ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΑΛΑΜΑΤΑΣ
ΠΑΡΑΡΤΗΜΑ ΣΠΑΡΤΗΣ

ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΚΑΤΑΣΚΕΥΗ ΚΑΙ ΜΕΛΕΤΗ ΚΩΔΙΚΩΝ ΕΠΑΝΑΛΗΨΗΣ ΚΑΙ ΣΥΣΣΩΡΕΥΣΗΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΧΡΗΣΤΟΣ ΨΑΡΡΑΣ

Α.Μ: 2007063

ΕΙΣΗΓΗΤΗΣ: ΒΑΣΙΛΕΙΟΣ ΜΠΟΖΑΝΤΖΗΣ

-2011-

Πίνακας περιεχομένων

ΚΕΦΑΛΑΙΟ 1 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

1.1Εισαγωγή

Η παρούσα διπλωματική εργασία πραγματεύεται τη μελέτη των αλγορίθμων κωδικοποίησης Turbo και LDPC, το σχεδιασμό, και την υλοποίηση τους. Τα τεχνολογικά επιτεύγματα του 20ου αιώνα συνέβαλαν σημαντική στη βελτίωση της ποιότητας ζωής του ανθρώπου καθώς διευκόλυναν την εξυπηρέτηση των καθημερινών αναγκών του. Ένας από τους σημαντικότερους παράγοντες της καθημερινότητας είναι οι κάθε μορφής επικοινωνίες, οι οποίες έχουν παρουσιάσει εξαιρετικό πρόοδο τις τελευταίες δεκαετίες.

Στα ψηφιακά συστήματα, η πληροφορία κωδικοποιείται σε ακολουθίες από 0 και 1, που αντιστοιχούν στις δύο δυνατές καταστάσεις των τρανζίστορ, on - off, τα οποία λειτουργούν ως διακόπτες. Το πλεονέκτημα αυτό έχει επιφέρει σημαντικές αλλαγές στον τρόπο επεξεργασίας της πληροφορίας.

Η πληροφορία στη φύση παρουσιάζεται αποκλειστικά σε αναλογικό μορφή, καθώς ο άνθρωπος μόνο αναλογικά σήματα μπορεί να αντιληφθεί. Έτσι λοιπόν, τα αναλογικά δεδομένα πρέπει να ψηφιοποιηθούν, να μετατραπούν δηλαδή σε ακολουθίες από 0 και 1, ώστε ο δέκτης να μην είναι υποχρεωμένος να κάνει μία εκτίμηση των άπειρων τιμών ενός αναλογικού σήματος, αλλά απλά να πάρει μία απόφαση μεταξύ των δύο διακριτών τιμών για κάθε σήμα, 0 ή 1. Η διαδικασία αυτή καθιστά τα ψηφιακά σήματα πιο αξιόπιστα για τη μετάδοση πληροφορίας σε ένα ενθόρυβο περιβάλλον, καθώς μπορούν να ανιχνεύονται σχεδόν τέλεια, όταν το επίπεδο του θορύβου δεν είναι ιδιαίτερα υψηλό, πράγμα που

επιτρέπει την ανάκτηση των ψηφιακών στοιχείων, και μέσω των τεχνικών διόρθωσης λαθών είναι δυνατό η διόρθωση σφαλμάτων που συμβαίνουν κατά τη μετάδοση.

Η ψηφιακή πληροφορία μπορεί να κωδικοποιηθεί με τέτοιο τρόπο ώστε να εισάγονται σε αυτόν επιπρόσθετα δυαδικά ψηφία που δε μεταφέρουν πληροφορία και ονομάζονται πλεονασμός (redundancy). Τα επιπρόσθετα ψηφία επιτρέπουν στον δέκτη να αναγνωρίσει τα σφάλματα που τυχόν προέκυψαν κατά τη μετάδοση.

