντελεστές και τους περιγράφουμε με μεγάλη ακρίβεια. Τους λιγότερο σημαντικούς μπορούμε να τους περιγράψουμε με μικρότερη ακρίβεια ή και να τους αγνοήσουμε τελείως. Κάνοντας κάτι τέτοιο η διαδικασία συμπίεσης έχει απώλειες. Παρ' όλα αυτά, οι μετασχηματισμοί από μόνοι τους είναι αντιστρεπτοί.

Κεφάλαιο 2 Κωδικοποίηση Huffman

2.1. Παρουσίαση του αλγορίθμου

Ο Huffman δημοσίευσε τις έρευνές του πάνω στο θέμα της κωδικοποίησης το 1952 παρουσιάζοντας έναν άπληστο αλγόριθμο για τον υπολογισμό ενός βέλτιστου κώδικα προθεμάτων, ο οποίος ονομάζεται κώδικας Huffman. Αμέσως έγινε η πιο πολυαναφερόμενη θεωρία της πληροφορίας. Είναι πιθανόν να είναι ακόμη. Η αρχική δουλειά του Huffman γέννησε πολυάριθμες ασήμαντες παραλλαγές. Ήταν ο επικρατέστερος στον κόσμο της κωδικοποίησης μέχρι τις αρχές του 1980.

Η συμπίεση δεδομένων αποτελείται από ένα χείμαρρο συμβόλων τα οποία μετατρέπονται σε κώδικες. Τα δεδομένα των αρχείων που συμπιέζει πρέπει να έχουν χαμηλή εντροπία (αταξία). Αυτό σημαίνει ότι η πιθανότητα εμφάνισης ενός συμβόλου αυξάνεται και συνεπώς αυξάνεται και η συμπίεση των δεδομένων.

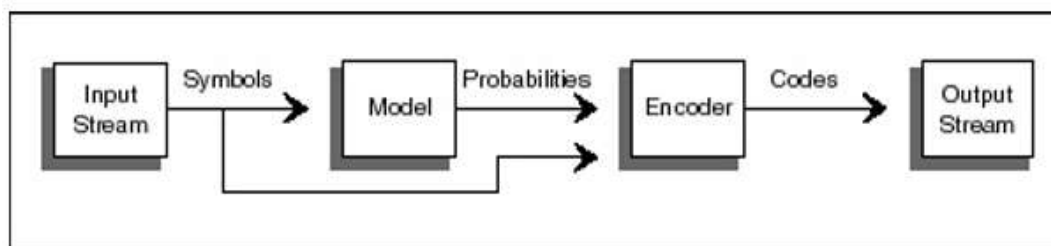
Η απόφαση να βγάλεις έναν βέβαιο κωδικό για ένα σύμβολο ή για μια ομάδα συμβόλων βασίζεται σε ένα μοντέλο. Το μοντέλο αυτό είναι απλά μια συλλογή από δεδομένα και κανόνες βασισμένα στη διαδικασία της εισαγωγής συμβόλων και του προσδιορισμού κάθε κωδικού σε έξοδο. Δεν παράγει έναν κωδικό για κάθε σύμβολο, αλλά για ολόκληρο το μήνυμα παράγει έναν κωδικό.

Κάθε σύμβολο προστίθεται στο μήνυμα προσαυξητικά τροποποιώντας τον κωδικό της εξόδου. Τέτοια αρχεία είναι τα αρχεία κειμένου και τα αρχεία εικόνων. Αυτό είναι μια βελτίωση γιατί η επίδραση του κάθε εισηγμένου συμβόλου στο τελικό αποτέλεσμα μπορεί να είναι αμελητέος αριθμός από bits αντί για ακέραιο αριθμό. Έτσι, λοιπόν, αν η εντροπία του χαρακτήρα 'e' είναι 2,5 bits είναι πιθανόν να προστεθούν ακριβώς 2,5 bits στον κώδικα της εξόδου.

Αν η συμπίεση είναι αποτελεσματική, οι κώδικες των συμβόλων θα είναι μικρότεροι από τα αρχικά σύμβολα. Η μέγιστη απόδοσή του είναι στις ασπρόμαυρες φωτογραφίες (φωτοτυπημένες εικόνες ή αρχεία FAX). Η μέθοδος είναι βέλτιστη, όταν οι πιθανότητες εμφάνισης των συμβόλων είναι δυνάμεις του y . Η πιθανότητα εμφάνισης ενός συμβόλου δίνεται από τη σχέση: $P(\text{συμβόλου}) = \text{Συχνότητα εμφάνισης ενός συμβόλου} / \text{Σύνολο των συμβόλων του αρχείου}$

2.2. Συμπίεση Δεδομένων: Μοντελοποίηση και Κωδικοποίηση

Μοντελοποίηση και κωδικοποίηση είναι δύο εντελώς διαφορετικά πράγματα. Οι άνθρωποι συχνά χρησιμοποιούν τον όρο κωδικοποίηση για να αναφερθούν σε ολόκληρη τη διαδικασία της συμπίεσης δεδομένων αντί για μόνο μια συνιστώσα της διαδικασίας. Η φράση «Huffman Coding» στην περιγραφή μιας τεχνικής συμπίεσης δεδομένων στην πραγματικότητα είναι μέθοδος συμπίεσης που συνδέει ένα μοντέλο με τη συμπίεση δεδομένων. Μια ανάλυση της διαδικασίας κωδικοποίησης Huffman μοιάζει σαν το παρακάτω σχήμα.



Στην περίπτωση της κωδικοποίησης Huffman τα πραγματικά αποτελέσματα του κωδικοποιητή προσδιορίζονται από ένα σύνολο πιθανοτήτων. Χρησιμοποιώντας αυτό το είδος κωδικοποίησης ένα σύμβολο που έχει πολύ υψηλή πιθανότητα ύπαρξης παράγει ένα κώδικα με πολύ λίγα bits.

Ένα σύμβολο με χαμηλή πιθανότητα ύπαρξης παράγει ένα κώδικα με μεγάλο αριθμό bits. Σκεφτόμαστε το μοντέλο και τη διαδικασία κωδικοποίησης διαφορετικά εξαιτίας των αναρίθμητων τρόπων μοντελοποίησης δεδομένων τα οποία μπορούν όλα να χρησιμοποιήσουν τις ίδιες διαδικασίες κωδικοποίησης για να παράγουν τα αποτελέσματά τους. Ένα απλό πρόγραμμα που μπορεί να χρησιμοποιεί αυτή την κωδικοποίηση μπορεί να χρησιμοποιήσει ένα μοντέλο που θα δίνει την πιθανότητα εμφάνισης του κάθε συμβόλου για οποιαδήποτε είσοδο συμβόλων. Ένα πιο εξελιγμένο πρόγραμμα μπορεί να υπολογίσει την πιθανότητα βασιζόμενο στα τελευταία 10 σύμβολα των συμβόλων που εισήχθησαν. Ακόμη και αν τα δύο προγράμματα χρησιμοποιούν την κωδικοποίηση Huffman για να παράγουν τα αποτελέσματά τους, οι αναλογίες συμπίεσης μπορεί να είναι ριζικά διαφορετικές. Η κωδικοποίηση Huffman πετυχαίνει το ελάχιστο δυνατό ποσό πλεονασμού σε ένα μείγμα από κωδικούς με κυμαινόμενο μήκος. Αυτό δεν σημαίνει πως είναι ο ευνοϊκότερος τρόπος κωδικοποίησης. Παρέχει καλύτερη διαδικασία προσέγγισης για κωδικοποίηση συμβόλων όταν χρησιμοποιούνται αμετάβλητα μήκη κωδικών.

Το πρόβλημα με την Huffman ή την Shannon - Fano κωδικοποίηση είναι ότι χρησιμοποιούν έναν ακέραιο αριθμό από bits σε κάθε κώδικα. Αν η εντροπία ενός χαρακτήρα είναι 2,5 bits με βάση την κωδικοποίηση Huffman ο χαρακτήρας θα πρέπει να έχει εντροπία μεταξύ του 2 ή του 3 και όχι 2,5. Γι' αυτό το λόγο η κωδικοποίηση αυτή δεν μπορεί να θεωρηθεί η βέλτιστη, αλλά είναι η καλύτερη διαδικασία προσέγγισης κωδικών με σταθερό αριθμό από bits. Ακολουθεί ένα δείγμα της κωδικοποίησης Huffman.

Symbol	Huffman Code
E	100
T	101
A	1100
I	11010
...	...
X	01101111
Q	01101110001
Z	01101110000

Ένα παράδειγμα, για να γίνει πιο κατανοητό, φαίνεται στον ακόλουθο πίνακα με την ανάλυση ενός φανταστικού μηνύματος. Η κωδικοποίηση Huffman αποδίδει 89 bits το μήκος το μηνύματος. Αλλά αριθμητικά ο κώδικας μπορεί να προσεγγίσει την πραγματική πληροφορία του μηνύματος με 83,56 bits. Η διαφορά ανάμεσα σε αυτά τα δυο μηνύματα είναι περίπου 6 τοις εκατό.

2.3. Εφαρμογή αλγορίθμου

Ο αλγόριθμος του Huffman υπολογίζει το δέντρο T που αντιστοιχεί σε έναν βέλτιστο κώδικα προχωρώντας από τα φύλλα προς τη ρίζα. Ο αλγόριθμος ξεκινά από ένα σύνολο C απομονωμένων φύλλων, τα οποία ενώνει σε ένα πλήρες δυαδικό με την εισαγωγή ακριβώς C - 1 εσωτερικών κόμβων.

Σε κάθε εσωτερικό κόμβο αντιστοιχούμε συχνότητα ίση με το άθροισμα των συχνοτήτων όλων των χαρακτήρων / φύλλων του υποδέντρου. Σε κάθε βήμα, τα δύο υποδέντρα με τις μικρότερες συχνότητες ενώνονται και αντικαθίστανται από ένα υποδέντρο που δημιουργείται με την προσθήκη ενός νέου εσωτερικού κόμβου.

Στον νέο κόμβο αντιστοιχούμε συχνότητα ίση με το άθροισμα των συχνοτήτων των ριζών των δύο υποδέντρων. Ο αλγόριθμος ολοκληρώνεται όταν απομείνει μόνο ένα δέντρο που περιέχει όλους τους χαρακτήρες του C σαν φύλλα. Ο αλγόριθμος λειτουργεί ως εξής:

Βήμα 1: Υπολογισμός των πιθανοτήτων εμφάνισης όλων των συμβόλων.

Βήμα 2: Πρόσθεση των δυο μικρότερων πιθανοτήτων.

Βήμα 3: Οι πιθανότητες των συμβόλων που απομένουν και οι πιθανότητες που προκύπτουν από αθροίσματα, συμμετέχουν στην επιλογή ξανά.

Βήμα 4: Επανάληψη των δυο προηγούμενων βημάτων μέχρι το άθροισμα να γίνει ίσο με τη μονάδα.

Κεφάλαιο 3 ΚΩΔΙΚΟΠΟΙΗΣΗ ΚΑΝΑΛΙΟΥ

Με τον όρο κωδικοποίηση καναλιού αναφερόμαστε στην εισαγωγή, πριν τη μετάδοση (ή εν γένει την περαιτέρω επεξεργασία) πλεονάζουσας πληροφορίας, η οποία χρησιμοποιείται στον δέκτη, κατά τις αντίστοιχες διαδικασίες αποκωδικοποίησης καναλιού, ώστε να εντοπίσει και εάν είναι δυνατόν να διορθώσει σφάλματα τα οποία έχουν συμβεί κατά τη μετάδοση (ή την οποιαδήποτε επεξεργασία).

Ο στόχος είναι να αυξήσουμε την αξιοπιστία του συστήματος και να βελτιώσουμε την ποιότητα του αναπαραγόμενου υλικού στον δέκτη (μέσω της μείωσης των σφαλμάτων που αναπόφευκτα έχουν προκύψει κατά τη μετάδοση). Λόγω ακριβώς της εισαγωγής της πλεονάζουσας πληροφορίας αυξάνει ο όγκος των δεδομένων σε σχέση με τον αρχικό όγκο της πληροφορίας. Οι λειτουργίες της κωδικοποίησης καναλιού εντάσσονται στη λογική της προληπτικής διόρθωσης σφαλμάτων (Forward Error Correction, FEC).

Τέτοιες τεχνικές χρησιμοποιούνται σε περιπτώσεις όπου είναι πιθανή η εμφάνιση σφαλμάτων, είτε εμπλέκεται μετάδοση της πληροφορίας είτε όχι (για παράδειγμα, χρησιμοποιούνται και στην περίπτωση της αποθήκευσης της πληροφορίας). Η αλληλουχία των ενεργειών παρουσιάζεται στην επόμενη εικόνα.



Οι μηχανισμοί προληπτικής διόρθωσης σφαλμάτων δεν χαρακτηρίζουν μόνο την ψηφιακή τηλεόραση, αφού χρησιμοποιούνται εκτενώς για τη μετάδοση αλλά και την αποθήκευση δεδομένων. Ο πιο γνωστός ίσως μηχανισμός προληπτικής διόρθωσης σφαλμάτων είναι ο έλεγχος ισοτιμίας (parity check). Κατά τον έλεγχο αυτό προστίθεται ένα bit ισοτιμίας ώστε να θέτουμε το συνολικό πλήθος των bits της λέξης ως περιττό ή άρτιο αριθμό. Αν το συνολικό πλήθος των bits με τιμή 1, είναι

περιττό μιλάμε για περιττή ισοτιμία, ενώ αν είναι άρτιο μιλάμε για άρτια ισοτιμία (odd και even parity αντίστοιχα). Το είδος της ισοτιμίας συμφωνείται εκ των προτέρων μεταξύ του πομπού και του δέκτη (ή εν γένει του αποστολέα και του παραλήπτη). Είναι σαφές ότι ο μηχανισμός αυτός είναι πολύ απλός και μπορεί να εντοπίσει περιττό αριθμό σφαλμάτων, αφού στην περίπτωση του ζυγού αριθμού σφαλμάτων αυτά αλληλοαναιρούνται.

Παρόλο που πλέον χρησιμοποιείται για λόγους συμβατότητας μπορεί να μας δώσει το γενικό πλαίσιο λειτουργίας των μηχανισμών προληπτικής διόρθωσης σφαλμάτων. Άλλο ένα εξίσου απλό παράδειγμα τέτοιου μηχανισμού είναι η αποστολή της ίδιας πληροφορίας περισσότερες από μία φορές. Για παράδειγμα, κάθε bit μπορεί να αποστέλλεται τρεις φορές αντί για μία.

Ο δέκτης αποφασίζει την τιμή 0 ή 1 κατά πλειοψηφία. Στην περίπτωση αυτή αντιμετωπίζεται (εντοπίζεται και διορθώνεται) η ύπαρξη ενός μόνο σφάλματος. Από την άλλη, βέβαια, αυξάνεται δραματικά ο όγκος των δεδομένων, στην περίπτωσή μας, δε, τριπλασιάζεται. Στον επόμενο πίνακα παρουσιάζεται μια μέθοδος προστασίας από σφάλματα με τον τριπλασιασμό των δεδομένων.

Παρατηρούμε τις τιμές της κωδικοποιημένης πληροφορίας και τι συμβαίνει στην περίπτωση που έχουμε ένα ή κανένα σφάλμα. Στην περίπτωση δύο σφαλμάτων, ο αλγόριθμος δεν δύναται να τα αντιμετωπίσει (θεωρεί, δηλαδή, ότι έχει σταλεί διαφορετική τιμή).

Τιμή προς αποστολή	Κωδικοποιημένο bit προς αποστολή	Διάλογος	Λαμβανόμενη πληροφορία	Μεταφραζόμενη πληροφορία
0	000	Κανένα σφάλμα	000	0
		Ένα σφάλμα	001	0
		Ένα σφάλμα	010	0
		Ένα σφάλμα	100	0
1	111	Κανένα σφάλμα	111	1
		Ένα σφάλμα	110	1
		Ένα σφάλμα	101	1
		Ένα σφάλμα	011	1

Στην κωδικοποίηση καναλιού, οι μηχανισμοί που χρησιμοποιούνται διακρίνονται σε δύο μεγάλες κατηγορίες, ανάλογα κυρίως με το τμήμα της πληροφορίας όπου εφαρμόζονται, καθώς και τον τρόπο με τον οποίο προστίθεται η πλεονάζουσα πληροφορία. Οι κατηγορίες αυτές είναι οι ακόλουθες:

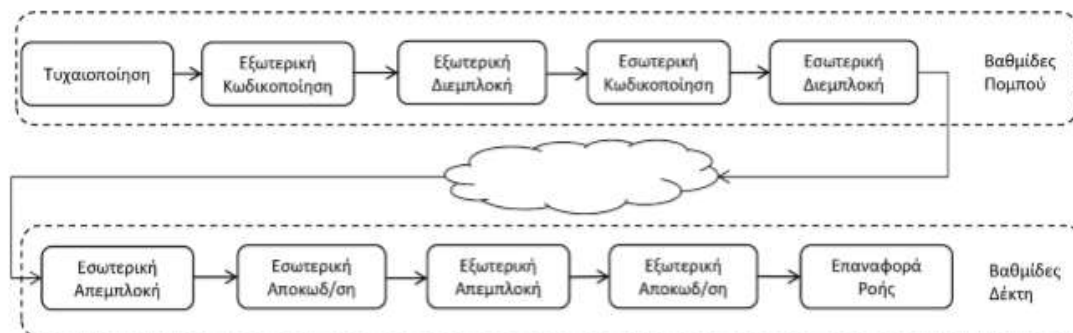
- Κώδικες μπλοκ (block code)
- Συνελικτικοί κώδικες (convolutional code)

Η σημαντικότερη διαφορά μεταξύ αυτών των κατηγοριών εντοπίζεται στο γεγονός ότι οι πρώτοι χωρίζουν την πληροφορία σε τμήματα, συγκεκριμένου μεγέθους (μπλοκ). Στα τμήματα αυτά γίνεται η κωδικοποίηση για να ληφθούν οι αντίστοιχες κωδικές λέξεις (κωδικολέξη). Κάτι τέτοιο δεν συμβαίνει στην περίπτωση των συνελικτικών μηχανισμών, αφού η κωδικοποίηση και η αποκωδικοποίηση λαμβάνουν χώρα στη συνολική ροή (συνήθως σε επίπεδο bit). Παράλληλα με την ανωτέρω διαφοροποίηση, υπάρχει ακόμα μία η οποία αφορά τη μνήμη. Συγκεκριμένα, οι συνελικτικοί κωδικοποιητές έχουν κατάσταση (άρα μνήμη). Αυτό σημαίνει ότι η έξοδος δεν εξαρτάται μόνο από την τιμή της εισόδου τη συγκεκριμένη χρονική στιγμή αλλά και από αντίστοιχες τιμές σε προηγούμενο χρόνο. Αυτό δεν συμβαίνει στους κώδικες τύπου μπλοκ, όπου κάθε κωδικολέξη είναι ανεξάρτητη από την προηγούμενη και την επόμενη. Η σύγκριση μεταξύ των δύο μηχανισμών παρουσιάζεται στον επόμενο πίνακα.

Κριτήριο	Κωδικοποίηση Τμήματος Δεδομένων (blockcode)	Συνελικτική κωδικοποίηση (Convolutional code)
Δεδομένα στα οποία εφαρμόζεται	Τα δεδομένα που ανήκουν στο τρέχον μπλοκ (τμήμα), χωρίς συσχέτιση με προηγούμενα δεδομένα	Τρέχοντα δεδομένα της ροής καθώς αυτά εισέρχονται στον κωδικοποιητή
Μνήμη	Όχι (κάθε μπλοκ πληροφορίας κωδικοποιείται ανεξάρτητα)	Ναι (η κωδικοποίηση των δεδομένων συνήθως εξαρτάται και από τα προηγούμενα δεδομένα)
Συμπεριφορά στις ριπές σφαλμάτων	Μη ικανοποιητική	Μη ικανοποιητική

Όπως θα δούμε στη συνέχεια, η απόδοση και των δύο τύπων κωδικοποίησης επηρεάζεται στην περίπτωση που έχουμε ριπές θορύβου. Για την αντιμετώπιση αυτού του χαρακτηριστικού λαμβάνει χώρα ένα ανακάτεμα, αντιμετάθεση ή αλλιώς διεμπλοκή (interleaving) της πληροφορίας, πριν την αποστολή. Η πληροφορία επανέρχεται στη σωστή σειρά στον αποκωδικοποιητή με τη διαδικασία που ονομάζουμε απεμπλοκή (de-interleaving).

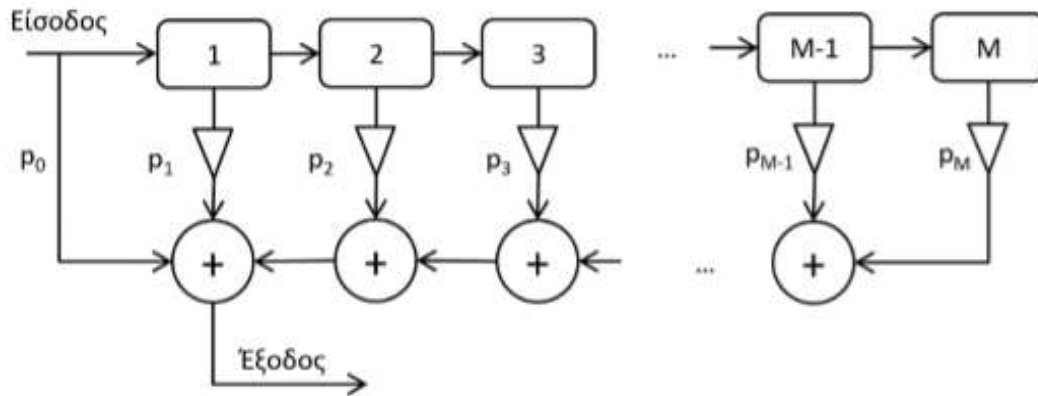
Στην περίπτωση της ψηφιακής τηλεόρασης χρησιμοποιούνται και οι δύο τύποι κωδικοποίησης (μπλοκ και συνελκτικοί), διαδοχικά, όπως φαίνεται στην επόμενη εικόνα. Μεταξύ αυτών λαμβάνει χώρα η διεμπλοκή για τους λόγους που προαναφέρθηκαν (δηλαδή την αντιμετώπιση ριπών σφαλμάτων). Κατά την αποκωδικοποίηση στον δέκτη λαμβάνουν χώρα οι αντίστροφες διαδικασίες.



Οι λειτουργίες που λαμβάνουν χώρα περιλαμβάνουν τα επόμενα στάδια: (α) τυχαιοποίηση (randomization) των δεδομένων (η οποία επιτυγχάνει διασπορά της ενέργειας), (β) εξωτερική κωδικοποίηση, (γ) εξωτερική διεμπλοκή, (δ) εσωτερική κωδικοποίηση και (ε) εσωτερική διεμπλοκή από την πλευρά του πομπού. Αντίστοιχα, από την πλευρά του δέκτη, έχουμε κατά σειρά: (α) την εσωτερική απεμπλοκή, (β) την εσωτερική αποκωδικοποίηση, (γ) την εξωτερική απεμπλοκή, (δ) την εξωτερική αποκωδικοποίηση και (ε) την επαναφορά της ροής.

3.1. Τυχαιοποίηση - Διασπορά Ενέργειας

Βαθμίδα Τυχαιοποίησης Η πρώτη βαθμίδα του μηχανισμού της κωδικοποίησης καναλιού έχει να κάνει με τη διασπορά της ενέργειας στα δυαδικά δεδομένα. Η διασπορά της ενέργειας συνίσταται στην αποφυγή μεγάλου συνεχόμενου πλήθους από μηδέν ή ένα στα δεδομένα (ροή bits). Η ενέργεια αυτή ισοδυναμεί με την τυχαιοποίηση της σειράς (scrambling) των δεδομένων, την εισαγωγή, δηλαδή, κάποιας τυχαιότητας στα δεδομένα. Η διασπορά της ενέργειας επιτυγχάνεται με βαθμίδα τυχαιοποίησης η οποία εφαρμόζεται στη ροή από bits, όπως φαίνεται στην επόμενη εικόνα.

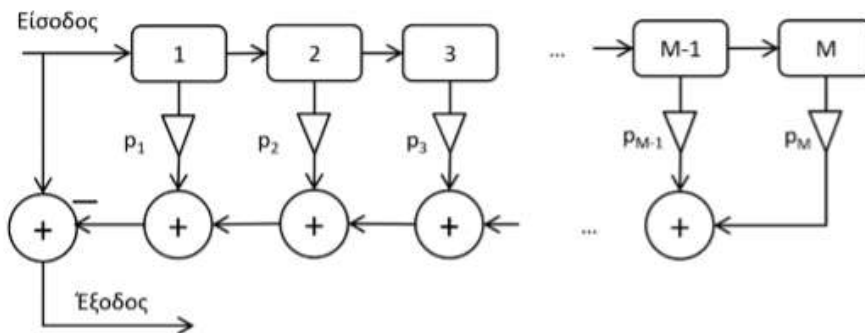


Η βαθμίδα τυχαιοποίησης αποτελείται από μια σειρά M καταχωρητών, αθροιστών και πολλαπλασιαστών. Η βάση υπολογισμού της βαθμίδας είναι N και οι τιμές της εισόδου κυμαίνονται στο διάστημα $[0, N-1]$. Σε αυτήν την περίπτωση οι αθροιστές έχουν βάση το N (είναι δηλαδή αθροιστές modulo N). Οι συντελεστές p_0, p_1, \dots, p_{M-1} και p_M καθορίζουν ποιες από τις τιμές των καταχωρητών λαμβάνονται υπόψη στα αθροίσματα. Το άθροισμα, δηλαδή, καθορίζεται από το αντίστοιχο πολυώνυμο γεννήτορα, το οποίο ορίζεται με χρήση των ανωτέρω συντελεστών:

$$p_0 + p_1z^{-1} + \dots + p_{M-1}z^{-(M-1)} + p_Mz^{-M} \quad (1)$$

Καθώς εισέρχεται το σήμα εισόδου στη βαθμίδα, το περιεχόμενο των καταχωρητών μετατοπίζεται προς τα δεξιά και γίνονται οι αντίστοιχοι υπολογισμοί της εξόδου. Άλλη μία παράμετρος η οποία έχει σημασία στους σχετικούς υπολογισμούς είναι η αρχική κατάσταση των καταχωρητών.

Η αντίστροφη διαδικασία γίνεται στον δέκτη στη βαθμίδα επαναφοράς της ροής (descrambler) ώστε να επανακτηθεί η αρχική ροή. Η διαδικασία που ακολουθείται παρουσιάζεται στο επόμενο σχήμα. Παρατηρούμε την ομοιότητα των δύο βαθμίδων με σημείο διαφοροποίησης τη μετατροπή του αθροίσματος σε διαφορά.



Στην περίπτωση της ψηφιακής τηλεόρασης, η είσοδος του συστήματος κωδικοποίησης καναλιού είναι ο Συρμός Μεταφοράς (MPEG2 Transport Stream, MPEG2-TS). Τα πακέτα του συρμού, όπως εξετάσαμε στο προηγούμενο κεφάλαιο, είναι σταθερού μήκους, συγκεκριμένα 188 Bytes. Το πρώτο Byte κάθε πακέτου είναι το Byte συγχρονισμού και έχει την τιμή 0x47. Η βαθμίδα τυχαιοποίησης (scrambler) αποτελείται από μια ψευδοτυχαία δυαδική ακολουθία των 16bit (Pseudo Random Binary Sequence, PRBS) με ένα πολώνυμο γεννήτορα 15ου βαθμού με τους επόμενους συντελεστές:

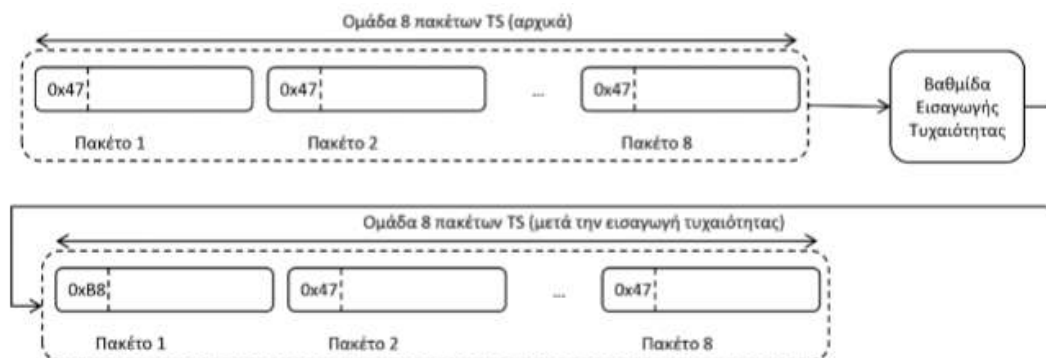
$$\begin{aligned}
 p_0 &= 1 \\
 p_{14} &= 1 \\
 p_{15} &= 1
 \end{aligned}
 \tag{2}$$

Στον καταχωρητή PRBS φορτώνεται η αρχική ακολουθία «1001010 1000 0000» και αρχικοποιείται εκ νέου με την έναρξη κάθε ομάδας 8 πακέτων MPEG2-TS.

Η τιμή του πρώτου Byte συγχρονισμού στην ομάδα των 8 πακέτων αντιστρέφεται δυαδικά (χωρίς, δηλαδή, να επηρεάζεται από τη λειτουργία της βαθμίδας) από 0x47 σε 0xB8. Η συγκεκριμένη αυτή τιμή χρησιμοποιείται για τον εντοπισμό των τυχαιοποιημένων ομάδων πακέτων. Η δυαδική αντιστροφή του Byte συγχρονισμού απεικονίζεται παρακάτω:

$$0x47 = 0100\ 0111_{\leftarrow} \qquad 10111000_{\leftarrow} = 0xB8$$

Αυτό σημαίνει ότι η συγκεκριμένη τιμή (0xB8) επαναλαμβάνεται περιοδικά μετά από 8 TS πακέτα, δηλαδή μετά από $8 \times 188 - 1 = 1.503$ Bytes.



Η βαθμίδα επαναφοράς της ακολουθίας πρέπει να είναι συγχρονισμένη με την αρχική, οπότε ο συγχρονισμός αυτός επιτυγχάνεται με τον μηχανισμό της αναστροφής του πρώτου Byte κάθε ομάδας 8 πακέτων όπως συζητήσαμε παραπάνω. Το πρώτο Byte των υπόλοιπων πακέτων της ομάδας δεν αντιστρέφεται, ούτε επηρεάζεται από τη λειτουργία της βαθμίδας και παραμένει στην τιμή 0x47.

3.2 Εξωτερική Κωδικοποίηση

Όπως συζητήθηκε στα προηγούμενα, η προληπτική διόρθωση σφαλμάτων (Forward Error Correction) με την προσθήκη πλεονάζουσας πληροφορίας στο σήμα από τον πομπό, δίνει τη δυνατότητα στον δέκτη να εντοπίσει και να διορθώσει σφάλματα τα οποία έχουν συμβεί κατά τη μετάδοση. Πολλές κωδικοποιήσεις έχουν σχεδιαστεί για αυτόν τον σκοπό με την κωδικοποίηση Reed Solomon να αποτελεί μια από τις πλέον συχνά χρησιμοποιούμενες, αφού ισορροπεί αρμονικά ανάμεσα στην απόδοση και στην πολυπλοκότητα. Η μεν απόδοση σχετίζεται με το πλήθος της πλεονάζουσας πληροφορίας που απαιτείται (και η οποία συνεπάγεται την αύξηση του όγκου), η δε πολυπλοκότητα σχετίζεται με τη δυσκολία στην κωδικοποίηση και την αποκωδικοποίηση.

Η κωδικοποίηση Reed Solomon χρησιμοποιείται στην επίγεια ψηφιακή τηλεόραση (DVB-T, ATSC) (40). Ο αλγόριθμος Reed Solomon είναι από τους πλέον συχνά χρησιμοποιούμενους αλγόριθμους και χρησιμοποιείται εκτός από την ψηφιακή τηλεόραση σε μια σειρά εφαρμογών συμπεριλαμβανομένων και των επομένων: CD, DVD, BlueDisc, DSL και WiMAX. Τόσο η κωδικοποίηση όσο και η αποκωδικοποίηση λαμβάνει χώρα με τη βοήθεια υλικού (hardware) για καλύτερη απόδοση. Η κωδικοποίηση Reed Solomon είναι κωδικοποίηση τύπου μπλοκ, όπου η πληροφορία χωρίζεται σε τμήματα συγκεκριμένου μήκους και σε αυτά τα τμήματα λαμβάνει χώρα η κωδικοποίηση και η αποκωδικοποίηση (ανεξάρτητα σε κάθε τμήμα).

Σε κάθε ένα από αυτά τα τμήματα προστίθεται ένα τμήμα προστασίας (parity information) και αυτό μαζί με την αρχική πληροφορία σχηματίζει ένα νέο μπλοκ το οποίο αποστέλλεται και κωδικοποιείται ανεξάρτητα από τα υπόλοιπα. Στην περίπτωση που η αρχική πληροφορία δεν υφίσταται κάποια αλλαγή στο νέο ενισχυμένο μπλοκ, δηλαδή απλώς προστίθεται το τμήμα των δεδομένων που αφορά

τον εντοπισμό και τη διόρθωση των σφαλμάτων είτε στο αρχικό τμήμα του μπλοκ είτε στο τέλος αυτού, τότε ο κώδικας ονομάζεται συστηματικός.

Αυτό σημαίνει ότι (πέρα από τον έλεγχο και τη διόρθωση των τυχόν σφαλμάτων) κατά την αποκωδικοποίηση το πλεονάζον αυτό τμήμα πληροφορίας αφαιρείται. Ο αλγόριθμος Reed Solomon είναι συστηματικός.

Ο RS είναι επιπλέον γραμμικός, δηλαδή η προσθήκη δύο κωδικών λέξεων τμημάτων δημιουργεί μια νέα κωδική λέξη. Επίσης είναι κυκλικός, με την έννοια ότι αν γίνει κυκλική μετατόπιση των συμβόλων η λέξη που θα προκύψει θα είναι επίσης μια κωδική λέξη. Η κωδικοποίηση RS ανήκει στην οικογένεια κωδικοποιήσεων Bose-Chaudhuri-Hocquenghem (BCH). Αν σε μια κωδική λέξη έστω και ένα από τα περιεχόμενα bits είναι εσφαλμένα, αυτό λογίζεται ως μια εσφαλμένη κωδική λέξη.

3.3. Εσωτερική Κωδικοποίηση

Η εσωτερική κωδικοποίηση αποσκοπεί στη διόρθωση λαθών και δρα συμπληρωματικά ως προς την κωδικοποίηση κατά Reed Solomon. Χρησιμοποιείται στο ψηφιακό βίντεο, στις ασύρματες και δορυφορικές επικοινωνίες. Η εσωτερική κωδικοποίηση ακολουθεί την εξωτερική διεμπλοκή. Πρόκειται περί συνελκτικής κωδικοποίησης (convolutional encoding) (45). Για τη μεν κωδικοποίηση χρησιμοποιείται ο κωδικοποιητής Trellis για την αποκωδικοποίηση ο αποκωδικοποιητής Viterbi. Επειδή η εσωτερική κωδικοποίηση κατά κανόνα δύναται να διπλασιάσει το πλήθος των δεδομένων, είναι δυνατόν να γίνει διάτρηση (puncturing) ώστε να μειωθεί ο ρυθμός, όπως θα φανεί στα επόμενα. Η λογική του συνελκτικού κωδικοποιητή είναι η χρήση καταχωρητών, κάθε ένας εκ των οποίων μπορεί να αποθηκεύσει ένα bit. Ο κωδικοποιητής έχει 2 αθροιστές (πύλες Exclusive Or – XOR), οι οποίες λαμβάνουν δεδομένα από κάποιους από τους καταχωρητές, χρησιμοποιώντας πολώνυμα – γεννήτορες. Υπενθυμίζεται το αποτέλεσμα της πράξης σε μία πύλη XOR, δύο εισόδων:

Είσοδος 1	Είσοδος 2	Έξοδος
0	0	0
0	1	1
1	0	1
1	1	0

Οι συνελκτικοί κωδικοποιητές είναι δυνατόν να χρησιμοποιούν ανάδραση ή όχι. Η μη χρήση ανάδρασης τους κάνει πιο απλούς στη δομή. Από συστημική άποψη οι κωδικοποιητές χωρίς ανάδραση μπορούν να θεωρηθούν ψηφιακά φίλτρα πεπερασμένης κρουστικής απόκρισης (FIR, Finite Impulse Response).

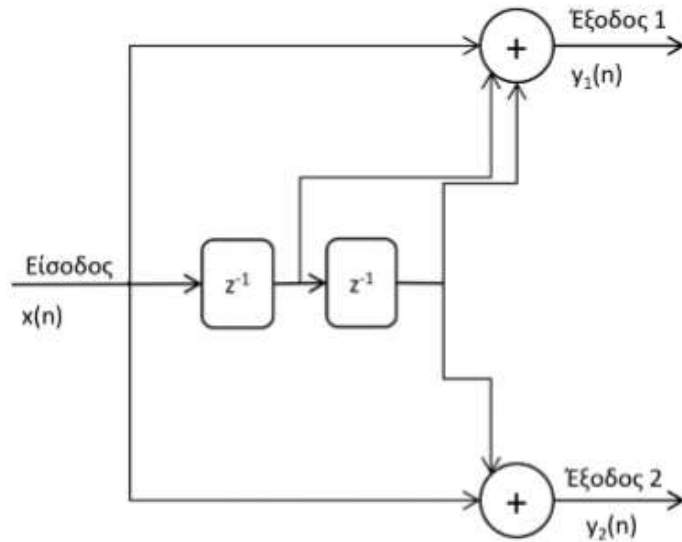
Οι κωδικοποιητές με ανάδραση είναι ψηφιακά φίλτρα μη πεπερασμένης κρουστικής απόκρισης (IIR, Infinite Impulse Response). Ένας άλλος τρόπος να θεωρήσουμε τον κωδικοποιητή είναι ως μηχανή πεπερασμένων καταστάσεων.

Η έξοδος προκύπτει από τις πράξεις που εκτελούνται στους αθροιστές και με χρήση των τιμών που είναι αποθηκευμένες στους καταχωρητές. Σε κάποιες περιπτώσεις μπορεί κάποια bits να τροφοδοτούν απευθείας την έξοδο. Το πλήθος των bits της εξόδου προκύπτει από το πλήθος της εισόδου επί το πλήθος των αθροιστών. Αυτό σημαίνει ότι στην τυπική περίπτωση που έχουμε δύο αθροιστές ο ρυθμός (σε bits per second) της εξόδου θα είναι διπλάσιος σε σχέση με αυτόν της εισόδου.

Ο συνελκτικός κωδικοποιητής χαρακτηρίζεται από το πλήθος των μνημών που διαθέτει. Το πλήθος των μνημών καθορίζει και το πλήθος των πιθανών καταστάσεων στις οποίες μπορεί να βρεθεί ο κωδικοποιητής. Δίνουμε κάποιους χρήσιμους ορισμούς:

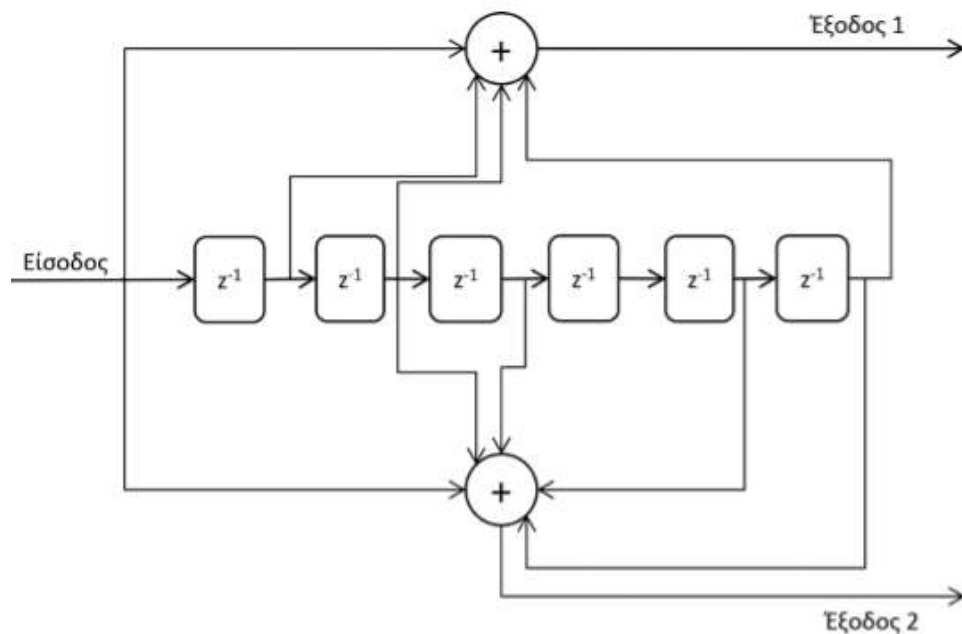
- Ρυθμός κώδικα (code rate) k/n , όπου k ο ρυθμός των εισερχόμενων δεδομένων στον κωδικοποιητή (σε bits / sec) και n ο ρυθμός των δεδομένων που εξέρχονται από τον κωδικοποιητή (σε bits / sec). Ο ρυθμός κώδικα εξαρτάται από το πλήθος των εξόδων του κωδικοποιητή. Η είσοδος θεωρούμε ότι αποτελείται από μία και μόνο ροή. Η βασική τιμή του ρυθμού κώδικα είναι $1/2$ (ένα προς δύο).
- Μήκος περιορισμού (constraint length) M , το οποίο αντιστοιχεί στις θέσεις μνήμης, δηλαδή, στο πλήθος bits που είναι αποθηκευμένα και επηρεάζουν την έξοδο του κωδικοποιητή. Το μήκος περιορισμού καθορίζει και τις καταστάσεις τις οποίες μπορεί να λάβει ο κωδικοποιητής.

Όπως και στην περίπτωση του κώδικα Reed Solomon, εάν τα δεδομένα της εισόδου χρησιμοποιούνται ως έχουν στην έξοδο, η κωδικοποίηση ονομάζεται συστηματική. Δεδομένου ότι ο εσωτερικός κωδικοποιητής ο οποίος προβλέπεται από το πρότυπο είναι αρκετά σύνθετος ώστε να δούμε απευθείας τον μηχανισμό λειτουργίας του, θα εξετάσουμε ένα πιο απλό και ευκολότερα παρουσιάσιμο παράδειγμα.



3.4. Συνελκτικός Κωδικοποιητής

Προτύπου του ETSI Ο τυπικός συνελκτικός κωδικοποιητής ο οποίος περιγράφεται στο σχετικό πρότυπο (46) παρίσταται στο επόμενο σχήμα. Η ύπαρξη 6 μνημών δημιουργεί 64 καταστάσεις.