Η τεχνική αυτή ονομάζεται **Κωδικοποίηση Ελέγχου Σφάλματος (Error Control Coding).** Τις τελευταίες δεκαετίες το ενδιαφέρον των επιστημόνων της **Θεωρίας Κωδίκων** έχει επικεντρωθεί στην κατασκευή κωδίκων διόρθωσης σφαλμάτων πολύ καλά δομημένων, οι οποίοι διαθέτουν μεγάλη ελάχιστη απόσταση, **dmin**. Η υψηλή ποιότητα των κωδίκων ως προς τη δομή τους, καθιστά αντιμετωπίσιμη την πολυπλοκότητα της αποκωδικοποίησης, ενώ παράλληλα η μεγάλη ελάχιστη απόσταση φέρεται να εξασφαλίζει την υψηλή απόδοση του κώδικα. Ωστόσο, η προσέγγιση αυτή δεν θα μπορούσε να μην έχει και ορισμένα μειονεκτήματα. Αρχικά, για να είναι ένα σχήμα κωδικοποίησης αξιόπιστο, η επιλογή των κωδίκων θα πρέπει να γίνει τυχαία. Το γεγονός αυτό όμως, έρχεται σε αντίθεση με το στόχο της θεωρίας κωδίκων, την κατασκευή δηλαδή, πολύ καλά δομημένων κωδίκων, οι οποίοι παράλληλα χαρακτηρίζονται από ένα απλό σχήμα αποκωδικοποίησης. Επίσης, συγκρινόμενη με τη χωρητικότητα του διαύλου (channel capacity), η ελάχιστη απόσταση, σε πρακτικό επίπεδο, αποτελεί μία μικρότερου ενδιαφέροντος παράμετρο σχετικά με την απόδοση του κώδικα. Από το 1993 και μετά, οι νεότερες τεχνικές κωδικοποίησης επέτρεψαν την κατασκευή κωδίκων των οποίων η απόδοση σε δίαυλο Προσθετικού Λευκού Γκαουσσιανού Θορύβου **(AWGN)** προσέγγιζε το όριο του Shannon με απόκλιση 1dB. Οι μέθοδοι αυτές, όπως για παράδειγμα οι Turbo και οι

LDPC κώδικες, χρησιμοποιούν μία εντελώς διαφορετική φιλοσοφία βασισμένη στα επαναληπτικά σχήματα κωδικοποίησης.

1.2 Χωρητικότητα καναλιού μετάδοσης

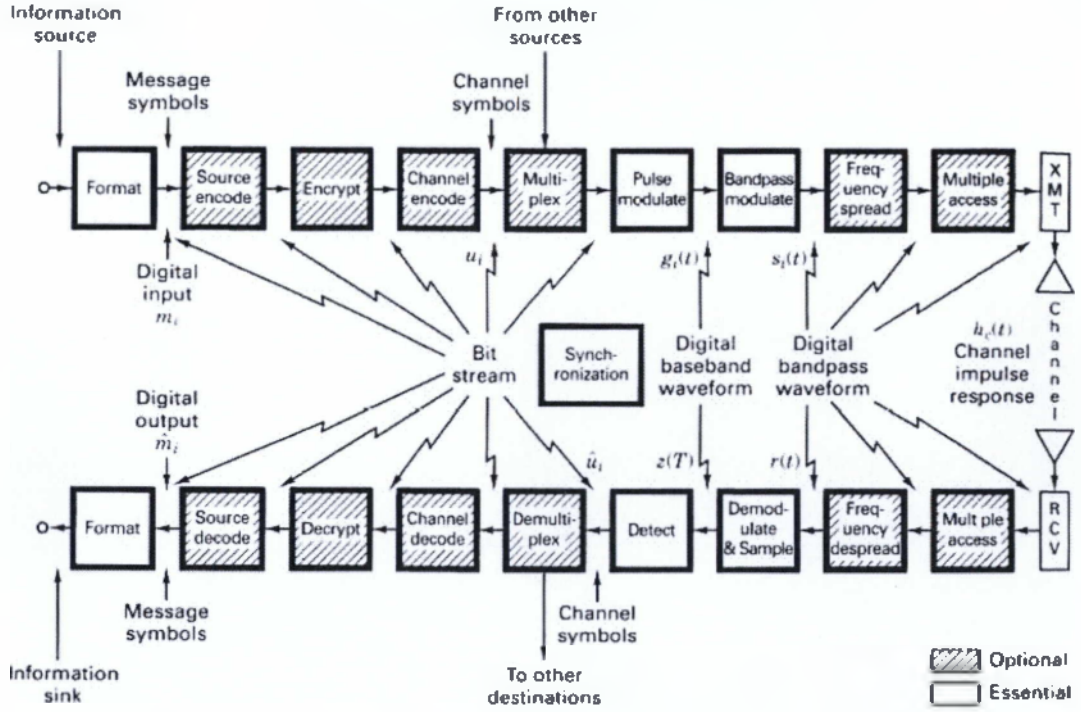
Το 1948 είναι μια χρονιά ορόσημο στη διαμόρφωση του σύγχρονου τεχνολογικού κόσμου, άποψη που δικαιολογείται από δύο επινοήσεις που ξεπήδησαν από τα εργαστήρια της εταιρείας Bell: αφ' ενός έχουμε την εφεύρεση του transistor από τους W. Shockley, J. Bardeen και W. Brattain η οποία θεωρείται η σημαντικότερη του $20^{ου}$ αιώνα, και αφ' ετέρου τη γέννηση της επιστήμης της Θεωρίας Πληροφορίας και Κωδικοποίησης από τον Claude E. Shannon. Οι δύο αυτές κατακτήσεις περπάτησαν μαζί, με την άνθιση και τις εφαρμογές της δεύτερης να εξαρτώνται εν πολλοίς από την εξέλιξη της πρώτης. Οι καταπληκτικές επιδόσεις των σημερινών τηλεπικοινωνιακών συστημάτων, δηλαδή η γρήγορη κι αξιόπιστη μετάδοση της πληροφορίας, έχουν τις ρίζες τους στις εν λόγω καινοτομίες.