Η συνελικτική κωδικοποίηση δεν εφαρμόζεται σε τμήματα (μπλοκ) δεδομένων αλλά σε συνεχόμενη ροή bit. Οπότε η δυνατότητα διόρθωσης σφαλμάτων εκ μέρους του κώδικα αφορά σφάλματα που λαμβάνουν χώρα κοντά το ένα με το άλλο. Στην περίπτωση μεγαλύτερων αποστάσεων μεταξύ των σφαλμάτων αυτά μπορούν να διορθωθούν σχετικά εύκολα. Για αυτό και στην περίπτωση αυτή πρέπει να γίνει διεμπλοκή των bits, την οποία θα εξετάσουμε στην επόμενη ενότητα.

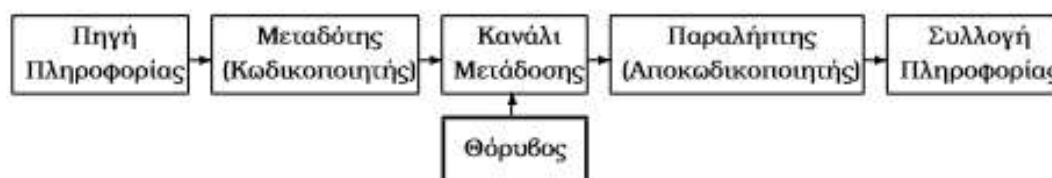
ΚΕΦΑΛΑΙΟ 4^ο ΘΕΩΡΙΑ ΚΩΔΙΚΩΝ

ΣΥΝΕΛΙΚΤΙΚΟΙ ΚΩΔΙΚΕΣ

Η Θεωρία Κωδίκων είναι η μελέτη των μεθόδων για αποτελεσματική και ακριβή μεταβίβαση πληροφοριών από το ένα μέρος στο άλλο. Η θεωρία έχει αναπτυχθεί για ποικίλες εφαρμογές, όπως ελαχιστοποίηση του θορύβου σε εγγραφές συμπαγών δίσκων, μετάδοση οικονομικών πληροφοριών μέσω τηλεφωνικών γραμμών, μετάδοση δεδομένων από τον έναν υπολογιστή στον άλλο ή από τη μνήμη στην κεντρική μονάδα επεξεργασίας και μετάδοση πληροφοριών από απομακρυσμένη πηγή, όπως μετεωρολογικούς ή επικοινωνιακούς δορυφόρους ή από το διαστημόπλοιο Voyager το οποίο στέλνει φωτογραφίες από τους πλανήτες Δία και Κρόνο στη Γη.

Το φυσικό μέσο, διαμέσου του οποίου η πληροφορία μεταδίδεται, λέγεται *κανάλι*. Οι τηλεφωνικές γραμμές και η ατμόσφαιρα είναι παραδείγματα καναλιών. Ανεπιθύμητες ενοχλήσεις, που ονομάζονται *Θόρυβος*, μπορεί να προκαλέσουν διαφοροποίηση μεταξύ της μεταδιδόμενης και της ληφθείσας πληροφορίας. Ο θόρυβος μπορεί να προκληθεί από ηλιακές κηλίδες, αστραπές, τσακίσματα της μαγνητικής ταινίας, βροχές μετεωριτών, φορτωμένες τηλεφωνικές γραμμές, τυχαία ραδιοπαρεμβολή, φτωχή δακτυλογράφηση, ασαφής ακρόαση, ασαφή ομιλία και πολλές άλλες αιτίες.

Η θεωρία Κωδίκων, ασχολείται με το πρόβλημα της ανίχνευσης και της διόρθωσης λαθών από μετάδοση, τα οποία προκαλούνται από τον θόρυβο στο κανάλι. Το διάγραμμα που ακολουθεί, μας δίνει μια γενική ιδέα ενός γενικού συστήματος μετάδοσης πληροφοριών.



Το πιο σημαντικό μέρος του διαγράμματος, γι'αυτά που μας ενδιαφέρουν εδώ, είναι ο θόρυβος (Noise), ελλείψει του οποίου δε θα υπήρχε ανάγκη για τη θεωρία Κωδίκων.

Στην πράξη, ο έλεγχος που έχουμε πάνω στο θόρυβο, είναι η επιλογή ενός καλού καναλιού προς χρήση για μετάδοση και η χρήση κάποιων φίλτρων θορύβου για την αντιμετώπιση συγκεκριμένων τύπων παρεμβολής που μπορεί ν' ανακύψουν. Αυτά είναι μηχανικά προβλήματα. Έτσι και καταλήξουμε στο καλύτερο μηχανικό σύστημα προκειμένου να λύσουμε αυτά τα προβλήματα, μπορούμε να επικεντρώσουμε την προσοχή μας στην κατασκευή του κωδικοποιητή και του αποκωδικοποιητή. Η πρόθεσή μας είναι να τα κατασκευάσουμε με τέτοιο τρόπο, ώστε να πετύχουμε:

- γρήγορη κωδικοποίηση πληροφοριών,
- εύκολη μετάδοση των κωδικοποιημένων μηνυμάτων,
- γρήγορη αποκωδικοποίηση των ληφθέντων μηνυμάτων,
- διόρθωση λαθών που εισάγονται στο κανάλι και
- μέγιστη μετάδοση πληροφοριών στη μονάδα του χρόνου.

Ο πρωταρχικός στόχος είναι ο τέταρτος από τους παραπάνω. Το πρόβλημα είναι ότι δεν είναι γενικά συμβατός με τον πέμπτο και ειδικά, ίσως δεν είναι συμβατός ούτε με τους άλλους τρεις. Έτσι, κάθε λύση επιτυγχάνεται ισορροπώντας και τους πέντε στόχους.

Στις καθημερινές προσωπικές μας επικοινωνίες, χρησιμοποιούμε βασικά λέξεις, προφορικά ή γραπτά, φτιαγμένες από περιορισμένο αλφάβητο. Έχουμε πληροφορίες για να επικοινωνούμε· τις κωδικοποιούμε σε μηνύματα που στη συνέχεια τα εκφράζουμε γραπτά ή προφορικά. Στη συνέχεια αυτά αποστέλλονται μέσω ενός καναλιού, που συνήθως είναι το διάστημα μεταξύ του στόματος και του αυτιού ή από το στυλό στο χαρτί και στη συνέχεια στο μάτι. Ο θόρυβος μπορεί να προκληθεί από ασαφή ομιλία, κακό άκουσμα, λανθασμένη γραμματική, δυνατό στερεοφωνικό συγκρότημα, φορτισμένη συζήτηση, ανορθογραφία, παρερμηνεία ή λανθασμένη δακτυλογράφηση. Ο αποκωδικοποιητής είναι το δικό μας διάβασμα (ή άκουσμα) και η κατανόηση του ληφθέντος μηνύματος.

Έχουμε εσωτερικές διαδικασίες διόρθωσης λαθών που ούτε καν είχαμε φανταστεί. Υποθέτοντας ότι λαμβάνουμε το μήνυμα "Apt natural. I have a gub.", το οποίο είναι μία ατάκα από το "Take the money and run" του Woody Allen. Αφού η Αγγλική γλώσσα δεν χρησιμοποιεί όλες τις πιθανές λέξεις οποιουδήποτε δοσμένου

μήκους, πιθανότατα θα αναγνωρίσουμε ότι το "gub" δεν είναι λέξη της Αγγλικής. Μπορούμε με ασφάλεια να υποθέσουμε ότι η μεταδοθείσα λέξη είναι κοντά στο "gub" κατά μία έννοια. Αρα είναι πιο πιθανό να ήταν "gut" ή "gun" ή "tub" παρά "firetruck" ή "rat". Είναι όμως μόνο τα συμφραζόμενα του μηνύματος που μας επιτρέπουν να διαλέξουμε το "gun" σαν την πιο πιθανή λέξη. Η λέξη "Art" είναι τέλεια ορθή στα Αγγλικά, αλλά πάλι από τα συμφραζόμενα οδηγούμαστε τελικά να τη διορθώσουμε σε "act". Αν επίσης τυχαίνει να είμαστε γνώστες της Αγγλικής, θα διορθώσουμε επιπλέον το "natural" σε "naturally", αν και αυτό το λάθος οφείλεται στην πηγή και όχι στο κανάλι.

Από αυτούς τους τύπους λαθών, μπορούμε να ασχοληθούμε μόνο με τον πρώτο: δηλαδή με το να επιλέξουμε την πιο πιθανή μεταδοθείσα λέξη. Η σίγουρη μέθοδος για την καταπολέμηση των λαθών, είναι μέσω της χρήσης επιπλέον ψηφίων. Πολλές επιχειρήσεις σήμερα, συνήθως προσθέτουν ψηφία ελέγχου σε νούμερα αναγνώρισης. Αυτά είναι επιπλέον ψηφία που χρησιμεύουν στον έλεγχο της ορθότητας δεδομένων ή αριθμών που προκύπτουν από πράξεις. Αυτή είναι μάλλον και η πιο συνήθως αναγνωρίσιμη μέθοδος κωδικοποίησης στην καθημερινή ζωή. Θα ασχοληθούμε με πιο επιτηδευμένες παρόμοιες ιδέες.

4.1. Κωδικοποίηση ελέγχου-διόρθωσης λαθών

Η κωδικοποίηση διαύλου αποτελεί ένα αδιαμφισβήτητο εργαλείο στα σύγχρονα συστήματα επικοινωνιών που διέπονται από αυστηρούς περιορισμούς της ισχύος και του εύρους ζώνης. Όπως έγινε ξεκάθαρο από τον Shannon το 1948, μπορούν να επιτευχθούν μεγάλα κέρδη κωδικοποίησης, για δεδομένη φασματική απόδοση, με την κωδικοποίηση της ακολουθίας πληροφορίας σε μεγάλα κομμάτια (blocks). Το μέγιστο κέρδος κωδικοποίησης για δεδομένο ρυθμό και μεγάλα κομμάτια (blocks) φράσσεται από το όριο χωρητικότητας που δίνεται από τον Shannon.

Με βάση τη δουλειά του Shannon, πολλοί μηχανικοί επικοινωνιών στην προσπάθεια να επιτύχουν ρυθμό μετάδοσης κοντά στο όριο του Shannon με υλοποιήσιμη πολυπλοκότητα, ανέπτυξαν πολυάριθμα σχήματα κωδικοποίησης για ένα μεγάλο εύρος εφαρμογών. Τα τελευταία πενήντα χρόνια έχει επιτευχθεί μεγάλη πρόοδος στην εφαρμογή τεχνικών κωδικοποίησης ελέγχου-διόρθωσης λαθών στις

ψηφιακές επικοινωνίες και στα αποθηκευτικά συστήματα. Η πρόοδος που σημειώθηκε στο σχεδιασμό πρακτικών συστημάτων μείωσε το χάσμα που υπήρχε μεταξύ της πραγματικής απόδοσης των συστημάτων και της χωρητικότητας του διαύλου. Οι εφαρμογές αυτές εστιάζονται στους παρακάτω τομείς: διαστημικές και δορυφορικές επικοινωνίες, μετάδοση δεδομένων, αποθήκευση δεδομένων, μετάδοση ψηφιακού ήχου και εικόνας, κινητές επικοινωνίες και μεταφορά αρχείων.

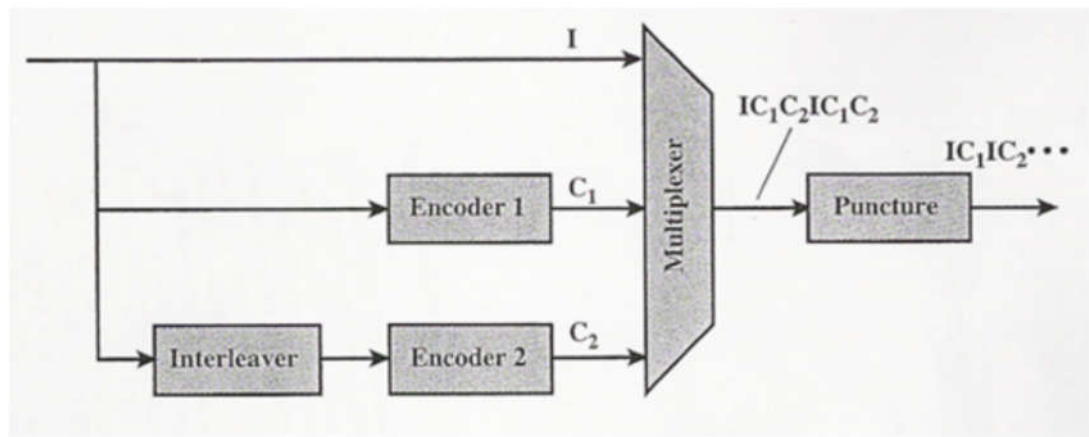
Συγκεκριμένα, οι βαθμίδες που προστίθενται στα τηλεπικοινωνιακά συστήματα είναι ο κωδικοποιητής και ο αποκωδικοποιητής διαύλου. Ο κωδικοποιητής "προσθέτει" συστηματικά ψηφία (bits) στα ψηφία που έχουν πληροφοριακό περιεχόμενο. Τα πρόσθετα αυτά ψηφία, αν και δεν περιέχουν πληροφορία, δίνουν τη δυνατότητα στον αποκωδικοποιητή να ανιχνεύει ή και να διορθώνει σφάλματα μετάδοσης που έχουν προκύψει κατά τη μετάδοση του σήματος. Με το τρόπο αυτό μειώνεται η πιθανότητα λάθους κατά τη μετάδοση. Επίσης η κωδικοποίηση μπορεί να χρησιμοποιείται για καλύτερη χρησιμοποίηση της χωρητικότητας του διαύλου.

Πριν από δέκα χρόνια, ο Berrou, ο Glavieux και ο Thitimajshima εισήγαγαν μια νέα προσέγγιση στην κωδικοποίηση διόρθωσης λαθών που έφερε επανάσταση στη θεωρία και στις τεχνικές κωδικοποίησης. Ανακάλυψαν ένα ψηφιακό σχήμα κωδικοποίησης που μπορούσε να παρέχει εικονικά αλάνθαστες επικοινωνίες σε μεγαλύτερους ρυθμούς δεδομένων και αποδόσεις μεταδιδόμενης ισχύος από ότι θεωρούσαν δυνατόν οι ειδικοί. Το σχήμα που ανακάλυψαν, το οποίο ονομάστηκε κώδικες turbo (turbo codes) σε αναλογία με τη μηχανή turbo και την αποτελεσματική χρήση της ανατροφοδότησης, οδήγησε σε τεχνικές κωδικοποίησης οι οποίες προσεγγίζουν τα απόλυτα όρια απόδοσης.

Η καινούρια τάξη κωδικών αποτελεί σημαντική επιλογή για τα ασύρματα συστήματα τρίτης γενιάς και συγκεκριμένα χρησιμοποιείται από τα συστήματα UMTS (Universal Mobile Telecommunications Systems) και τα κυψελωτά συστήματα τρίτης γενιάς cdma2000. Οι διαφορετικοί τύποι των κωδικοποιητών και αποκωδικοποιητών turbo στηρίζονται στη συνελκτική κωδικοποίηση.

Στο σχήμα απεικονίζεται ένας κωδικοποιητής turbo, ο οποίος αποτελείται από δύο όμοιους κωδικοποιητές. Ο πρώτος κωδικοποιητής (encoder 1) δέχεται μια ακολουθία ψηφίων εισόδου και παράγει για κάθε εισερχόμενο ψηφίο ένα ψηφίο ελέγχου C1 (check bit). Η είσοδος στο δεύτερο κωδικοποιητή (encoder 2) είναι μια

αναδιαταγμένη εκδοχή της ακολουθίας ψηφίων εισόδου, παράγοντας έτσι μία ακολουθία ψηφίων ελέγχου C_2 . Τα αρχικά ψηφία εισόδου σε συνδυασμό με τα δύο ψηφία ελέγχου πολυπλέκονται στη συνέχεια και παράγουν την ακολουθία $I C_1 C_2 I C_1 C_2, \dots$, η οποία αποτελείται από το πρώτο ψηφίο εισόδου που ακολουθείται από το πρώτο ψηφίο του πρώτου κωδικοποιητή, που ακολουθείται από το πρώτο ψηφίο του δεύτερου κωδικοποιητή, κ.ο.κ. Η τελική ακολουθία έχει ρυθμό κωδικοποίησης $1/3$. Λαμβάνοντας μόνο τα μισά από τα ψηφία ελέγχου, εναλλάσσοντας τις εξόδους των δύο κωδικοποιητών, μπορεί να επιτευχθεί ρυθμός κωδικοποίησης $1/2$. Αυτή η διαδικασία ονομάζεται διάτρηση (puncturing). Οι ρυθμοί κωδικοποίησης $1/3$ και $1/2$ συναντώνται στα συστήματα τρίτης γενιάς.



Σχήμα 4.1: Κωδικοποίηση turbo

Πρέπει να σημειωθεί ότι κάθε κωδικοποιητής παράγει ένα μόνο ψηφίο ελέγχου για κάθε ψηφίο εισόδου καθώς επίσης και ότι το ψηφίο εισόδου διατηρείται. Για την κωδικοποίηση turbo χρησιμοποιείται μια παραλλαγή της συνελκτικής κωδικοποίησης, γνωστή ως επαναληπτική συστηματική συνελκτική κωδικοποίηση RSC (Recursive Systematic Convolutional).

4.2. Συνελικτικοί Κώδικες

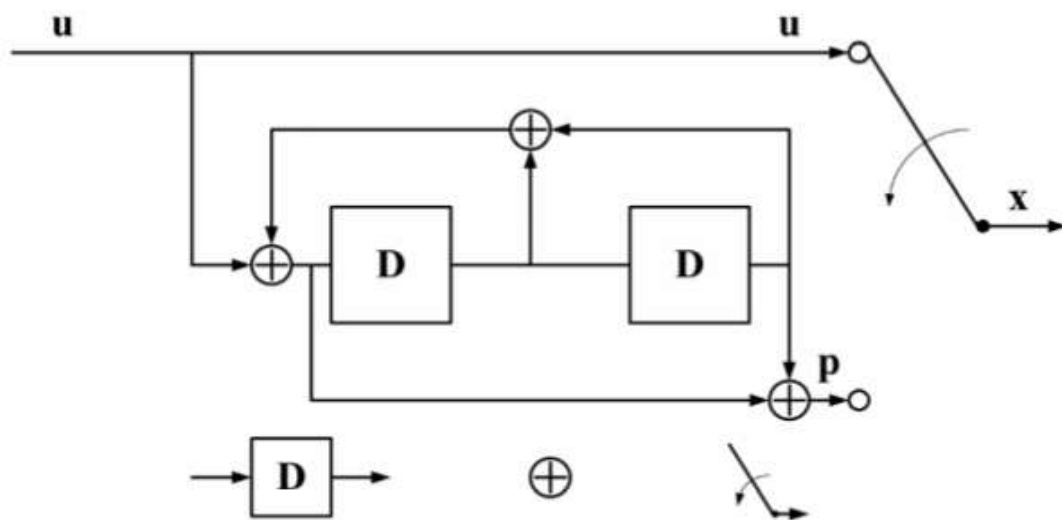
Οι συνελικτικοί κώδικες ανακαλύφθηκαν από τον Elias το 1954 ενώ η αλγεβρική θεωρία τους αναλύθηκε από τον Forney . παρόμοια με τους κώδικες δομής ορίζονται και αυτοί ως η απεικόνιση από έναν διανυσματικό χώρο σε έναν άλλον. Η κύρια διαφορά τους εδράζεται στο γεγονός ότι πραγματοποιούν την διαδικασία της κωδικοποίησης σε μία μεγάλου μήκους και τυχαία ακολουθία δεδομένων και όχι σε ανεξάρτητες δομές δεδομένων. Στους κώδικες αυτούς η έξοδος σε κάθε χρονική στιγμή εξαρτάται όχι μόνο από την τρέχουσα είσοδο αλλά και από κάποιον αριθμό από παρελθούσες εισόδους. Οι συνελικτικοί κώδικες ανήκουν στην κατηγορία των κωδίκων trellis (trellis codes). Οι κώδικες αυτοί κωδικοποιούν τα δεδομένα αδιαλείπτως.

Ένας συνελικτικός κώδικας είναι κώδικας διόρθωση λάθους που επεξεργάζεται πληροφορίες σειριακά ή διαδοχικά, σε σύντομα μήκους μπλοκ. Ένας συνελικτικός κωδικοποιητής έχει μνήμη, με την έννοια ότι τα σύμβολα εξόδου δεν εξαρτώνται μόνο από τα σύμβολα εισόδου, αλλά και από προηγούμενες καταστάσεις εισροών ή εκροών. Με άλλα λόγια, ο κωδικοποιητής είναι ένα συνεχές κύκλωμα ή μια μηχανή πεπερασμένου αριθμού καταστάσεων.

Στα προγράμματα ηλεκτρονικών υπολογιστών που εφαρμόζουν του αλγορίθμου Viterbi και άλλες trellis διαδικασίες αποκωδικοποίησης, βρέθηκε πίνακας μετάβασης, υποδεικνύοντας τη σχέση μεταξύ της εισόδου, της προηγούμενη και της τρέχουσας κατάστασης, καθώς και της τρέχουσα παραγόμενης εξόδου. Στη θεωρία, οι διαδοχικές ακολουθίες που παράγουν έναν συνελικτικό κώδικα έχουν άπειρη διάρκεια ενώ στην πράξη, η τρέχουσα κατάσταση ενισχύεται περιοδικά σε μια γνωστή κατάσταση και οι ακολουθίες του κώδικα παράγονται σε μορφή block ακολουθιών.

Γενικά ένας συνελικτικός κωδικοποιητής με ρυθμό k/n αποτελείται από k καταχωρητές μετάβασης, έναν για κάθε bit εισόδου πληροφορίας και n κωδικοποιημένα bit εξόδου ως γραμμικοί συνδυασμοί των περιεχομένων των καταχωρητών και των bit της πληροφορίας εισόδου.

Ένας κωδικοποιητής για έναν συνελικτικό κώδικα φαίνεται στο σχήμα 2-6 και αποτελεί μια μηχανή πεπερασμένων καταστάσεων δηλαδή ένα ακολουθιακό λογικό κύκλωμα μνήμης m .



Ο γεννήτορας πίνακας Γενός συνελκτικού κώδικα έχει την ακόλουθη γενική μορφή

$$G(D) = \begin{pmatrix} G_{11}(D) & G_{12}(D) & \dots & G_{1n}(D) \\ \vdots & \vdots & \ddots & \vdots \\ G_{k1}(D) & G_{k2}(D) & \dots & G_{kn}(D) \end{pmatrix}$$

όπου το κάθε στοιχείο $G_{ij}(D)$ αναπαριστά μια συνάρτηση μεταφοράς (Kronecker-delta απόκρισης) συστήματος διακριτού-χρόνου τάξης m :

Στον κωδικοποιητή του σχήματος παρατηρούμε 2 πολυώνυμα που χαρακτηρίζουν τα γεωμετρικά χαρακτηριστικά και την τοπολογία του $G11=1+D2=101$ και $G12(D)=1+D+D2=111$. Ο γεννήτορας πίνακας $G(1,101/111)$ χαρακτηρίζει τον

$$G_{ij}(D) = \frac{a_m D^m + a_{m-1} D^{m-1} + \dots + a_0}{b_m D^m + b_{m-1} D^{m-1} + \dots + b_0}$$

κώδικα αυτόν με τον πρώτο όρο να αντιστοιχεί στην έξοδο u και τον δεύτερο όρο της παρενθέσεως να αντιστοιχεί στην έξοδο p . Τα πολυώνυμα αυτά καθορίζουν τις συνδέσεις μεταξύ των καταχωρητών ολίσθησης και των πυλών XOR. Για τον συγκεκριμένο αποκωδικοποιητή το πολυώνυμο $G12$ οριοθετεί την ανάδρασή του. Ο ρυθμός του συνελκτικού κώδικα ορίζεται από τον λόγο των bit εισόδου προς τον αριθμό των εξερχόμενων κωδικών bit. $R=k/n$ (2-27) όπου k και n είναι οι διαστάσεις του $G(D)$.

Στο παράδειγμα του σχήματος, ο ρυθμός του κώδικα είναι $R=1/2$. Ένα άλλο χαρακτηριστικό των συνελκτικών κωδικών είναι το μήκος εξαναγκασμού (constraint length) k , με $k=m+1$ και φανερώνει μετά από πόσες διακριτές χρονικές στιγμές ο κωδικοποιητής εξάγει το τρέχον bit εισόδου. Τυπικά, ο ρυθμός R του κώδικα διατηρείται σταθερός, δεδομένου ότι η τάξη m της μνήμης μπορεί να αυξάνει για να αντισταθμίσει τον θόρυβο του διαύλου. Αυτό σημαίνει ότι η πολυπλοκότητα του κώδικα αυξάνει για να χρησιμοποιηθεί περισσότερη πλεονάζουσα πληροφορία.

Γενικά, η διαδικασία εύρεσης καταλλήλου κώδικα περιλαμβάνει τόσο την διατήρηση του ρυθμού του κώδικα κοντά στην μονάδα και όσο και την ανάπτυξη ισχυρών δυνατοτήτων διόρθωσης λαθών. Συγκεκριμένα, κώδικες διόρθωσης λαθών ρυθμού μεγαλύτερου του 0.9 χρησιμοποιούνται κυρίως για εφαρμογές μαγνητικής αποθήκευσης, όπου η πυκνότητα της πληροφορίας στο μαγνητικό μέσο περιορίζεται κυρίως από το χρησιμοποιούμενο υλικό και την ταχύτητα μετάδοσης.

Για εφαρμογές επικοινωνίας δεδομένων, ο περιορισμός του υψηλού ρυθμού κωδικοποίησης δεν λαμβάνεται τόσο υπ' όψιν όσο η καλή διορθωτική ικανότητα του κώδικα. Για διαύλους υψηλού βαθμού θορύβου, χρησιμοποιούνται κώδικες με ρυθμούς $\frac{1}{2}$ και μικρότερους.

Οι συνελκτικοί κώδικες αποτελούν μια πολύ σημαντική κατηγορία κωδίκων, οι οποίοι επιτυγχάνουν αρκετά μεγάλα κέρδη στην κωδικοποίηση και η χρήση τους είναι ευρεία στα τηλεπικοινωνιακά συστήματα. Η ονομασία τους οφείλεται στον τρόπο με τον τρόπο που υλοποιείται η κωδικοποίηση της πληροφορίας, καθώς σχηματίζονται με συνέλιξη των ψηφίων της πληροφορίας με την κρουστική απόκρουση ενός καταχωρητή ολίσθησης.

Με την χρήση των συνελκτικών κωδίκων, ένα μπλοκ των k δυαδικών ψηφίων αντιστοιχίζεται σε ένα block των n δυαδικών ψηφίων. Κύριο χαρακτηριστικό γνώρισμα των συνελκτικών κωδίκων είναι ότι το νέο μπλοκ των n δυαδικών ψηφίων δεν εξαρτάται μόνο από τα τρέχοντα k δυαδικά ψηφία, αλλά και από προηγούμενα δυαδικά ψηφία. Αυτό σημαίνει ότι ο κωδικοποιητής πρέπει να έχει μνήμη και ως εκ τούτου μετατρέπεται σε μια Μηχανή Πεπερασμένων Καταστάσεων.

4.2.1. Συναλυσώμενη Κωδικοποίηση

Εξετάζοντας τους συνελκτικούς κώδικες όπως είδαμε στην προηγούμενη ενότητα, εάν η ικανότητα διόρθωσης λαθών του κώδικα ξεπεραστεί από το βάρος της ακολουθίας λάθους που θα προστεθεί από το κανάλι, τότε είναι πιθανόν η αποκωδικοποιημένη πληροφορία να περιέχει τμήματα μεγάλου μήκους λαθών. Για το λόγο αυτό, οι συνελκτικοί κώδικες χρησιμοποιούνται ευρέως ως εσωτερικοί κώδικες της σειράς συνενωμένων συστήματα κωδικοποίησης, όπου ως εξωτερικοί χρησιμοποιούνται συνήθως οι Reed-Solomon (RS) που διαθέτουν υψηλή ικανότητα διόρθωσης μεγάλου μήκους λαθών. Με αυτόν τον τρόπο, με τη βοήθεια του

εξωτερικού κώδικα, οποιαδήποτε ακολουθία λαθών κατά αποκωδικοποίηση συνελκτικών κωδίκων μπορεί να εξαλειφθεί. Αυτός ο συνδυασμός συνελκτικού (εσωτερικό) και κώδικες RS (εξωτερικό) των κωδίκων έχει βρεθεί σε πολλά πρακτικές εφαρμογές, συμπεριλαμβανομένης της ψηφιακής δορυφορικής μετάδοσης καθώς και διαστημικές επικοινωνίες

4.2.2. Concatenation Μέθοδοι

Οι Concatenation κώδικες είναι μια πολύ χρήσιμη τεχνική που οδηγεί στην κατασκευή πολύ αποτελεσματικών κωδίκων με τη χρήση δύο ή περισσότερων συνιστωσών κωδίκων σχετικά μικρού μεγέθους και πολυπλοκότητας. Έτσι, ένας ισχυρός κώδικας με υψηλές BER (Bit Error Performance) επιδόσεις αλλά ανέφικτης πολυπλοκότητας, μπορεί να κατασκευαστεί σε μια ισοδύναμη μορφή από την συνένωση συνδυασμών δύο ή περισσότερων κωδίκων ώστε να παρέχουν τις ίδιες επιδόσεις με μικρότερο κόστος από την άποψη της πολυπλοκότητας. Η μειωμένη πολυπλοκότητα είναι ιδιαίτερα σημαντική για την αποκωδικοποίηση των κωδίκων αυτών, οι οποίες μπορούν να επωφεληθούν από τη συνδυασμένη δομή ενός Concatenation κώδικα.

Η αποκωδικοποίηση γίνεται με το συνδυασμό δύο ή περισσότερων σχετικά χαμηλής πολυπλοκότητας αποκωδικοποιητές, έτσι που αποσυντίθενται αποτελεσματικά το πρόβλημα της αποκωδικοποίησης ενός μεγάλου κώδικα. Εάν αυτοί οι αποκωδικοποιητές μοιραστούν κατάλληλα την πληροφορία προς αποκωδικοποίηση με τη χρήση επαναληπτικής τεχνικής, για παράδειγμα, τότε δεν θα υπάρχει απώλεια της απόδοσης.

Υπάρχουν ουσιαστικά δύο τρόποι υλοποίησης concatenating κωδίκων: παραδοσιακά, με τη χρήση της λεγόμενης σειριακής (serial) concatenation, και πιο πρόσφατα, συνενώνοντας παράλληλα δομές της πρώτης turbo κατηγορίας συστημάτων κωδικοποίησης. Η επαναληπτική διαδικασία αποκωδικοποίησης επιτρέπεται στην χρήση και των δύο αυτών concatenation τεχνικών.

4.2.3. Block Interleaving

Ένας interleaver παίρνει μια ακολουθία από σύμβολα και τα συνδυάζει μεταξύ τους. Στο δέκτη, το ακολουθία συνδυάζεται έτσι ώστε να επιστρέψει στην αρχική της σειρά από το σύστημα του deinterleaver. Οι Interleavers είναι αποτελεσματικοί στην αντιμετώπιση μεγάλων εξαπλώσεων λαθών, διότι καθώς μεταθέτει τα σύμβολα στο

δέκτη, σφάλματα που εμφανίζονται σε κοντινή απόσταση μπορούν να κατατμηθούν και να εξαπλωθούν σε μεγάλο εύρος, δημιουργώντας έτσι ένα αποτελεσματικό τυχαίο κανάλι .

Ένας κοινός τρόπος συνδυασμού και μετατόπισης συμβόλων είναι οι μπλοκ interleavers. Αποτελούνται από πίνακα $*NM$ διαστάσεων που μπορεί να διαβαστεί και να γραφτεί με διάφορους συνδυασμούς ακολουθιών. Συνήθως, η εισερχόμενη ακολουθία συμβόλων είναι γραμμένη στον interleaver ακολουθώντας τις σειρές και διαβάζεται ακολουθώντας τις στήλες. Στο παρακάτω σχήμα απεικονίζεται ένας 3×4 interleaver. Η ακολουθία των εισροών $0\ 1\ 11\ , \dots, x\ xx$ δίνεται σε σειρές του πίνακα, όπως φαίνεται, και διαβάζεται όπως η ακολουθία:

$\chi_0\ \chi_4\ \chi_8\ \chi_{12}\ \chi_1\ \chi_5\ \chi_9\ \chi_{13}\ \chi_2\ \chi_6\ \chi_{10}\ \chi_{14}\ \chi_3\ \chi_7\ \chi_{11}$

4.3. Βασικά Στοιχεία Θεωρίας

Ένας συνελκτικός κωδικοποιητής υλοποιείται με έναν απλό καταχωρητή ολίστησης και modulo-2 αθροιστές. Ένας συνελκτικός κώδικας χαρακτηρίζεται από την τριάδα (n,k,L) .

Ο ακέραιος αριθμός k αποτελεί τον αριθμό των ψηφίων που εισέρχονται στον κωδικοποιητή, n είναι ο αριθμός των ψηφίων της εξόδου του κωδικοποιητή σε κάθε χρονικό βήμα και L ονομάζεται το μήκος περιορισμού και δηλώνει τον αριθμό των ομάδων των k ψηφίων που βρίσκονται κάθε στιγμή στον καταχωρητή.

Ρυθμός ενός συνελκτικού κώδικα ορίζεται ως ο λόγος του αριθμού των ψηφίων στην είσοδο του κωδικοποιητή προς τον αριθμό των ψηφίων στην έξοδο του κωδικοποιητή. Με άλλα λόγια:

$$R=k/n$$

Επίσης ως ελεύθερη απόσταση ενός συνελκτικού κώδικα ορίζεται η ελάχιστη απόσταση μεταξύ δύο κωδίκων ακολουθιών.

Επειδή, όπως αναφέρθηκε και προηγουμένως, ένας συνελκτικός κωδικοποιητής είναι μια διάταξη με μνήμη, είναι φανερό ότι οι n -άδες εξόδου του κωδικοποιητή δεν εξαρτώνται μόνο από τα k πιο πρόσφατα δυαδικά ψηφία στην είσοδο του κωδικοποιητή, αλλά επίσης και από τα $(L-1)k$ δυαδικά ψηφία που περιείχαν οι πρώτες $(L-1)k$ βαθμίδες του καταχωρητή πριν φθάσουν τα νέα k bits. Επομένως,

προκύπτει ότι ο καταχωρητής ολίσθησης είναι μια μηχανή πεπερασμένων καταστάσεων με $2(L-1)k$ καταστάσεις.

Σε κάθε χρονική στιγμή, εισέρχονται στον καταχωρητή ολίσθησης k δυαδικά ψηφία πληροφορίας και εξέρχονται αυτού τα k δυαδικά ψηφία του τελευταίου σταδίου του καταχωρητή ολίσθησης. Από την στιγμή που τα k δυαδικά ψηφία εισέρχονται στον καταχωρητή ολίσθησης, υπολογίζονται n γραμμικοί συνδυασμοί των περιεχομένων του καταχωρητή ολίσθησης και αυτοί χρησιμοποιούνται για την παραγωγή της κωδικοποιημένης εξόδου.

4.4. Συνάρτηση Μεταφοράς

Για κάθε συνελκτικό κώδικα, η πληροφορία που μας δίνει η συνάρτηση μεταφοράς αφορά τις διάφορες διαδρομές μέσα από το trellis διάγραμμα, που ξεκινούν από την κατάσταση με όλα μηδενικά και επανέρχονται στην κατάσταση αυτή για πρώτη φορά. Όπως αναφέρθηκε και προηγουμένως, κάθε κωδική λέξη ενός συνελκτικού κώδικα αντιστοιχεί σε μία διαδρομή από το trellis που ξεκινά από την κατάσταση με όλα μηδενικά και επανέρχεται στην ίδια ακριβώς κατάσταση.

Η διαδικασία που ακολουθείται για τον υπολογισμό της συνάρτησης μεταφοράς ενός συνελκτικού κώδικα είναι η εξής: Διαιρείται η όλα μηδενικά κατάσταση σε 2 «υποκαταστάσεις», μια που δηλώνει την κατάσταση εκκίνησης και μία που δηλώνει την κατάσταση πρώτης επιστροφής στην όλα μηδενικά κατάσταση. Δύο καταστάσεις συνδέονται μέσω ενός κλάδου, ο οποίος ορίζεται μία συνάρτηση της μορφής: $D\alpha N\beta J$. Το α ισούται με τον αριθμό των μονάδων που περιέχει η έξοδος για τον κλάδο αυτό και το β ισούται με τον αριθμό των μονάδων που περιέχει η είσοδος για τον κλάδο αυτό. Έτσι, η συνάρτηση μεταφοράς του συνελκτικού κώδικα ορίζεται ως η συνάρτηση μεταφοράς του γραφήματος ροής μεταξύ της όλα μηδενικά κατάστασης εκκίνησης και της τελικής όλα-μηδενικά κατάστασης. Επίσης, θα είναι συνάρτηση των τριών παραμέτρων D, N, J και θα συμβολίζεται με $T(D, N, J)$. Κάθε στοιχείο της συνάρτησης μεταφοράς αντιστοιχεί σε μία διαδρομή μέσα από το trellis διάγραμμα, που ξεκινάει από την όλα-μηδενικά κατάσταση και καταλήγει στην όλα μηδενικά κατάσταση. Ο εκθέτης του J ισούται με το πλήθος των κλάδων που καλύπτει η διαδρομή, ο εκθέτης του D ισούται με το πλήθος των μονάδων που περιέχονται στην κωδική λέξη που αντιστοιχεί στην διαδρομή και ο εκθέτης του N ισούται με το πλήθος των μονάδων της εισόδου. Τέλος, πρέπει να αναφερθεί ότι για

τον υπολογισμό της συνάρτησης μεταφοράς, αγνοούμε όλους του βρόχους ανακύκλωσης στην με-όλα μηδενικά κατάσταση.

Θα πρέπει να αναφέρουμε ότι υπάρχουν τρεις τρόποι για την αναπαράσταση ενός συνελκτικού κωδικοποιητή. Και αυτοί είναι:

1. Διάγραμμα Καταστάσεων
2. Δενδρικό Διάγραμμα
3. Διάγραμμα Trellis

4.4.1. Διάγραμμα Καταστάσεων (State Diagram)

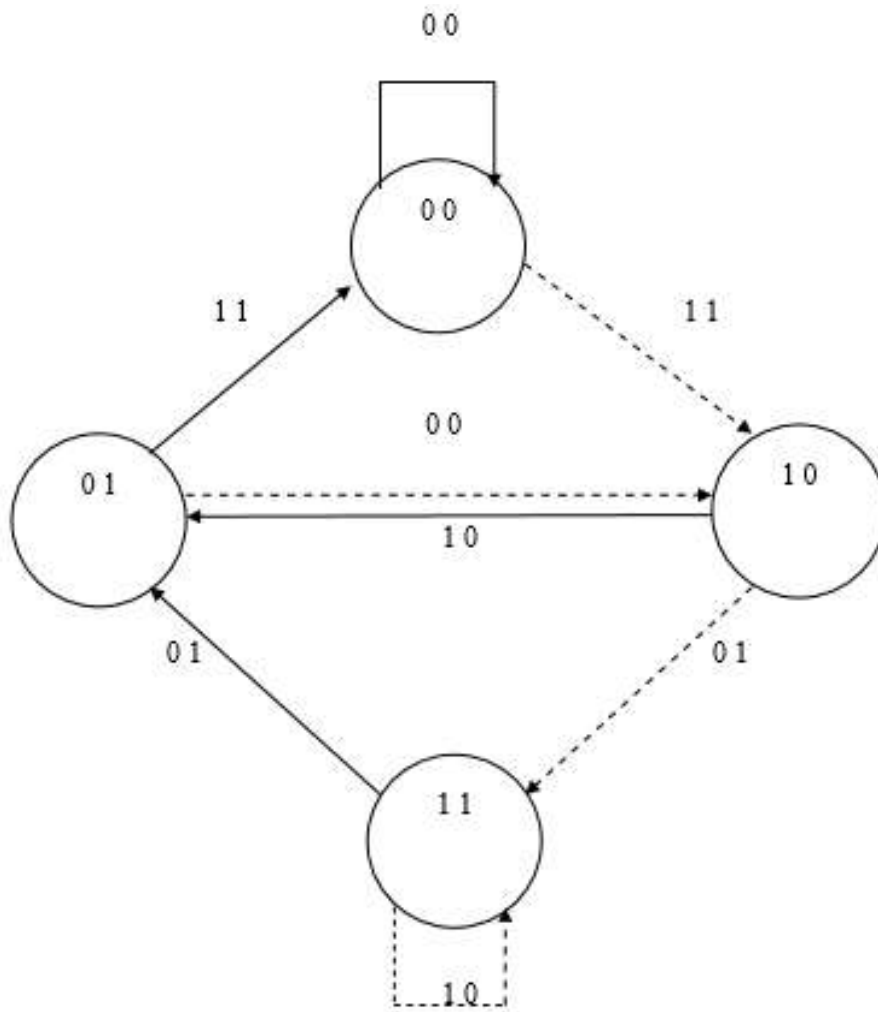
Ένας συνελκτικός κωδικοποιητής ισοδυναμεί με μια μηχανή πεπερασμένων καταστάσεων. Αυτό σημαίνει ότι ένας συνελκτικός κώδικας έχει πεπερασμένη μνήμη και άρα μπορούν να παρατηρηθούν μεταβάσεις σε έναν συγκεκριμένο αριθμό μοναδικών καταστάσεων. Ο συνελκτικός κώδικας επομένως μπορεί να αναπαρασταθεί με ένα διάγραμμα μετάπτωσης καταστάσεων.

Στο διάγραμμα αυτό, κάθε κατάσταση του συνελκτικού κώδικα μπορεί να αναπαρασταθεί με ένα τετράγωνο (ή έναν κύκλο) και οι μεταπτώσεις (μεταβάσεις) μεταξύ των καταστάσεων δηλώνονται με βέλη που συνδέουν τα τετράγωνα (κύκλους), προσδιορίζοντας με αυτόν τον τρόπο τις αφίξεις προς ή τις αποχωρήσεις από την κατάσταση. Επιπλέον, πρέπει να αναφερθεί ότι σε κάθε βέλος προσδιορίζονται τόσο η είσοδος του κωδικοποιητή που προκάλεσε την συγκεκριμένη μετάβαση όσο και η έξοδος του κωδικοποιητή). Συνοψίζοντας, ο κωδικοποιητής βρίσκεται σε μια συγκεκριμένη κατάσταση και δεχόμενος μια είσοδο μεταβαίνει σε μια κατάσταση(μπορεί να είναι και η ίδια) και παράγει την έξοδό του. Αυτό που επίσης πρέπει να επισημανθεί είναι ότι δεν μπορεί να πραγματοποιηθεί απευθείας από μία κατάσταση προς όλες τις καταστάσεις

Μία ακόμη σημαντική παρατήρηση είναι ότι το πλήθος των βελών που εξέρχονται από κάθε κατάσταση ισούται με το πλήθος των δυνατών εισόδων του κωδικοποιητή όταν βρίσκεται στην κατάσταση αυτή. Τέλος, το πλήθος των βελών που συγκλίνουν προς κάθε κατάσταση είναι ίσο με το πλήθος των καταστάσεων από τις οποίες είναι δυνατή μια μετάπτωση προς την κατάσταση αυτή.