Ωστόσο, όσο καταπληκτικές κι αν κρίνονται, είναι απλά καταπληκτικότερες των προηγούμενων, που σημαίνει ότι δεν παύουν να εξαιρούνται από τις διεργασίες που ονομάζουμε «πρόοδο», δηλαδή την κίνηση προς ένα επιθυμητό τέλος. Ο Shannon στην ιστορική εργασία του «A Mathematical Theory of Communication» ασχολήθηκε με τον προσδιορισμό αυτού του επιθυμητού τέλους, του θεωρητικών ορίων, δηλαδή, των τηλεπικοινωνιακών συστημάτων. Για το σκοπό αυτό εισήγαγε και θεμελίωσε έννοιες όπως εντροπία πηγής πληροφορίας (information source entropy), αμοιβαία πληροφορία καναλιού (mutual information) και τέλος, χωρητικότητα καναλιού (channel capacity).

Σε ένα τηλεπικοινωνιακό σύστημα όπως το προσδιόρισε ο Shannon

(Σχήμα 1), εντροπία της πηγής της πληροφορίας X ορίστηκε το ακόλουθο μέγεθος, όπου N το πλήθος των συμβόλων που μπορεί να εκπέμψει η πηγή X (πλήθος στοιχείων αλφαβήτου πηγής πληροφορίας), και p_i είναι η πιθανότητα εκπομπής του συμβόλου i

$$H(X)=\sum_{i=1}^{N} p_i \quad * \log(p_i)$$



Σχήμα 1. Ένα ψηφιακό τηλεπικοινωνιακό σύστημα

Εφόσον η X μοντελοποιείται ως τυχαία μεταβλητή, διότι δεν είναι γνωστό εκ των προτέρων το περιεχόμενό της, η φυσική ερμηνεία της εντροπίας της είναι το ποσό της μέσης αβεβαιότητας που καλύπτει κάθε έξοδος της X σε έναν παρατηρητή αυτών των εξόδων, το ποσό της μέσης πληροφορίας, δηλαδή, που του παρέχει. Αλλιώς ειπωμένο, είναι

ένα μέτρο του τυχαίο και απρόβλεπτου των εξόδων της X:
Αμοιβαία πληροφορία του διαύλου ορίστηκε το μέγεθος απρόβλεπτου των εξόδων της X, όπου Y είναι η αλλοιωμένη από τον θόρυβο ληφθείσα ακολουθία πληροφορίας (τυχαία μεταβλητή επίσης), και H(X|Y) είναι η υπό συνθήκη εντροπία της X δεδομένης της Y, και παριστάνει την μέση εναπομείνασα αβεβαιότητα για την X μετά την παρατήρηση της Y:

$$\mathrm{I}(\mathrm{X};\mathrm{Y}) = \mathrm{H}(\mathrm{X}) - \mathrm{H}(\mathrm{X}|\mathrm{Y}) \quad (1)$$

Χωρητικότητα διαύλου ορίστηκε το μέγεθος:

$$C = \max_{p(x)} I\,(X;Y) \quad (2)$$

και ταυτίζεται με τη μέγιστη αμοιβαία πληροφορία που μπορεί να επιτευχθεί με κατάλληλη επιλογή της κατανομής πιθανότητας των εξόδων της X, p(X). Μεταφράζοντας τον μαθηματικό ορισμό της, μπορούμε να πούμε ότι χωρητικότητα είναι η μέγιστη μέση πληροφορία που μπορεί να διοχετευτεί μέσα από ένα κανάλι για κάθε εκπομπή της X. Η μονάδα μέτρησής της είναι bits/σύμβολο, ή bits/διάσταση για πιο εύληπτες συγκρίσεις μεταξύ των σχημάτων διαμόρφωσης.