Να επισημανθεί, ότι στο διάγραμμα καταστάσεων που ακολουθεί τα διακεκομμένα βέλη αναπαριστούν μεταβάσεις που προκλήθηκαν λόγω εισόδου 0, και τα βέλη με

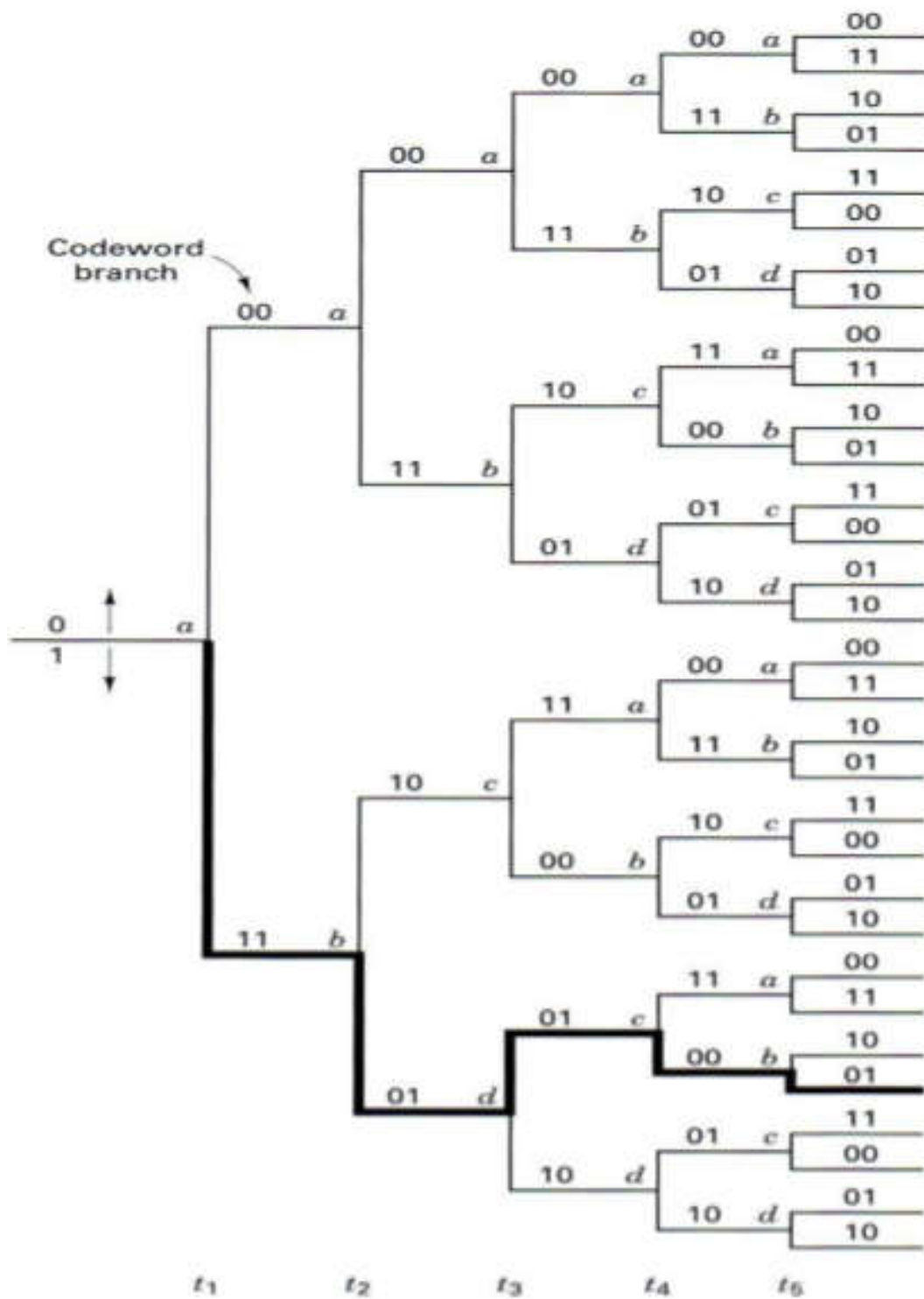
συνεχή γραμμή αναπαριστούν μεταβάσεις που προκλήθηκαν λόγω εισόδου 1. Έτσι, παρουσιάζεται το ακόλουθο διάγραμμα καταστάσεων για τον κωδικοποιητή του παραδείγματός μας:



4.4.2. Δεντρικό Διάγραμμα (Tree Diagram)

Ένα πρόβλημα που υπάρχει με την αναπαράσταση με χρήση του Διαγράμματος Καταστάσεων είναι ότι δεν μπορούσε να αναπαραστήσει το χρόνο. Το πρόβλημα αυτό λύθηκε με την χρήση του Δεντρικού Διαγράμματος, το οποίο προσθέτει την διάσταση του χρόνου στην αναπαράσταση.

Αν ξεκινήσουμε από την ρίζα του δέντρου, σε κάθε βήμα της κωδικοποίησης, προχωράμε προς τα δεξιά και πάνω εάν είναι μηδέν (0) η είσοδος του κωδικοποιητή ή δεξιά και κάτω εάν είναι ένα (1) η είσοδος του κωδικοποιητή. Επιπλέον, σε κάθε κόμβο του δέντρου αναγράφεται η τρέχουσα κατάσταση του κωδικοποιητή. Σε κάθε κλαδί του δέντρου αναγράφεται η λέξη εξόδου, η οποία είναι αποτέλεσμα της παραγωγής από τον κωδικοποιητή κατά την συγκεκριμένη μετάβαση



Μπορεί η αναπαράσταση με την χρήση Δεντρικού Διαγράμματος να έδωσε λύση στο πρόβλημα του χρόνου, όμως εισήγαγε ένα άλλο σημαντικό μειονέκτημα: επειδή είναι δυναμικό, το μέγεθός του μετά από λίγες χρονικές στιγμές αυξάνεται

υπερβολικά πολύ. Το γεγονός αυτό καθιστά την χρήση Δεντρικών Διαγραμμάτων απαγορευτική. Ας αναλογιστούμε το εξής: Για δυαδικούς συνελκτικούς κώδικες με $k=1$ σε κάθε χρονική στιγμή φεύγουν δύο διακλαδώσεις, κάτι το οποίο συνεπάγεται ότι στην i -στη χρονική στιγμή θα έχουν προκύψει 2^{i-1} κόμβοι.

4.4.3. Διάγραμμα trellis (Trellis Diagram)

Ένα trellis είναι ένας γράφος που χρησιμοποιείται συνήθως για την αναπαράσταση των συνελκτικών κωδίκων καθώς και κωδικών δομής. Πριν προχωρήσουμε στους ορισμούς του trellis διαγράμματος, θα δώσουμε την περιγραφή του διαγράμματος καταστάσεων ενός συνελκτικού κώδικα.

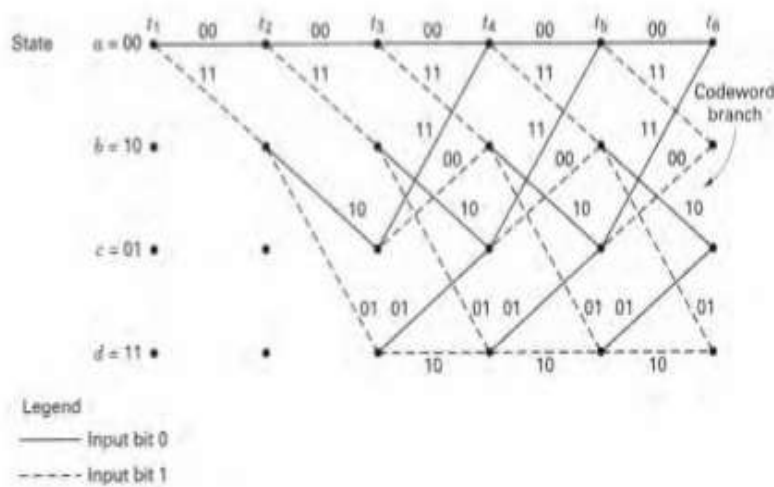
Ένας συνελκτικός κωδικοποιητής, όντας ένα γραμμικό κύκλωμα πεπερασμένων καταστάσεων, μπορεί να περιγραφεί από ένα διάγραμμα καταστάσεων. Η κατάσταση του κωδικοποιητή ορίζεται από το περιεχόμενο της μνήμης του. Αν η μνήμη του κωδικοποιητή είναι K , ο συνολικός αριθμός των καταστάσεων του είναι 2^K . Η τρέχουσα κατάσταση και η έξοδος του αποκωδικοποιητή προσδιορίζονται μοναδικά από την προηγούμενη κατάσταση και την τρέχουσα είσοδό του.

Ο κωδικοποιητής υφίσταται μια μετάβαση καταστάσεως όταν μια δομή μηνυμάτων ολισθαίνει μέσα στον κωδικοποιητή. Το διάγραμμα καταστάσεων αποτελεί εξίσου ένα γράφο που αποτελείται από κόμβους, οι οποίοι αναπαριστούν τις καταστάσεις του κωδικοποιητή, και κατευθυνόμενες γραμμές, οι οποίοι αναπαριστούν τις μεταβάσεις των καταστάσεων.

Η κάθε κατευθυνόμενη γραμμή επιγράφεται από το ζευγάρι είσοδος/έξοδος. Δοθείσας μιας τρέχουσας κατάστασης του κωδικοποιητή, η πληροφοριακή ακολουθία στην είσοδο καθορίζει τη διαδρομή που ακολουθείται μέσω του διαγράμματος καταστάσεων και της ακολουθίας εξόδου. Ας θεωρήσουμε τον συστηματικό συνελκτικό κώδικα του σχήματος. Ο κωδικοποιητής περιλαμβάνει τέσσερις καταστάσεις, τις S_0, S_1, S_2 και S_3 όπου οι δείκτες τους αντιστοιχούν στις ακέραιες αναπαραστάσεις των περιεχομένων των στοιχείων των καταχωρητών ολισθήσεως. Υπάρχουν δύο διαδρομές που αφήνουν την κάθε κατάσταση αναλόγως τις δύο πιθανές τιμές του bit εισόδου.

Μια τρίτη και πιο δημοφιλής μέθοδος για την περιγραφή των συνελκτικών κωδίκων είναι ο καθορισμός του διαγράμματος δικτύωματος (διάγραμμα trellis) που

αντιστοιχεί στον κώδικα. Το διάγραμμα trellis αποτελεί έναν τρόπο παρουσίασης των μεταπτώσεων μεταξύ των διαφόρων καταστάσεων, καθώς εξελίσσεται ο χρόνος. Ο τρόπος κατασκευής του διαγράμματος trellis είναι ο εξής: Τοποθετούμε σε έναν κατακόρυφο άξονα όλες τις δυνατές καταστάσεις και επαναλαμβάνουμε τον άξονα αυτό για κάθε χρονική περίοδο μετάπτωσης (μετάβασης). Κάθε μετάπτωση (μετάβαση) από μια κατάσταση σε μία άλλη αναπαρίσταται με ένα ευθύγραμμο τμήμα που συνδέει τα παραστατικά σημεία των δύο καταστάσεων σε δύο διαδοχικούς κατακόρυφους άξονες. Θα μπορούσε κανείς να αναφέρει ότι το διάγραμμα trellis αποτελεί επανάληψη του διαγράμματος καταστάσεων στον άξονα του χρόνου. Στο παρακάτω σχήμα που ακολουθεί παρουσιάζεται το διάγραμμα trellis που αντιστοιχεί στο παράδειγμά μας. Με βέλη διακεκομμένα συμβολίζονται οι μεταβάσεις προκαλούμενες λόγω εισόδου ένα (1) και με συνεχόμενα βέλη οι μεταβάσεις που προκαλούνται λόγω εισόδου μηδέν (0). [1,2,3,5,10]



4.5. Παράλληλα Συνδεδεμένα Συνελκτικοί Κώδικες (Parallel Concatenated Convolutional Codes-PPCCC)-Turbo Κώδικες

Ένα κλασικό σχήμα turbo κωδικοποιητή αποτελείται από δύο επιμέρους συνελκτικούς κωδικοποιητές ρυθμού $\frac{1}{2}$, οι οποίοι διαχωρίζονται μεταξύ τους από έναν N-bit interleaver. Το σχήμα περιγράφει ένα τέτοιο σχήμα κωδικοποίησης. Στην περίπτωση, που δεν περιλαμβάνεται ένας μηχανισμός απαλοιφής των σειριακών

δεδομένων εξόδου, ο ρυθμός του κωδικοποιητή είναι ίσος με 1/3, και τα N bit κωδικοποιούνται σε 3N κωδικά bit.

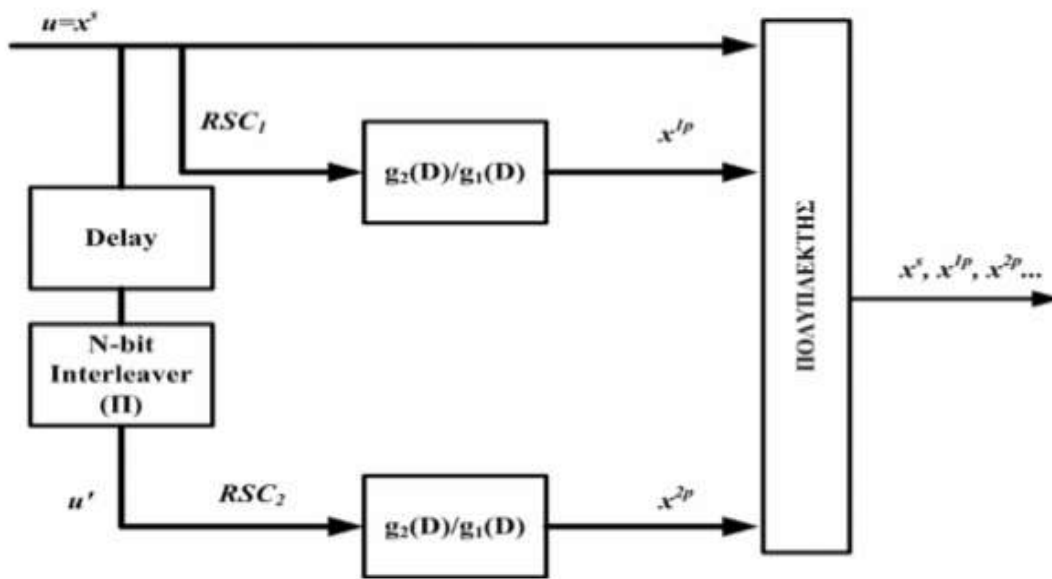
Όπως φαίνεται από το σχήμα, οι κωδικοποιητές διευθετούνται σε παράλληλη σύνδεση (parallel concatenation). Οι επιμέρους κωδικοποιητές ανήκουν στην κατηγορία των επαναληπτικών και συστηματικών συνελκτικών κωδίκων και είναι, κατά γενικό κανόνα, ταυτόσημοι. Ο πολυπλέκτης αφού μετατρέψει τα δεδομένα σε σειριακά, τα τροφοδοτεί στην επόμενη βαθμίδα του τηλεπικοινωνιακού συστήματος.

Η δομή Delay χρησιμοποιείται για τον συγχρονισμό κωδικοποιητή και αποκωδικοποιητή και ισούται με την καθυστέρηση, που εισάγει ο ένας MAP αποκωδικοποιητής [5, 7]. Όπως προαναφέρθηκε, ο γεννήτορας πίνακας ενός επαναληπτικού συστηματικού συνελκτικού κώδικα ρυθμού 1/2 δίδεται ως ακολούθως:

$$G_{RSC} = \begin{bmatrix} 1 & \frac{g_2(D)}{g_1(D)} \end{bmatrix}$$

Παρατηρούμε πως, για τον παραπάνω κώδικα, η ακολουθία των κωδικών λέξεων, που προκύπτει, θα είναι πεπερασμένου βάρους αν και μόνον αν η ακολουθία της εισόδου διαιρείται από το πολώνυμο $g_1(D)$. Συνεπώς, μια είσοδος πληροφοριακών δεδομένων βάρους ένα θα παράγει μια κωδική λέξη άπειρου βάρους, δεδομένου ότι μια τέτοια είσοδος δεν μπορεί να διαιρεθεί ποτέ από το πολώνυμο $g_1(D)$. Η είσοδος βάρους-ένα θα δημιουργήσει μια διαδρομή στο trellis, που αποκλίνει από την όλο-μηδενική διαδρομή, αλλά ποτέ δεν επανασυγκλίνει. Επίσης, για κάποιο πολώνυμο $g_1(D)$, υπάρχει μια οικογένεια εισόδων βάρους-δύο της μορφής $D^j(1+D^q-1)$, $j \geq 0$, η οποία παράγει εξόδους πεπερασμένου βάρους (η είσοδος διαιρείται από το πολώνυμο $g_1(D)$). Όταν το $g_1(D)$ είναι πρωτογενή πολώνυμο βαθμού m, τότε $q=2m$, όπου το $q-1$ είναι το μήκος της ψευδοτυχαίας ακολουθίας παραγόμενη από το $g_1(D)$. Αυτό δηλώνει ότι πάντα θα υπάρχει μια διαδρομή στο trellis, το οποίο θα αποκλίνει και θα συγκλίνει αργότερα και θα αντιστοιχεί σε μια

ακολουθία δεδομένων βάρους-δύο. Συμπεραίνουμε, λοιπόν, πως αν χρησιμοποιούσαμε μη-επαναληπτικούς συνελκτικούς κώδικες, τότε για πληροφοριακή ακολουθία δεδομένων βάρους ίσου με ένα θα είχαμε τη δημιουργία μιας κωδικής λέξης από τον Turbo κωδικοποιητή πολύ χαμηλού βάρους (η κωδική λέξη που παράγει ο μη-επαναληπτικός συνελκτικός κώδικας για τέτοια είσοδο ισούται με τα διανύσματα του γεννήτορα πίνακά του). Επομένως, εισάγοντας RSC κώδικες στους Turbo κώδικες, αυξάνουμε τη μέγιστη ελάχιστη απόσταση του κώδικα με απόρροια την επίτευξη ενισχυμένης διορθωτικής ικανότητας.



Ο interleaver (II) είναι μια συνάρτηση, η οποία λαμβάνει μια δομή N δεδομένων και την επαναδιατάσσει βάσει μιας μήτρας και την τροφοδοτεί στον δεύτερο RSC. Οι δύο βασικές απαιτήσεις για την κατάλληλη και αποδοτική λειτουργία ενός interleaver είναι πρώτον, να επαναδιατάσσει μεγάλο αριθμό δεδομένων και δεύτερον, η επαναδιάταξη αυτή να γίνεται με όσον το δυνατόν πιο τυχαίο τρόπο. Ο αριθμός N των δεδομένων καλείται ως μήκος των πληροφοριακών δεδομένων. Ο interleaver προστατεύει το τηλεπικοινωνιακό σύστημα από σφάλματα ριπής, που προκαλεί το κανάλι, διότι, ουσιαστικά, αποσυσχετίζει τα δεδομένα εισόδου. Επομένως, δίδεται η δυνατότητα της δημιουργίας (με υψηλή πιθανότητα) διαφορετικών ακολουθιών από bit ισοτιμίας και η κωδική λέξη προέρχεται από δύο διακριτές διαδρομές σε κάθε έναν από τους δύο RSC κωδικοποιητές. Εφόσον, οι κωδικοποιητές είναι της μορφής RSC δεν είναι δυνατόν ο τερματισμός και των δύο ταυτοχρόνως στην μηδενική κατάσταση.

Η κοινή πρακτική βασίζεται στον τερματισμό του ενός από τους δύο κωδικοποιητές (αποστέλλουμε, επιπλέον, v bits ίσο με την μνήμη του RSC) ή για την μη ελάττωση του ρυθμού του Turbo κώδικα διατηρούμε και τους δύο RSC χωρίς τερματισμό. Στην πρώτη περίπτωση, η εισαγωγή των v bits αυξάνει το βάρος της κωδικής λέξης ενώ στην δεύτερη, επιλέγουμε κατάλληλες δομές interleaver για την αποφυγή δημιουργίας κωδικών λέξεων μικρού βάρους. Οι ιδιότητες ενός Turbo κώδικα οφείλονται στο γεγονός ότι ο κωδικοποιητής προσφέρει ταυτόχρονα τυχαιότητα αλλά ταυτοχρόνως παράγει κωδικές λέξεις βασιζόμενος σε αλγεβρική δομή. Οι δύο RSC κωδικοποιητές είναι γραμμικοί, μ' αποτέλεσμα να είναι δυνατή η χρήση της πλήρους-μηδενικής ακολουθίας ως αναφορά.