Σύμφωνα με το θεώρημα κωδικοποίησης ενθόρυβου καναλιού (noisy channel coding theorem) του Shannon, όταν ο ρυθμός μετάδοσης πληροφορίας είναι μικρότερος από τη χωρητικότητα είναι δυνατή η αξιόπιστη επικοινωνία, δηλαδή η προστασία της πληροφορίας από τον θόρυβο οσοδήποτε καλά επιθυμούμε.
Όταν, όμως, υπερβληθεί αυτό το όριο δεν μπορούμε κατ' ουδένα τρόπο να ελέγξουμε τη συχνότητα εμφάνισης σφαλμάτων. Η χωρητικότητα, επομένως, σα μια νοητά χαραγμένη γραμμή οριοθετεί δύο περιοχές: σ' αυτήν που βρίσκεται από κάτω, μας δίνεται η δυνατότητα

καθορισμού μιας επιθυμητής πιθανότητας σφάλματος οσοδήποτε κοντά στο 0, με την οποία θα συμμορφώνεται η επικοινωνία μας. Στην δε άνω της χωρητικότητας περιοχή τα λάθη γίνονται ανεξέλεγκτα, κι η πιθανότητα σφάλματος απομακρύνεται από το 0. Θα επανέλθουμε σ' αυτό το θεώρημα αργότερα. Η σχέση που δίνει τη χωρητικότητα για διακριτά κανάλια χωρίς μνήμη (Discrete Memoryless Channels) διαβρωμένα από AWGN και με περιορισμό ισχύος εισόδου είναι:

$$C = \frac{1}{2} log_2(1 + \frac{S}{N}) \quad \text{[bits/διάσταση]} \ (3)$$

όπου S είναι η ισχύς του σήματος εισόδου και N του θορύβου. Μέσα από αυτήν τη σχέση ορίζεται η μέγιστη πληροφορία που μπορεί να αποσταλεί για δεδομένο SNR. Ωστόσο, η παραπάνω έκφραση της χωρητικότητας είναι ένα εντελώς θεωρητικό όριο, διότι ισχύει μόνο όταν τα στατιστικά χαρακτηριστικά της πηγής πληροφορίας είναι αυτά μίας λευκής και γκαουσιανής διαδικασίας, και το αλφάβητό της είναι συνεχές. Όπως θα δούμε παρακάτω, για τις πρακτικές πηγές πληροφορίας η έκφραση της χωρητικότητας είναι διαφορετική Η καμπύλη αυτής της χωρητικότητας φαίνεται στο Σχήμα 2, για τιμές του SNR -20dB έως 30dB. Έχει σημειωθεί και η περιοχή αξιόπιστης επικοινωνίας, όπου για τον ρυθμό μετάδοσης πληροφορίας R ισχύει R < C. Επίσης, φαίνεται και η περιοχή μη αξιόπιστης επικοινωνίας, R > C.

Όταν το κανάλι είναι συνεχούς (continuous) χρόνου, με περιορισμένο εύρος ζώνης W και περιορισμό ισχύος S, μπορούμε δειγματοληπτώντας την έξοδό του με ρυθμό 2W να το μετατρέψουμε σε διακριτού χρόνου, και χρησιμοποιώντας την προηγούμενη σχέση να καταλήξουμε στον δημοφιλή τύπο του Shannon για τη χωρητικότητα καναλιού συνεχούς χρόνου:

$$C = W \cdot log_2 (1 + \frac{S}{N}) \ \text{[bits/s]} \ (4)$$

Για αξιόπιστη μετάδοση ρυθμού R < C, ο παραπάνω τύπος γίνεται:

$$R < W \cdot \log_2(1+\frac{S}{N}) \quad (5)$$

Διαιρώντας τον ρυθμό με το εύρος ζώνης, προκύπτει ένα καινούριο μέγεθος το οποίο ονομάζεται φασματική αποδοτικότητα (spectral/bandwidth efficiency):