Τα όρια της απόδοσης των Turbo κωδίκων αναπτύχθηκαν θεωρώντας κώδικες trellis, των οποίων το μήκος του trellis είναι μη πεπερασμένο. Για τον διαχωρισμό μεταξύ των Turbo κωδίκων και των συνελκτικών κωδίκων, είναι χρήσιμο να θεωρήσουμε και τους δύο κώδικες ως κώδικες δομής περιορίζοντας το μήκος των πληροφοριακών bit ίσο με το μήκος του interleaver. Με τη χρήση πεπερασμένου μήκους N ακολουθιών πληροφοριακών δεδομένων, ένας $(2, 1, v)$ συνελκτικός κώδικας μπορεί να θεωρηθεί ως κώδικας δομής με $2N$ κωδικές λέξεις μήκους $2(N+v)$. Τα τελευταία v bits χρησιμοποιούνται για να ωθήσουν τον κωδικοποιητή στην μηδενική του κατάσταση και ονομάζονται bit ουράς (tail bits). Η BER απόδοση P_b ενός πεπερασμένου μήκους συνελκτικού κώδικα με αποκωδικοποίηση μεγίστης πιθανοφάνειας (maximum likelihood) και διαμόρφωση σήματος BPSK για διάλο AWGN με SNR ίσο με E_b/N_0 είναι άνω φραγμένη ως εξής

$$P_b \leq \sum_{i=1}^{2^N} \frac{w_i}{N} Q\left(\sqrt{d_i \frac{2RE_b}{N_0}}\right)$$

όπου w_i είναι το βάρος της ακολουθίας των πληροφοριακών bits και d_i το βάρος της ισοτής κωδικής λέξης. Συγκεντρώνοντας τις κωδικές λέξεις, που έχουν το ίδιο βάρος, και ορίζοντας το πληροφοριακό βάρος μέσου όρου ανά κωδική λέξη ως ακολούθως:

$$\tilde{w}_d = w_d / N_d$$

όπου w_d είναι το συνολικό πληροφοριακό βάρος όλων των κωδικών λέξεων βάρους d και N_d είναι ο αριθμός, ή πολλαπλότητα, των κωδικών λέξεων βάρους d , τότε οδηγούμαστε στην εξίσωση 2-32, με d_{free} να αποτελεί την ελεύθερη απόσταση του κώδικα.

$$P_b \leq \sum_{d=d_{free}}^{\infty} \frac{N_d \tilde{w}_d}{N} Q \left(\sqrt{d \frac{2RE_b}{N_0}} \right)$$

Αν ένας συνελκτικός κώδικας έχει $0 < d_{free} < \infty$ κωδικές λέξεις βάρους d , που παράγονται από τις πληροφοριακές ακολουθίες $u(D)$, εκ των οποίων το πρώτο ένα εμφανίζεται στην χρονική στιγμή 0, τότε έχει $0 < d_{free} < \infty$ κωδικές λέξεις βάρους d , οι οποίες δημιουργούνται από τις πληροφοριακές ακολουθίες D $u(D)$, $0 < d_{free} < \infty$ κωδικές λέξεις βάρους d , που προέρχονται από τις πληροφοριακές ακολουθίες D^2 $u(D)$, κ.ο.κ. Συνεπώς, για $1 \leq d \leq d_{free}$, έχουμε

$$\lim_{N \rightarrow \infty} \frac{N_d}{N} = N_d, \quad \lim_{N \rightarrow \infty} \tilde{w}_d = \lim_{N \rightarrow \infty} \frac{w_d}{N_d} = \frac{w_d^0}{N_d} \equiv \tilde{w}_d^0$$

όπου $0 < dw$ είναι το συνολικό πληροφοριακό βάρος όλων των κωδικών λέξεων με βάρος d , τα οποία δημιουργούνται από πληροφοριακές ακολουθίες των οποίων το πρώτο ένα λαμβάνει χώρα την χρονική στιγμή 0. Για $d_{free} \leq d < \infty$, η ανάλυση είναι πιο πολύπλοκη εξαιτίας του γεγονότος ότι οι κωδικές λέξεις μπορεί να αποτελούνται

από πολλαπλά λανθασμένα γεγονότα. Όταν $\infty \rightarrow N$, $() mNNNN ddd + = 0$, όπου $()$

$mN d$ είναι ο αριθμός των κωδικών λέξεων βάρους d δημιουργούμενων από m λανθασμένων συμβάντων. Το όριο πάνω στο BER ενός πεπερασμένου μήκους τερματισμένου συνελκτικού κώδικα με ML αποκωδικοποίηση γίνεται

$$P_b \leq \sum_{d=d_{free}}^{2d_{free}-1} N_d^0 \tilde{W}_d^0 Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right) + \sum_{d=2d_{free}}^{\infty} \frac{N_d \tilde{W}_d}{N} Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right)$$

$$= \sum_{d=d_{free}}^{2d_{free}-1} w_d^0 Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right) + \sum_{d=2d_{free}}^{\infty} \frac{N_d \tilde{W}_d}{N} Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right)$$

Συμπερασματικά, η εύρεση αποτελεσματικών συνελκτικών κωδικών, για χρήση με ML αποκωδικοποίηση, εστιάζεται στην εξεύρεση κωδικών, που μεγιστοποιούν την ελεύθερη απόσταση d_{free} και ελαχιστοποιεί τον αριθμό των μονοπατιών ελεύθερης απόστασης $0 < d < d_{free}$ για δεδομένο ρυθμό και μνήμη του κωδικοποιητή.

Η απόδοση ενός Turbo κώδικα με αποκωδικοποίηση βασισμένη στον αλγόριθμο ML φράζεται επίσης από την εξίσωση 2-30. Συγκεντρώνοντας τις κωδικές λέξεις που έχουν το ίδιο Hamming βάρος, το άνω όριο πάνω στον ρυθμό BER για έναν Turbo κώδικα δίνεται επίσης από την ακόλουθη εξίσωση και ονομάζεται φράγμα ένωσης (union bound)

$$P_b \leq \sum_{d=d_{free}}^{\infty} \frac{N_d \tilde{W}_d}{N} Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right)$$

Οι Turbo κώδικες, όμως, εξαιτίας του interleaver, ο οποίος αντιστοιχεί, τις ακολουθίες εισόδου $u(D)$ σε $u'(D)$ και τις ακολουθίες εισόδου $Du(D)$ σε μια ακολουθία $u''(D)$, η οποία είναι με υψηλή πιθανότητα διαφορετική από την ακολουθία $D'u(D)$. Σε αντίθεση με τους συνελκτικούς κώδικες, οι ακολουθίες

εισόδου $u(D)$ και $Du(D)$ παράγουν διαφορετικές κωδικές λέξεις με διαφορετικά βάρη Hamming. Επομένως, η ποσότητα $d \sim \sqrt{N}$ είναι

πολύ μικρότερη του N για κωδικές λέξεις μικρού βάρους και ισχύει: $1 \sim \leq N \sim d$ όπου ο όρος

$N \sim d$ καλείται ως αποτελεσματική πολλαπλότητα (effective distance) των κωδικών λέξεων βάρους d . Για μέτριους έως υψηλούς λόγους SNR, ο όρος της ελεύθερης απόστασης (ελάχιστη απόσταση Hamming μεταξύ των κωδικών λέξεων) του κώδικα d_{free} στο φράγμα της ένωσης για την BER απόδοσή του είναι καθοριστικός. Η απόδοση ενός Turbo κώδικα προσεγγίζει ασυμπτωτικά την ακόλουθη εξίσωση:

$$P_b \approx \frac{N_{free} \tilde{w}_{free}}{N} Q \left(\sqrt{d_{free} \frac{2RE_b}{N_0}} \right)$$

όπου N_{free} είναι η πολλαπλότητα των κωδικών λέξεων της ελεύθερης απόστασης και \tilde{w}_{free} είναι το μέσου όρου βάρος των πληροφοριακών ακολουθιών που οδηγούν στις κωδικές λέξεις της ελεύθερης απόστασης. Το δεύτερο μέρος της παραπάνω εξίσωσης και η αντίστοιχη γραφική του παράσταση καλείται ασύμπτωτη ελεύθερης απόστασης (free distance asymptote), P_{free} ενός Turbo κώδικα. Ο Turbo κώδικας (37,21,65536) [5] έχει $N_{free}=3$ διαδρομές με βάρος $d_{free}=6$. Η κάθε μια από τις διαδρομές δημιουργήθηκε από μια ακολουθία εισόδου βάρους 2 και επομένως, $\tilde{w}_{free} = 2$. Αυτό το αποτέλεσμα οφείλεται στον ψευδοτυχαίο interleaver με $N=65536$ [5, 7] και η ασύμπτωτη της ελεύθερης απόστασης δίνεται ως εξής:

$$P_b = \frac{3 \times 2}{65536} Q \left(\sqrt{6 \frac{E_b}{N_0}} \right)$$

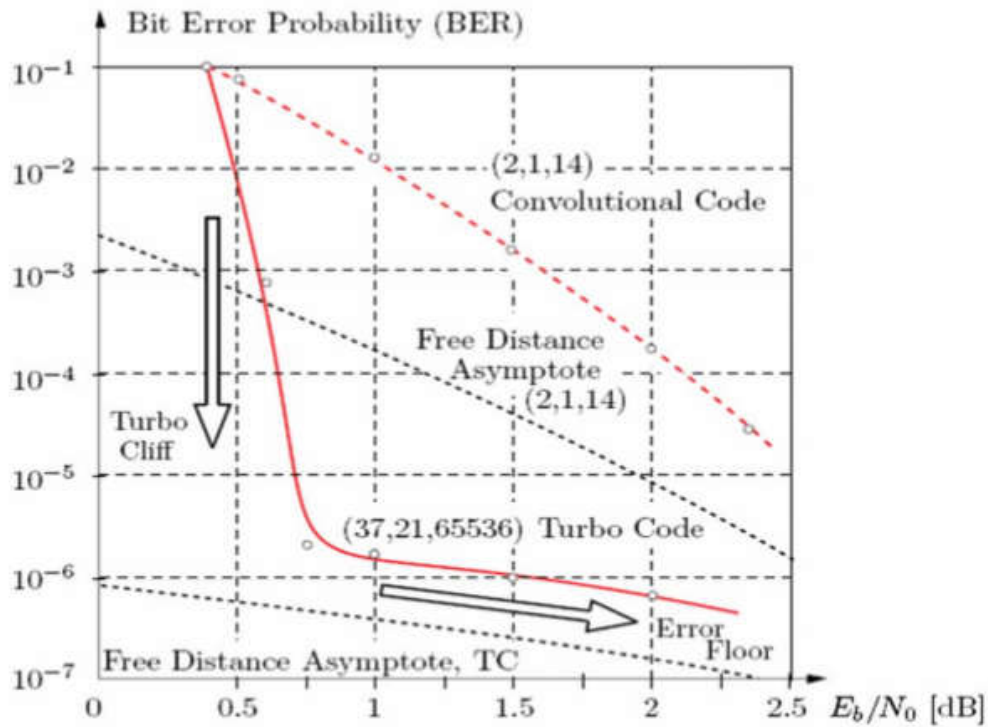
όπου η απώλεια στο ρυθμό λόγω των τεσσάρων bit ουράς έχει αγνοηθεί και

$65536 \sim 3 =$

$N \sim N_{free}$

είναι η effective multiplicity. Η γραφική παράσταση της ασύμπτωτης της ελεύθερης απόστασης και της απόδοσης BER του συγκεκριμένου Turbo κώδικα (μετά από 18 επαναλήψεις αποκωδικοποίησης) και ενός (2, 1, 14) συνελκτικού κώδικα φαίνεται στο σχήμα. Η επαναληπτική συμπεριφορά των επιμέρους αποκωδικοποιητών μελετείται στατιστικά μετρώντας την βελτίωση της αμοιβαίας πληροφορίας σε κάθε επανάληψη.

Οι μετρήσεις αυτές ποσοτικοποιούνται με την βοήθεια των διαγραμμάτων ανταλλαγής της εξωγενούς πληροφορίας (extrinsic information exchange, EXIT charts)



Στο προηγούμενο σχήμα παρατηρούμε το σημαντικό ρόλο που διαδραματίζει η ελεύθερη απόσταση στην απόδοση του Turbo κώδικα. Η απότομη κλίση (Turbo cliff) της χαρακτηριστικής του ρυθμού BER για μικρές τιμές λόγου σήματος προς θόρυβο δείχνει την εκπληκτική απόδοση του κώδικα καθώς προσεγγίζει τα 10^{-6} σχεδόν αμέσως την ασύμπτωτη ελεύθερης απόστασης.

Στην συνέχεια, με την αύξηση της τιμής του λόγου σήματος προς θόρυβο η χαρακτηριστική αυτή ακολουθεί ασυμπτωτικά τη χαρακτηριστική της ασύμπτωτης ελεύθερης απόστασης και παρατηρείται ένας κορεσμός στην τιμή του BER, που επιτυγχάνεται. Αυτήν η περιοχή ονομάζεται error floor-flare region (περιοχή διαπλάτυνσης ή λάθους) και η απόδοσή του κώδικα περιορίζεται από την ελεύθερη απόστασή του.

Επομένως, σχετικά μικρή ελεύθερη απόσταση οδηγεί σε μεγαλύτερη error floor. Από την εξίσωση 2-37 διαπιστώνουμε ότι αυξάνοντας το μήκος του interleaver διατηρώντας όμως, την ελεύθερη απόσταση και την πολλαπλότητα στα ίδια επίπεδα τιμών, η χαρακτηριστική της ασύμπτωτης ελεύθερης απόστασης χαμηλώνει. Αν διατηρήσουμε σταθερή την τιμή του interleaver αυξάνοντας την ελεύθερη απόσταση του κώδικα, διαλέγοντας το πολυώνυμο της ανάδρασης να είναι πρωτογενές.

ΕΜΠΕΙΡΙΚΗ ΔΙΕΡΕΥΝΗΣΗ

Το MATLAB αποτελεί ένα εμπορικό εργαλείο το οποίο προσφέρει ένα διαδραστικό προγραμματιστικό περιβάλλον στον χρήστη και χρησιμοποιείται σε ένα μεγάλο εύρος εφαρμογών. Ενσωματώνει μια υψηλού επιπέδου γλώσσα προγραμματισμού, κατάλληλη για τη μοντελοποίηση και επίλυση σύνθετων μαθηματικών, και όχι μόνο, προβλημάτων. Το όνομά του προέρχεται από τις λέξεις MATrix LABoratory (εργαστήριο πινάκων) πράγμα που υποδηλώνει το γεγονός ότι η λειτουργία του βασίζεται εξ ολοκλήρου στη χρήση πινάκων, στοιχεία των οποίων μπορεί να είναι πραγματικοί ή μιγαδικοί αριθμοί (ακόμα και ένας μεμονωμένος αριθμός, όπως για παράδειγμα το 8, θεωρείται ως πίνακας με ένα στοιχείο). Ανάμεσα σε ένα πλήθος άλλων ευκολιών που προσφέρει, επιτρέπει τον εύκολο χειρισμό πινάκων, τη γραφική απεικόνιση (plotting) συναρτήσεων και δεδομένων, την υλοποίηση αλγορίθμων, την δημιουργία γραφικών διεπαφών και τη συνεργασία και διαλειτουργικότητα με προγράμματα γραμμένα σε άλλες γλώσσες προγραμματισμού. Λόγω, δε, του ότι το MATLAB βρίσκει εφαρμογή σε ποικίλα επιστημονικά πεδία (επεξεργασία σήματος, νευρωνικά δίκτυα, συστήματα ελέγχου κ.τ.λ.), πρόσθετα πακέτα, που ονομάζονται toolboxes, ενσωματώνονται σε αυτό και προσφέρουν χρήσιμες εξειδικευμένες συναρτήσεις.

Τα βασικά συστατικά του MATLAB είναι:

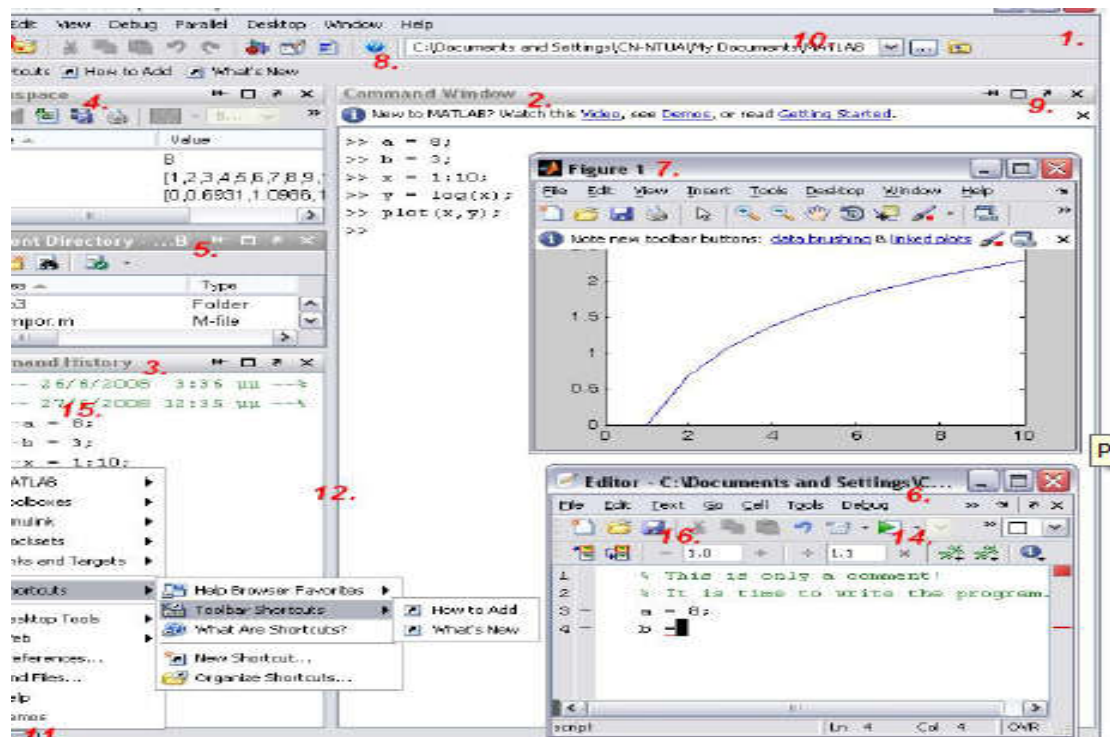
λ το Περιβάλλον Ανάπτυξης και τα αντίστοιχα εργαλεία του: περιλαμβάνει διάφορα παράθυρα, όπως το Παράθυρο Εντολών (Command Window) και το Ιστορικό Εντολών (Command History) και άλλα εργαλεία για αποσφαλμάτωση (debugging), ανάλυση κώδικα και πλοήγηση στο σύστημα αρχείων

λ η βιβλιοθήκη μαθηματικών συναρτήσεων: ίσως το πιο σημαντικό συστατικό του MATLAB, με στοιχειώδεις αριθμητικές συναρτήσεις, αλλά και πιο πολύπλοκες

λ η γλώσσα προγραμματισμού: μια υψηλού επιπέδου προγραμματιστική γλώσσα με δομές δεδομένων, συναρτήσεις, εντολές ελέγχου ροής, εντολές εισόδου/εξόδου και στοιχεία από αντικειμενοστραφείς γλώσσες προγραμματισμού

λ τα γραφικά συστατικά: το MATLAB παρέχει μια πληθώρα δυνατοτήτων απεικόνισης διανυσμάτων, πινάκων και γραφημάτων στις 2 και 3 διαστάσεις

Στο σημείο αυτό κρίνουμε απαραίτητη τη συνοπτική περιγραφή του Περιβάλλοντος Ανάπτυξης (Development Environment) του MATLAB, με σκοπό τη γρήγορη εξοικείωση του χρήστη με αυτό άρα και την αποδοτικότερη εκμάθηση αυτού. Με αναφορά στο επόμενο δεικτοδοτημένο σχήμα, περιγράφουμε κάποια από τα βασικά συστατικά της επιφάνειας εργασίας του MATLAB.



MATLAB Desktop (Επιφάνεια Εργασίας MATLAB) [1.]

Το MATLAB Desktop (Επιφάνεια Εργασίας MATLAB) αποτελεί τη βάση του Περιβάλλοντος Ανάπτυξης του συγκεκριμένου εργαλείου. Παρέχει στο χρήστη ένα προγραμματιστικό περιβάλλον το οποίο του επιτρέπει την εκτέλεση εντολών, την οπτικοποίηση δεδομένων και αποτελεσμάτων, τη δημιουργία και τροποποίηση αρχείων m-files (απλά αρχεία κειμένου με κατάληξη .m που περιέχουν εντολές MATLAB) και την λήψη στοχευμένης βοήθειας. Επίσης, παρέχει εργαλεία διαχείρισης και προσωποποίησης των εκάστοτε αναπτυσσόμενων projects. Αποτελείται από έναν αριθμό άλλων παραθύρων των οποίων η χρησιμότητα θα αναπτυχθεί στην συνέχεια.

Παράθυρο Εντολών (Command Window) [2.]

Το Παράθυρο Εντολών χρησιμοποιείται για την εκτέλεση MATLAB εντολών με σκοπό τη διεκπεραίωση ενός αριθμού εργασιών όπως για παράδειγμα την

δημιουργία και επεξεργασία μεταβλητών (variables), τη δημιουργία, τροποποίηση και κλήση αρχείων m-files, τη διαχείριση δεδομένων και πολλά άλλα. Στο συγκεκριμένο παράθυρο, η εκτέλεση της εκάστοτε εντολής πραγματοποιείται μόλις πιεστεί το πλήκτρο “Enter” ενώ τα αποτελέσματα της εκτέλεσης είναι άμεσα ορατά στον χώρο αμέσως κάτω από την εκτελούμενη εντολή (εκτός και αν τοποθετηθεί ένα semicolon (ελληνικό ερωτηματικό, ;) στο τέλος κάθε εντολής κάνοντας τα αποτελέσματα μη ορατά στον χρήστη). Επίσης, σημειώνουμε ότι αν τη στιγμή που γράφουμε μια εντολή στην γραμμή εκτέλεσης πιέσουμε το πλήκτρο “Tab”, εμφανίζονται όλες οι πιθανές επιλογές εντολών, συναρτήσεων και μεταβλητών που αρχίζουν με τους ως τώρα εισηγμένους χαρακτήρες. Επίσης, είναι δυνατή η επανάκληση προηγούμενα εκτελεσμένων εντολών απλά χρησιμοποιώντας τα Up & Down Arrow Keys (πάνω και κάτω βελάκι).

Ιστορικό Εντολών (Command History) [3.]

Κάθε εντολή που εκτελείται στο Παράθυρο Εντολών (Command Window) καταγράφεται ταυτόχρονα σε ένα ιστορικό. Οι εγγραφές του ιστορικού [15.] εμφανίζονται, ταξινομημένες σύμφωνα με την ώρα δημιουργία τους, στο παράθυρο το οποίο ονομάζεται Ιστορικό Εντολών. Είναι έτσι δυνατή η επανεκτέλεση μιας ήδη εισηγμένης εντολής απλά κάνοντας διπλό-κλικ πάνω της ή σέρνοντάς τη στο Παράθυρο Εντολών για περαιτέρω επεξεργασία. Οι εγγραφές του Ιστορικού Εντολών, τέλος, μπορούν να διαγραφούν, να ομαδοποιηθούν, να τροποποιηθούν ή να σωθούν σε ένα m-file.

Χώρος Εργασίας (Workspace) [4.]

Όλες οι μεταβλητές που έχουν οριστεί από τη στιγμή εκκίνησης του MATLAB είναι αποθηκευμένες σε χώρο μνήμης προσβάσιμο από όλα τα προγράμματα και αποτελούν το Χώρο Εργασίας (Workspace). Κάθε μεταβλητή αποθηκεύεται εκεί μαζί με πληροφορίες όπως το όνομά της, η τιμή της, στατιστικά στοιχεία και άλλα. Οι πληροφορίες αυτές γίνονται εμφανείς στον χρήστη μέσω του Workspace Browser. Δυνατότητες όπως η διαδραστική ανάγνωση και τροποποίηση των χαρακτηριστικών των αποθηκευμένων μεταβλητών (με διπλό-κλικ επάνω στην επιθυμητή μεταβλητή και χρήση του Array Editor) και η γραφική απεικόνισή τους (plotting) είναι μερικές από τις παρεχόμενες από αυτόν ευκολίες. Η συμβολή του,

δε, κατά την διαδικασία της αποσφαλμάτωσης (debugging) προγραμμάτων είναι καταλυτική.

Φυλλομετρητής Τρέχοντος Καταλόγου (Current Directory Browser) [5.] / Τρέχων Κατάλογος (Current Directory) [10.]

Κάθε λειτουργία η οποία στηρίζεται στην χρήση αρχείων, χρησιμοποιεί τον Τρέχοντα Κατάλογο (Current Directory) ως σημείο αναφοράς. Κάθε αρχείο που επιθυμούμε να εκτελέσουμε ή να χρησιμοποιήσουμε θα πρέπει αναγκαστικά να βρίσκεται μέσα στο συγκεκριμένο φάκελο εκτός και αν εμείς το ορίσουμε διαφορετικά. Στο περιβάλλον εργασίας του MATLAB, λοιπόν, παρέχονται δύο κύρια εργαλεία, ο Φυλλομετρητής Τρέχοντος Καταλόγου (Current Directory Browser) και το πεδίο Τρέχων Κατάλογος (Current Directory). Το πρώτο παρουσιάζει εύληπτα τα περιεχόμενα του δεύτερου παρέχοντας ταυτόχρονα πολλές πληροφορίες για το καθένα. Παράλληλα, επιτρέπει την περιήγηση του χρήστη στο σύστημα αρχείων του εκάστοτε υπολογιστή, το άνοιγμα (με το κατάλληλο προκαθορισμένο πρόγραμμα), την αναζήτηση και τροποποίηση σχετικών αρχείων ή καταλόγων καθώς και την διαχείριση m-files.

Παράθυρο Εισαγωγής Κειμένου (Editor window) [6.]

Το παρόν παράθυρο καθιστά δυνατή τη συγγραφή μιας δέσμης εντολών και συναρτήσεων και την αποθήκευσή τους ως m-file, με σκοπό την μετέπειτα εκτέλεση. Το εν λόγω παράθυρο εμφανίζεται με το πάτημα του κουμπιού NEW [13.] ή εναλλακτικά πληκτρολογώντας edit στο Παράθυρο Εντολών, ενώ αφού συντάξουμε το πρόγραμμα, μπορούμε είτε να το αποθηκεύσουμε στον Τρέχοντα Κατάλογο (Current Directory) πιέζοντας το κουμπί SAVE [16.] είτε να το εκτελέσουμε άμεσα πιέζοντας το κουμπί RUN [14.]. Τα ως τώρα δημιουργημένα m-files παρουσιάζονται στον Φυλλομετρητή Τρέχοντος Καταλόγου (Current Directory Browser) από τον οποίο είναι, άλλωστε, και διαθέσιμα προς εκτέλεση, τροποποίηση, ανάλυση και αποσφαλμάτωση.

Παράθυρο Σχημάτων (Figure window) [7.]

Στο Παράθυρο Σχημάτων σχεδιάζουμε γραφικές παραστάσεις με τη χρήση κατάλληλων σχεδιαστικών εντολών οι οποίες γράφονται είτε στη γραμμή εντολών είτε σε προγράμματα MATLAB.

Παράθυρο Βοήθειας (Help Window)

Το παράθυρο αυτό παρέχει πολύτιμη βοήθεια τόσο σε αρχάριους όσο και πιο έμπειρους χρήστες καθώς αποτελεί ένα θησαυρό οδηγιών και παραδειγμάτων που αφορούν το εργαλείο MATLAB και όχι μόνο. Καθίσταται προσβάσιμο απλά πιέζοντας το κουμπί [8] ή εναλλακτικά, πληκτρολογώντας στο Παράθυρο Εντολών `help` ή `doc` ακολουθούμενα από το όνομα της συνάρτησης για την οποία θέλουμε να λάβουμε πληροφορίες.

Γενικές οδηγίες

Κρατώντας πατημένο το αριστερό πλήκτρο του ποντικιού πάνω στα όρια του εκάστοτε παραθύρου, π.χ. στο σημείο [12.], και μετακινώντας αυτό κατάλληλα μπορούμε να αλλάξουμε το μέγεθος ή ακόμα και τη θέση του στο MATLAB Desktop. Επίσης, στο χώρο [9.] παρέχονται κουμπιά επιλογών όπως το κλείσιμο του συγκεκριμένου παραθύρου, η “επικόλλησή” του στο Desktop και πολλά άλλα.

Τέλος, μέσω του κουμπιού Start [11.] είναι διαθέσιμα διάφορα εργαλεία του MATLAB, παραδείγματα, demos, αρχεία τεκμηρίωσης, παράθυρα τροποποίησης και ορισμού επιλογών, παράθυρα προσωποποίησης του περιβάλλοντος εργασίας, ακόμα και τα πρόσθετα (add-ons) εκείνα εργαλεία τα οποία ενδέχεται ο εκάστοτε χρήστης, ανάλογα με τις ανάγκες του, να έχει εγκαταστήσει. Τέλος, μέσω αυτού του κουμπιού καθίστανται προσβάσιμα τα διάφορα shortcuts εντολών και εργαλείων που ο χρήστης χρησιμοποιεί πιο συχνά.

Βασικές εντολές στο Περιβάλλον Εργασίας

Κάποιες βασικές εντολές που μπορεί να εισάγει ο χρήστης για να δει πληροφορίες είναι η `clear` ακολουθούμενη από το όνομα μιας μεταβλητής και η `clear all`, οι οποίες διαγράφουν μια συγκεκριμένη μεταβλητή ή όλες τις μεταβλητές που βρίσκονται στο Χώρο Εργασίας αντίστοιχα. Η χρήση της `clear all` πριν από την εκτέλεση κάθε νέου προγράμματος κρίνεται απαραίτητη, για να αποφευχθούν προβλήματα ή τυχόν σφάλματα από παλιότερες ανεπιθύμητες τιμές μεταβλητών. Οι εντολές `who` και `whos` εμφανίζουν μια λίστα με όλες τις μεταβλητές που βρίσκονται στο Χώρο Εργασίας μια δεδομένη χρονική στιγμή, με τη `whos` να εμφανίζει επιπλέον και πληροφορίες όπως το μέγεθος και ο τύπος κάθε μεταβλητής. Βέβαια, υπενθυμίζουμε ότι όλες αυτές οι πληροφορίες μπορούν ανά πάσα στιγμή να διαβαστούν από τον Workspace Browser.

Επιπλέον, σημειώνουμε ότι δεν είναι απαραίτητο κάθε εντολή να βρίσκεται σε μια γραμμή, καθώς μπορούμε να πληκτρολογήσουμε περισσότερες της μίας εντολές στην ίδια γραμμή, παρεμβάλλοντας μεταξύ τους ένα ελληνικό ερωτηματικό ;. Αντίστροφα, αν μια εντολή είναι πολύ μεγάλη για να χωρέσει σε μία γραμμή, μπορούμε στο τέλος της πρώτης γραμμής να πληκτρολογήσουμε τρεις τελείες ... και να συνεχίσουμε την εντολή στην επόμενη γραμμή. Η εισαγωγή σχολίων στο MATLAB σηματοδοτείται με το χαρακτήρα %, οπότε ότι έπεται αυτού του χαρακτήρα δε λαμβάνεται υπόψη από το MATLAB και δεν εκτελείται.

Αρχεία προγραμμάτων και συναρτήσεων

Γενικά, το MATLAB προσφέρει δύο εναλλακτικούς τρόπους εργασίας: είτε μέσω του Παραθύρου Εντολών (Command Window) όπου ο χρήστης δίνει μεμονωμένες εντολές, μπορεί να κάνει κάποιες δοκιμές και παίρνει άμεσα κάποια αποτελέσματα, είτε μέσω scripts, προγραμμάτων δηλαδή, τα οποία ο χρήστης γράφει στο περιβάλλον του Παραθύρου Εισαγωγής Κειμένου (Editor). Τα προγράμματα MATLAB αποθηκεύονται σε αρχεία απλού κειμένου (plain text) με την κατάληξη .m και ονομάζονται m-files. Τα ονόματα των m-files πρέπει να αρχίζουν με κάποιο γράμμα ακολουθούμενο από άλλα γράμματα (μέχρι 18) ή/και ψηφία ή/και το χαρακτήρα υπογράμμισης (το μείον «-» και η τελεία «.» δεν επιτρέπονται). Για την εκτέλεση του προγράμματος, δεν απαιτείται να γίνει πρώτα από το χρήστη μεταγλώττιση (compiling), όπως απαιτούν άλλες γλώσσες προγραμματισμού, αλλά αυτή γίνεται αυτόματα πριν από την εκτέλεση.

Υπάρχουν δύο είδη m-files: τα αρχεία προγραμμάτων (scripts) και τα αρχεία συναρτήσεων (functions). Ένα script είναι ένα αυτόνομο, εκτελέσιμο πρόγραμμα το οποίο δε δέχεται ορίσματα εισόδου και δεν επιστρέφει μεταβλητές εξόδου, αλλά αντίθετα χρησιμοποιεί υπάρχουσες ή δημιουργεί νέες μεταβλητές στο Χώρο Εργασίας (Workspace). Ουσιαστικά, τα scripts δεν είναι τίποτα άλλο από μια σειρά εντολών, ενώ αν και δεν έχουν μεταβλητές εξόδου, μπορούν να δημιουργήσουν νέες μεταβλητές στο Χώρο Εργασίας, που μπορούν να προσπελαστούν μόλις τελειώσει η εκτέλεσή τους. Τα scripts εκτελούνται πληκτρολογώντας το όνομά τους στο Παράθυρο Εντολών ή πατώντας το κουμπί RUN στο παράθυρο του editor.

Το δεύτερο είδος m-file είναι τα αρχεία συναρτήσεων ή functions, που δέχονται ορίσματα εισόδου και προαιρετικά, επιστρέφουν μεταβλητές εξόδου. Το όνομα μιας

συνάρτησης μαζί με τα ορίσματα εισόδου και τις μεταβλητές εξόδου γράφονται στην πρώτη γραμμή του αντίστοιχου m-file συνοδευόμενα από τη λέξη-κλειδί function, ενώ στην τελευταία γραμμή του αρχείου, γράφεται η λέξη-κλειδί end. Για παράδειγμα, η πρώτη γραμμή της συνάρτησης polyval, που είναι αποθηκευμένη στο αρχείο polyval.m θα έχει ως εξής:

```
function y=polyval(p,x)
```

που σημαίνει ότι η συνάρτηση polyval παίρνει δύο ορίσματα εισόδου, τα p και x και επιστρέφει μια τιμή που αποθηκεύεται στη μεταβλητή εξόδου y. Επίσης, το m-file που περιέχει μια συνάρτηση πρέπει να έχει το ίδιο όνομα με αυτή, ενώ αξίζει να σημειώσουμε ότι μια συνάρτηση χρησιμοποιεί δικό της χώρο μεταβλητών, ο οποίος δεν είναι προσπελάσιμος μετά την εκτέλεσή της. Οι συναρτήσεις μπορούν να κληθούν μέσα από scripts ή από άλλες συναρτήσεις για διάφορες τιμές των ορισμάτων εισόδου τους.

Εισαγωγή δεδομένων – Διαμόρφωση Εξόδου

Το MATLAB χρησιμοποιεί τη συνήθη δεκαδική παράσταση αριθμών. Η περιοχή που μπορεί να παρασταθεί είναι προσεγγιστικά η $\pm(10^{-308}, 10^{308})$. Πολύ μικροί ή πολύ μεγάλοι αριθμοί παριστάνονται σε εκθετική μορφή (ή κινητής υποδιαστολής). Τυπικά παραδείγματα παραστάσεων αριθμών στο MATLAB είναι 67423.97215 2468.1357e-4 -0.1234e3 με το e3 να ισούται με 103. Όλες οι πράξεις στο MATLAB γίνονται εσωτερικά με διπλή ακρίβεια (double precision), ανεξάρτητα από τον τρόπο αναπαράστασης των αποτελεσμάτων στην έξοδο. Για τον καθορισμό της μορφής αναπαράστασης χρησιμοποιείται η εντολή format ακολουθούμενη από την επιθυμητή μορφή (π.χ. short, long, short e, long g κ.τ.λ.). Όταν η format καλείται χωρίς παραμέτρους, τότε η μορφή αναπαράστασης είναι η προκαθορισμένη short, η οποία χρησιμοποιεί 4 δεκαδικά ψηφία. Με την εντολή format long έχουμε εμφάνιση 15 δεκαδικών ψηφίων, ενώ η προσθήκη του χαρακτήρα e δίνει εκθετική μορφή, σαν και αυτήν που είδαμε παραπάνω. Οι εντολές format short g και format long g επιλέγουν την καλύτερη από τις δύο μορφές αναπαράστασης (σταθερής ή κινητής υποδιαστολής) και χρησιμοποιούν 4 ή 15 δεκαδικά ψηφία αντίστοιχα.

Επίσης, μπορούμε να εισάγουμε δεδομένα κατά τη διάρκεια εκτέλεσης ενός προγράμματος με τη χρήση της συνάρτησης input. Η εμφάνιση δεδομένων στην οθόνη, εκτός από το συνηθισμένο τρόπο της αναγραφής μεταβλητών χωρίς ; στο

τέλος, επιτυγχάνεται και με τη συνάρτηση disp. Έτσι, οι παρακάτω γραμμές, εμφανίζουν ένα μήνυμα στο χρήστη, ο οποίος πληκτρολογεί έναν αριθμό, που αποθηκεύεται στη μεταβλητή x, την οποία και εμφανίζουμε στην οθόνη.

ΠΑΡΑΔΕΙΓΜΑΤΑ

(1) ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΕΛΙΚΤΙΚΟΥ ΚΩΔΙΚΟΠΟΙΗΤΗ ΚΑΙ ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗ VITERBI

Με αυτήν την εντολή στο command prompt (αναποδα εισαγωγικά) του Matlab:openExample('comm/CreateConvolutionalCodeExample') εκτελείται το εξής πρόγραμμα:

```
// ***ΚΩΔΙΚΟΠΟΙΗΣΗ

t = poly2trellis([4 3],[4 5 17;7 4 2]); //Ορισμος ενος δενδρου trellis
x = ones(100,1);

code = convenc(x,t);//Κωδικοποιηση ενος διανυσματος με 100 "1"

//***ΑΠΟΚΩΔΙΚΟΠΟΙΗΣΗ

tb = 2; //Ορισμος μηκους ανιχνευσης προς τα πισω

decoded = vitdec(code,t,tb,'trunc','hard'); //Αποκωδικοποιηση

//ΣΥΓΚΡΙΣΗ ΕΙΣΟΔΟΥ-ΕΞΟΔΟΥ (πρεπει να προκυψει 1 - true)

isequal(decoded,ones(100,1))
```

(2) ΥΠΟΛΟΓΙΣΜΟΣ ΕΝΟΣ ΚΩΔΙΚΑ 2/3 ΓΙΑ ΨΗΦΙΑΚΗ ΔΙΑΜΟΡΦΩΣΗ 16-QAM ΚΑΙ ΚΑΝΑΛΙ AWGN

Με αυτήν την εντολή στο command prompt (αναποδα εισαγωγικά) του Matlab:openExample('comm/ViterbiDecodeAWGNSignalExample')

εκτελείται το εξής πρόγραμμα:

```
M = 16; // Αριθμός σημάτων

k = log2(M); //bits per symbol

dataIn = randi([0 1],100000,1); //Είσοδος, είναι πακέτο των 100000 bits

tPoly = poly2trellis([5 4],[23 35 0; 0 5 13]); //Ορισμος του δενδρου trellis

codeRate = 2/3; //Ορισμος ρυθμου κωδικα (2/3)
```

```

codeword = convenc(dataIn,tPoly); //Κωδικοποίηση εισόδου
codewordMat = reshape(codeword,length(codeword)/k,k); // Μετατροπή του
διανύσματος codeword σε πίνακα με k στηλές
txSym = bi2de(codewordMat); //Μετατροπή κάθε σειράς ψηφίων του πίνακα σε
δεκαδικό (από 0 έως 2^M-1)
txSig = qammod(txSym,M); //Δημιουργία των QAM συμβόλων
EbNo = 10; // Ορισμός του Eb/N0
snr = EbNo + 10*log10(k*codeRate); //Υπολογισμός του SNR
rxSig = awgn(txSig,snr,'measured'); // Μετάδοση των συμβόλων QAM
μέσα από το κανάλι
demodSig = qamdemod(rxSig,M); //Αποδιαμόρφωση
demodSigMat = de2bi(demodSig,k); //Μετατροπή δεκαδικών σε ψηφιακές
σειρές
demodSigBinary = demodSigMat(:); //Μετατροπή του πίνακα σε διάνυσμα
traceBack = 16; //Ορισμός μήκους ανιχνεύσης προς τα πίσω
dataOut = vitdec(demodSigBinary,tPoly,traceBack,'cont','hard'); /
/Αποκωδικοποίηση Viterbi
decDelay = 2*traceBack; //Υπολογισμός καθυστέρησης
[numErrors,ber] = biterr(dataIn(1:end-
decDelay),dataOut(decDelay+1:end)) //Υπολογισμός αριθμού λαθών
(numErrors) και BER (ber)
berUncoded = berawgn(EbNo,'qam',M); //Υπολογισμός BER της περίπτωσης
ΧΩΡΙΣ κωδικοποίηση (berUncoded)
berUncoded/ber // Υπολογισμός του ποσες φορές είναι καλύτερο το BER όταν
χρησιμοποιούμε κωδικοποίηση
(3) ΕΞΟΜΟΙΩΣΗΣ ΓΙΑ ΤΟΝ ΚΩΔΙΚΑ BCH (255, 231) ΜΕΣΩ
ΜΟΝΤΕΛΟΥ ΣΤΟ MATLAB ΓΙΑ ΤΟ ΔΙΑΣΤΗΜΑ (2, 5.1)

```

number =

Columns 1 through 7

1007 1001 1003 1007 1004 1001 1010

Columns 8 through 14

1001 1001 1008 1002 1000 1004 1003

Columns 15 through 21

1000 1003 1005 1004 1005 1004 1003

Columns 22 through 28

1002 1004 1002 1004 1002 1001 1003

Columns 29 through 32

1007 1003 1001 1005

ratio =

Columns 1 through 9

0.0436 0.0438 0.0443 0.0415 0.0402 0.0377 0.0361 0.0347
0.0355

Columns 10 through 18

0.0319 0.0295 0.0271 0.0270 0.0248 0.0246 0.0210 0.0215
0.0193

Columns 19 through 27

0.0187 0.0151 0.0142 0.0127 0.0110 0.0104 0.0081 0.0078
0.0067

Columns 28 through 32

0.0064 0.0045 0.0040 0.0029 0.0024

EbNo =

Columns 1 through 9

2.0000 2.1000 2.2000 2.3000 2.4000 2.5000 2.6000 2.7000
2.8000

Columns 10 through 18

2.9000 3.0000 3.1000 3.2000 3.3000 3.4000 3.5000 3.6000
3.7000

Columns 19 through 27

3.8000 3.9000 4.0000 4.1000 4.2000 4.3000 4.4000 4.5000
4.6000

Columns 28 through 32

4.7000 4.8000 4.9000 5.0000 5.1000

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] C. E. Shannon, “A mathematical theory of communication”, Bell System Technical Journal. 27:379-423 (Part I) and 623-656 (Part II), July and October 1948.
- [2] J. Bardeen and W. H. Brattain, “The transistor, a semiconductor triode,” Physical Review, vol. 74, no. 2, pp. 230–231, July 1948.
- [3] J. T. Coffey, and R.M. Goodman, “Any code of which we cannot think is good,” IEEE Transaction on Information Theory, IT-36 (6): 1453–1461, November 1990.
- [4] R. G. Gallager, “Low-density parity-check codes,” IRE Transactions on Information Theory, vol. IT-8, pp. 21-28, January 1962.
- [5] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error correcting coding and decoding; turbo codes,” in Proceedings of IEEE International Conference on Communications, pp. 1064-1070, Geneva, May 1993.
- [6] D. J. C. MacKay and R. M. Neal, “Near Shannon-limit performance of Low-Density ParityCheck codes,” Electronics Letters, vol. 32, no.18, pp. 1645-1646, August 1996.
- [7] B. Vucetic, and J. Yuan, Turbo Codes: Principles and Applications, Kluwer Academic Publishers, 2000.
- [8] M. G. Rekoﬀ, Jr., Analog Computer Programming, C. E. Merril Books, 1967.
- [9] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, A Design Perspective, Second Edition , Prentice Hall, 2003.
- [10] G. M. Shepherd. Neurobiology. Oxford University Press, New York, 1983.
- [11] C. A. Mead. Analog VLSI and Neural Systems. Addison Wesley, Reading, MA, 1989.
- [12] P. Masa, P. Heim, E. Franzi, X. Arréguit, F. Heitger, P. Ruedi, P. Nussbaum, P. Piiloud, and E. Vittoz. “10mW CMOS retina and classifier for handheld, 1000 images/s optical character recognition system.” in Proceedings of the IEEE International Solid-State Circuits Conference, pp. 204–205. San Francisco, CA, February 1999.