$$n = \frac{R}{W} < log_2(1+\frac{S}{N}) \text{ [bits/s/Hz]} \quad (6)$$

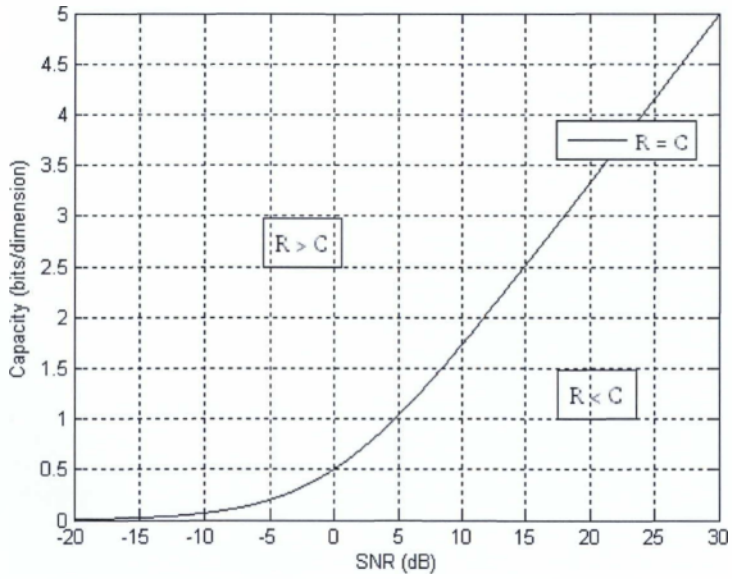
και είναι ο κανονικοποιημένος στη μονάδα της συχνότητας ρυθμός μετάδοσης πληροφορίας. Χρησιμοποιείται εκτενώς για τη σύγκριση και αξιολόγηση τηλεπικοινωνιακών συστημάτων, καθώς η ανεξαρτησία της από το εκάστοτε διαθέσιμο εύρος ζώνης πληροφορεί εύγλωττα για την ικανότητά τους ως προς την ταχύτητα μετάδοσης της πληροφορίας.

Μέσα από αυτό το θεώρημα εισάγεται η έννοια της κωδικοποίησης καναλιού (channel coding), η συνοδεία, δηλαδή, της πληροφορίας αυτής καθαυτή από πλεονασματική (redundant) «πληροφορία». Το πλεόνασμα αυτό δεν έχει κανένα πληροφοριακό φορτίο, κι ο ρόλος του είναι η απόδοση στον δέκτη μιας ικανότητας διόρθωσης των λαθών που συνέβησαν στην αποσταλθείσα ακολουθία πληροφορίας, λόγω της θορυβώδους φύσης του μέσου μετάδοσης.

Αυτό το θεώρημα είναι η βάση για τους κώδικες διόρθωσης λαθών (error correcting codes), οι οποίοι μας επιτρέπουν σήμερα την επανάκτηση της αρχικής ακολουθίας πληροφορίας ακόμα κι όταν φτάνει στον δέκτη με ισχύ χαμηλότερη από αυτήν του θορύβου.

Σήμερα, 60 χρόνια μετά την εργασία του Shannon, η Θεωρία Πληροφορίας και Κωδικοποίησης έχει να προτείνει ως τις καλύτερες λύσεις για αξιόπιστη επικοινωνία πολύ κοντά στη χωρητικότητα τους κώδικες τούρμπο (Turbo Codes) και τους LDPC (Low Density Parity Check codes). Ο «διαγωνισμός» για το ποιος από τους δύο είναι

καλύτερος διαρκεί ακόμα, με τους πρώτους να έχουν ένα προβάδισμα, τόσο από άποψη επίδοσης όσο και πολυπλοκότητας, στους χαμηλούς κωδικούς ρυθμούς (code rates), π.χ. 1/2, 1/3,1/4, 1/6, ενώ οι δεύτεροι προτιμώνται για υψηλότερους ρυθμούς, 2/3, 4/5 ή 7/8. Ωστόσο, στις περιπτώσεις που προέχει η χαμηλή πολυπλοκότητα του αποκωδικοποιητή, οι κλασσικοί συνελικτικοί κώδικες με αποκωδικοποίηση Viterbi παραμένουν αξεπέραστοι.