- [13] A. Mortara, P. Heim, P. Masa, E. Franzi, P. F. Ruedi, F. Heitger, and J. Baxter. “An optoelectronic 18 b/revolution absolute angle and torque sensor for automotive steering applications.” in Proceedings of the IEEE International Solid-State Circuits Conference, pp. 182–183. San Francisco, CA, February 2000.
- [14] H. T. Siegelmann, *Neural Networks and Analog Computation: Beyond the Turing Limit*, Birkhäuser, Boston, 1999.
- [15] F. Lustenberger, *On the Design of Analog VLSI Iterative Decoders*, Ph.D. dissertation, No 13879, Hartung-Gorre, Konstanz, Series in Signal and Information Processing, vol. 2, ISBN 389649-622-0, ISSN 1616-671X, ETHZ, Zurich, Switzerland, November 2000.
- [16] Jie Dai, *Design Methodology for Analog VLSI Implementations of Error Control Decoders*, Ph.D. dissertation, University of Utah, December 2002.
- [17] S. Hemati, *Iterative Decoding in Analog VLSI*, Ph.D. dissertation, Carleton University, Canada, August 2005.
- [18] J. Hagenauer and M. Winklhofer, “The analog decoder,” in Proceedings of IEEE International Symposium on Information Theory (ISIT98), p.145, 16-21, Boston, USA, August 1998.
- [19] C. Winstead, *Analog Iterative Error Control Decoders*, Ph.D. dissertation, University of Alberta, Edmonton, Canada, Spring 2005.
- [20] A. Carusone, K. Farzan, and D. A. Johns, “Differential signaling with a reduced number of signal paths,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, pp. 294-300, March 2001.
- [21] J. B. Berner and K. S. Andrews, “Deep space network turbo decoder implementation,” in Proceedings of the IEEE Aerospace Conference, pp. 1149-1157, 2001.
- [22] A. J. Blanksby and C. J. Howland. “A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density paritycheck code decoder.” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp: 404–412, March 2002.
- [23] G. D. Forney. “The Viterbi algorithm.” in Proceedings of the IEEE, vol. 61, no. 3, pp: 268–278, March 1973.

- [24] A. Acampora and R. Gilmore. “Analog Viterbi decoding for high speed digital satellite channels.” *IEEE Transactions on Communications*, vol. 26, no. 10, pp: 1463–1470, October 1978.
- [25] A. Demosthenous and J. Taylor. “A 100-Mb/s 2.8-V CMOS current-mode analog Viterbi decoder.” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 7, pp: 904–910, July 2002.
- [26] Hyongsuk Kim, Hongrak Son, T. Roska, and L. O. Chua. “Very high speed Viterbi decoder with circularly connected analog CNN cell array.” in *Proceedings of the International Symposium on Circuits and Systems (ISCAS '04)*, 3:III, pp:97–100, May 2004.
- [27] T. W. Matthews and R. R. Spencer. “An integrated analog CMOS Viterbi detector for digital magnetic recording.” *IEEE Journal of Solid-State Circuits*, vol. 28, no. 12, pp: 1294–1302, December 1993.
- [28] M.H. Shakiba, D.A. Johns, and K.W. Martin. “An integrated 200MHz 3.3V BiCMOS class-IV partial-response analog Viterbi decoder.” *IEEE Transactions on Circuits and Systems II*, vol.1, pp: 61–75, January 1998.
- [29] Xiao-An Wang and S. B. Wicker. “An artificial neural net Viterbi decoder.” *IEEE Transactions on Communications*, vol. 44, no. 2, pp: 165–171, February 1996.
- [30] F. Lustenberger, M. Helfenstein, G. S. Moschytz, H. A. Loeliger, and F. Tarkoy. “All analog decoder for (18,9,5) tail-biting trellis code.” in *Proceedings of the European Solid-State Circuits Conference (ESSCIRC)*, Duisberg, Germany, pp: 362–365, Sept.1999.
- [31] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. “Optimal decoding of linear codes for minimizing symbol error rate.” *IEEE Transactions on Information Theory*, pp: 284–287, March 1974.
- [32] L. D. Rudolph, C. R. P. Hartmann, T.Y. Hwang, and N. Q. Duc. “Algebraic analog decoding of linear binary codes.” *IEEE Transactions on Information Theory*, vol. 25 no. 4, pp: 430–440, July 1979.
- [33] Xiao-An Wang and S. B. Wicker. “An artificial neural net Viterbi decoder.” *IEEE Transactions on Communications*, vol. 44, no.2, pp: 165–171, February 1996.

- [34] N. Wiberg, Codes and decoding on general graphs, Ph.D. dissertation 440, Linkoping Studies in Science and Technology, University of Linkoping, Linkoping, Sweden, 1996.
- [35] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, February 2001.
- [36] H. A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkoy. “Probability propagation and decoding in analog VLSI.” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp: 837–843, February 2001.
- [37] G. D. Forney. “Codes on graphs: normal realizations.” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp: 520–548, February 2001.
- [38] M. Moerz, T. Gabara, R. Yan, and J. Hagenauer. “An analog 0.25 μ m BiCMOS tailbiting MAP decoder.” in *International Solid State Circuits Conference*, pp: 356–357, February 2000.
- [39] C. Winstead, J. Die, W.J. Kim, S. Little, Y. -B. Kim, C. J. Myers, and C. Schlegel. “Analog MAP decoder for (8,4) Hamming code in subthreshold CMOS.” in *Advanced Research in VLSI*, pp: 132–147, March 2001.
- [40] C. Winstead, J. Die, S. Yu, R. Harrison, C. J. Myers, and C. Schlegel. “Analog MAP decoder for (8,4) Hamming code in subthreshold CMOS.” in *Proceedings of the International Symposium on Information Theory*, pp: 330, June 2001.
- [41] C. Winstead, J. Dai, S. Yu, R. Harrison, C. J. Myers, and C. Schlegel. “CMOS analog decoder for (8,4) Hamming code.” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp: 122–131, January 2004.
- [42] A. Xotta, D. Vogrig, A. Gerosa, A. Neviani, A. Graell-Amat, G. Montorsi, M. Bruccoleri, and G. Betti. “An all-analog CMOS implementation of a Turbo decoder for hard-disk drive read channels.” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '02)*, pp: 69–72, Scottsdale, USA, May 2002.
- [43] Alexandre Graell i Amat, Sergio Benedetto, Guido Montorsi, Daniele Vogrig, Andrea Neviani, and Andrea Gerosa. “An analog Turbo decoder for the UMTS

standard.” in Proceedings of the International Symposium on Information Theory, 2004.

[44] D. Vogrig, A. Gerosa, A. Neviani, A. Graef i Amat, G. Montorsi, and S. Benedetto. “A 0.35- μm CMOS Analog Turbo Decoder for the 40-bit Rate 1/3 UMTS Channel Code.” IEEE Journal of Solid-State Circuits, vol. 40, no. 3, pp: 753-762, March 2005.

[45] M. Perenzoni, A. Gerosa, and A. Neviani. “Analog CMOS implementation of Gallager’s iterative decoding algorithm applied to a block Turbo code.” in the Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’03), vol. V, pp: V – 813–816, May 2003.

[46] V. C. Gaudet and P. G. Gulak. “A 13.3-Mb/s 0.35- μm CMOS analog turbo decoder IC with a configurable interleaver.” IEEE Journal of Solid-State Circuits, vol. 38, no. 11, pp: 2010–2015, November 2003.

[47] C. Winstead, V. C. Gaudet, and C. Schlegel. “A CMOS Analog (16, 11)₂ Turbo Product Decoder.” In 3rd Analog Decoding Workshop (ADW ’04), Alberta, Canada, July 2004.

[48] M. Moerz, A. Schaefer, and E. Offer. “Analog decoding of high rate tailbiting codes using the dual trellis.” in Proceedings of the International Symposium on Information Theory, pp: 331, June 2001.

[49] A. Schaefer, M. Moerz, J. Hagenauer, A. Sridharan, and D. J. Costello Jr. “Analog rotating ring decoder for an LDPC convolutional code.” In Proceedings of the International Information Theory Workshop, vol. 31, pp: 226–229, April 2003.

[50] A. Mondragon-Torres and E. Sanchez-Sinencio. “Floating-gate analog implementation of the additive soft-input soft-output decoding algorithm.” in Proceedings of the International Symposium on Circuits and Systems, pp: 89–92, May 2002.

[51] A. F. Mondragon-Torres, E. Sanchez-Sinencio, and K. R. Narayanan. “Floating-gate analog implementation of the additive soft-input soft-output decoding algorithm.” IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 50, no. 10, pp: 1256–1269, October 2003.

- [52] Shih-Chii Liu, Jörg Kramer, Giacomo Indiveri, Tobias Delbrück, and Rodney Douglas. Analog VLSI: Circuits and Principles. MIT Press, 2002.
- [53] A. H. Banihashemi and S. Hemati. “Decoding in optics.” in Proceedings of the International Symposium on Information Theory, pp: 231, June 2002.
- [54] A. H. Banihashemi and S. Hemati. “Analog min-log APP decoder.” In Proceedings of the International Symposium on Information Theory, June 2003.
- [55] S. Hemati, A. H. Banihashemi, and C. Plett. “An 80-Mb/s 0.18- μ m CMOS analog min-sum iterative decoder for a (32,8,10) LDPC code,” in Proceedings of the IEEE Custom Integrated Circuit Conference (CICC '05), September 2005.
- [56] Matthias Frey, Hans-Andrea Loeliger, Felix Lustenberger, Patrick Merkli, and Patrik Strebler. “Analog-decoder experiments with subthreshold CMOS softgates.” in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'03), pp: 85–88, Bangkok, Thailand, May 2003.
- [57] J. Hagenauer, M. Moerz, and A. Schaefer. “Analog decoders and receivers for high speed applications.” In Proceedings of the International Zurich Seminar on Broadband Communication, pp: 3–1–3–8, 2002.
- [58] W. Huang, V. Ijure, G. Rose, Y. Zhang, M. Stan. “Analog Turbo Decoder Implemented in SiGe BiCMOS Technology,” in the 40th Design Automation Conference Student Design Contest (DAC '03), Anaheim, California, June 2003.
- [59] W. Huang, V. Ijure, G. Rose, Y. Zhang, and M. Stan. Analog Turbo decoder implemented in SiGe BiCMOS technology, July 2003. Available at <http://www.ece.virginia.edu/hplp/turbo.html>.
- [60] M. Arzel, C. Lahuev, E. Seguin, D. Gnaedig, and M. Jezequel. “Analog Slice Turbo Decoding,” in Proceedings of the International Symposium on Circuits and Systems, pp: 332-335, Kobe, Japan, May 2005.
- [61] F. Gioulekas, M. Birbas, A. Birbas, and G. Bilionis, “A High-Speed Analog Turbo Decoder with Low-Energy Consumption,” in Proceedings of the IEEE International Symposium on Communications and Information Technologies 2004 (ISCIT 2004), pp: 33-38, Sapporo, Japan, October 26-29, 2004.

- [62] F. Gioulekas, M. Birbas, A. Birbas, and G. Bilonis. “A High-Speed Analog Turbo Decoder,” Transactions on Electrical Engineering, Electronics, and Communications, vol. 3, no.2 pp. 7785, August 2005.
- [63] F. Gioulekas, M. Birbas, G. Bilonis, and A. Birbas. “A High-Speed Analog Trellis Decoder with Low-Energy Consumption,” in Proceedings of the 4th International Symposium on Turbo Codes in connection with the 6th International ITG-Conference on Source and Channel Coding, Munich, Germany, April 03-07, 2006.
- [64] F. Gioulekas, M. Birbas, A. Birbas, G. Bilonis, K. Efstathiou, and M. Papamichail. “A HighSpeed SiGe BiCMOS Analog MAP Decoder,” in Proceedings of the 5th Analog Decoding Workshop, Torino, Italy, June05-06, 2006.
- [65] S. Benedetto, D. Dinsalar, G. Montorsi, and F. Pollara. “Soft-Input Soft-Output Module for the construction and distributed iterative decoding of code networks,” European Transactions on Telecommunications, vol. 9, pp: 155-172, March 1998.
- [66] Y. Tsividis. Operation and Modeling of the MOS Transistor.2nd edition, McGraw-Hill, April 1999.
- [67] C. Berrou, P. Combelles, P. Penard, and B. Talibart. “An IC for Turbocodes Encoding and Decoding,”in Proceedings of the IEEE International Solid-State Circuits Conference Digest Technical Papers (ISSCC '95), pp: 90-91, San Francisco, USA, February 1995.
- [68] TMS320C55XTM DSP, Texas Instruments. Available at <http://www.ti.com>.
- [69] S. Hong, J. Yi, and W. E. Stark. “VLSI design and implementation of low complexity adaptive Turbo-code encoder and decoder for wireless mobile communication applications.” in Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS'98), pp: 233–242, October 1998.
- [70] M. Bickerstaff, D. Garrett, T. Prokop, C. Thomas, B. Widdup, G. Zhou, C. Nicol, and R.-H. Yan. “A unified Turbo / Viterbi channel decoder for 3GPP mobile wireless in 0.18 μ m CMOS.” in Proceedings of the IEEE International Solid State Circuits Conference (ISSCC'02), pp: 90– 91, February 2002.

- [71] J. Kaza and C. Chakrabarti. “Design and implementation of low energy Turbo decoders.” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 9, pp: 968–977, September 2004.
- [72] I. Al-Mohondes and M. Elmasry. “A low-power 5Mb/s Turbo decoder for third-generation wireless terminals.” in Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE’04), vol. 4, pp: 2387–2390, 2004.
- [73] J. G. Proakis. Digital Communications. 4th edition, McGraw-Hill, 2001.
- [74] I. S. Reed and G. Solomon, “Polynomial Codes over Certain Finite Fields.” SIAM J., vol. 8, pp: 300-304, June 1960.
- [75] R. C. Bose and D. K. Ray-Chaudhuri, “On a Class of Error Correcting Binary Group Codes,” Information Control, vol. 3, pp: 68-79, March 1960.
- [76] P. Elias. “Error-free Coding.” IRE Transactions on Information Theory, vol. IT-4, pp: 29-37, 1954.
- [77] G. D. Forney, Jr., “Convolutional Codes I: Algebraic Structure.” IEEE Transactions on Information Theory, vol. IT-16, no. 6, pp: 720-738, Nov. 1970, and vol. IT-17, no. 3, p: 360, May 1971.
- [78] R. Hamming, “Error Detecting and Error Correcting Codes,” Bell Syst. Tech. J., vol. 29, pp: 147160, 1950.
- [79] S. ten Brink, “Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes,” IEEE Transactions on Communications, vol. 49 no. 10, pp: 1727–1737, October 2001.
- [80] Sae-Young Chung, G. David Forney Jr., Thomas J. Richardson, and Rudiger Urbanke, “On the design of Low-Density Parity-Check codes within 0.0045dB of the Shannon Limit,” IEEE Communications Letters, vol.5, no. 2, pp: 58-60, February 2001.
- [81] H. Jin, A. Khandekar, and R. McEliece, “Irregular repeat-accumulate codes,” in Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics, pp: 1–8, Brest, France, September 2000.
- [82] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” IEEE Transactions on Information Theory, vol. IT-13, pp: 260–269, 1967.

- [83] R. M. Tanner, "A recursive approach to low complexity codes," IEEE Transactions on Information Theory, vol. 27, no. 5, pp: 533–547, September 1981.
- [84] B. J. Frey, Bayesian Networks For Pattern Classification, Data Compression, and Channel Coding, Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada, 1997.
- [85] R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics. Reading, MA: AddisonWesley, 1989.
- [86] P.R. Gray and R.G. Meyer, Analysis and Design of Analog Integrated Circuits, 3rd edition, John Wiley & Sons, 1993.
- [87] Y. Tsividis, Operation and Modeling of the MOS Transistor, 2nd edition, McGraw-Hill, Apr. 1999.
- [88] B. Gilbert. "Translinear circuits: A proposed classification." Electronics Letters, vol. 11, no. 1, pp. 14–16, January 1975.
- [89] B. Gilbert. "Translinear circuits: An historical overview." Analog Integrated Circuits and Signal Processing, vol. 9, no. 2, pp. 95–118, March 1996. Special Issue: Translinear Circuits.
- [90] B. Gilbert. "A precise four-quadrant multiplier with subnanosecond response." IEEE Journal of Solid-State Circuits, vol. 3, pp. 365–373, 1968.
- [91] B. Gilbert. "A monolithic 16-channel analog array normalizer." IEEE Journal of Solid-State Circuits, vol. 19, pp. 956–963, 1984.
- [92] K. Kimura. "Some circuit design techniques using two cross- coupled pairs." IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications, vol. 41, no. 5, pp. 411–423, May 1994.
- [93] C. F. Chan, H. Ling, and O. Choy. "A one volt four-quadrant analog current mode multiplier cell." IEEE Journal of Solid-State Circuits, vol. 30, no. 9, pp. 1018–1019, Sept. 1995.
- [94] G. Colli and F. Montecchi. "Low voltage low power CMOS four-quadrant analog multiplier for neural network applications." In Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 1, pp. 496–499. 1996.

- [95] W. Gai, H. Chen, and E. Seevinck. “Quadratic-translinear CMOS multiplier-divider circuit.” *Electronics Letters*, vol. 33, no. 10, pp. 860–861, May 1997.
- [96] R. J. Wiegerink. “A CMOS four-quadrant analog current multiplier.” In *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 2244–2247. 1991.
- [97] K. Kimura. “A bipolar low-voltage quarter-square multiplier with a resistive-input based on the bias offset technique.” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 2, pp. 258–266, Feb. 1997.
- [98] H. R. Mehrvarz and C. Y. Kwok. “A novel multi-input floating-gate MOS four-quadrant analog multiplier.” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 8, pp. 1123–1131, Aug. 1996.
- [99] J. Ramirez-Angulo. “A 0.75V BiCMOS four-quadrant analog multiplier with rail-rail input signal-swing.” In *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 242–245. 1996.
- [100] M. O’Halloran and R. Sarpeshkar, “A 10-nW 12-bit accurate analog storage cell with 10-aA leakage,” *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 1985–1996, Nov. 2004.
- [101] S. S. Pietrobon, "Implementation and performance of a turbo/MAP decoder," *International Journal of Satellite Communications*, vol. 16, pp. 23-46, Jan.-Feb. 1998.
- [102] Small World Communications, <http://www.sworld.com.au>.
- [103] J. Steensma, C. Dick, “FPGA Implementations of a 3GPP Turbo codec,” *Proceedings of 35th IEEE Asilomar conference on Signal, Systems and Computers (Asilomar01)*, Pacific Grove, CA, USA, pp. 61-65, Nov. 2001.
- [104] Comtech AHA Corporation, <http://www.aha.com>.
- [105] V. Boonsobhak, A. Worapishet, and J.B. Hughes, “Reduced-Kickback Current Comparator for High-Speed Switched-Current Pipeline Analogue-to-Digital Converters,” *IEE Electronics Letters*, vol. 39, No.1, Jan. 2003.
- [106] M. Ferrari, S. Belline “Importance sampling simulation of turbo product codes.” *ICC2001, The IEEE International Conference on Communications 9*, pp: 2773–2777, 2001.