Σχήμα 2. Περιοχή αξιόπιστης επικοινωνίας, όπου για τον ρυθμό μετάδοσης πληροφορίας R ισχύει R < C για πηγή πληροφορίας White-Gaussian

1.3 Βασικές Έννοιες

1.3.1 Κώδικες δομής (Block Codes)

Σε ένα κώδικα δομής, ο κωδικοποιητής δέχεται στην είσοδο του μια σταθερούς μήκους ακολουθία πληροφοριακών ψηφίων, το μήνυμα (message), που αποτελείται από k bits. Στη συνέχεια, μετατρέπει την πληροφορία αυτό σε μια ακολουθία n bits και σχηματίζει την κωδική λέξη (codeword). Όπως προαναφέρθηκε, το σημαντικό χαρακτηριστικό των κωδικοποιητών δομής είναι ότι αποτελούν διατάξεις χωρίς μνήμη διότι δε χρησιμοποιούν ψηφία από προηγούμενα μπλοκ. Ένας κώδικας δομής χαρακτηρίζεται από τις παραμέτρους (n, k) και απεικονίζει κάθε k- σύμβολο ενός διανυσματικού χώρου F^k πάνω από το πεδίο Galois F=GF(2) σε ένα μεγαλύτερο διανυσματικό χώρο δεδομένων F^n αποτελούμενη από n-σύμβολα και ονομάζεται κωδική λέξη, ακλουθώντας έναν γενικό αλγεβρικό κανόνα, οποίος περιγράφεται από γεννήτορες πίνακες (generator matrix) G ή πολυώνυμα.

Σε έναν (n,k) κώδικα δόμος υπάρχουν 2^K ξεχωριστή μηνύματα. Αφού λοιπόν σε κάθε μήνυμα αντιστοιχεί μια και μόνο κωδική λέξη, θα υπάρχουν και 2^K ξεχωριστές κώδικες λέξεις, καθεμιά από τις οποίες έχει μήκος n. Ο ρυθμός κώδικα (code rate), R = k/n, καθορίζει και την ποσότητα του πλεονασμού. Ένας κώδικας δομής είναι γραμμικός (linear) αν το άθροισμα δύο κωδικών λέξεων αποτελεί μία άλλη κωδική λέξη και αν ο κώδικας περιέχει και τη μηδενική κωδική λέξη.

Ένας κώδικας δομής παρήγεται από ένα σύνολο k γραμμικών

ανεξάρτητων n-διαστατών διανυσμάτων g0, g1 , ... , gκ-1. Οι κώδικες λέξεις αποτελούν γραμμικό συνδυασμό αυτών των k (n-διαστατών) διανυσμάτων. Συνεπώς, η κωδικό λέξη ενός μηνύματος c = (c0, c1 , ... , ck-1) μπορεί να αναπαρασταθεί με τη μορφή v = c0·g0 + c1·g1 + ... + ck-1 gk-1. Τα k (n−διαστατών) διανύσματα που δημιουργούν τον κώδικα g0, g1 , ... , gκ-1 μπορούν να αποτελέσουν τις γραμμές ενός πίνακα G διαστάσεων k×n.

1.3.2 Συστηματικοί κώδικες δομής (Systematic block codes)

Ένας κώδικας δομής λέγεται συστηματικός (systematic) όταν η ακολουθία δυαδικών ψηφίων, δηλαδή το μήνυμα, αποτελεί σαφές τμήμα της κωδικής λέξης, όταν δηλαδή τα k πληροφοριακά bits (information bits) βρίσκονται στην αρχή της κωδικής λέξης. Τα υπόλοιπα n-k δυαδικά ψηφία αποτελούν τα bits ισοτιμίας (parity bits). Η διάταξη αυτή επιτρέπει την άμεση εξαγωγή των πληροφοριακών bits από την κωδική λέξη.

1.3.3 Πίνακας έλεγχου ισοτιμίας

Για την ανίχνευση σφαλμάτων χρησιμοποιείται ο Πίνακας Έλεγχου Ισοτιμίας (parity check matrix) H, ο οποίος και εξετάζει αν η λέξη που φτάνει στο δέκτη αποτελεί κωδικό λέξη ή όχι. Ο πίνακας έλεγχου ισοτιμίας δύνεται από το συνδυασμό του μοναδιαίου πίνακα I_{n-k} με τον ανάστροφο του πίνακα P, τον P^T.

Συγκεκριμένα, ο πίνακας H περιγράφεται από τη σχέση:

$H = [I_{n-k}\ P^T]$ (7)

Ο έλεγχος σφαλμάτων που πραγματοποιείται από τον πίνακα έλεγχου ισοτιμίας, Η, γίνεται εξετάζοντας αν ισχύει η σχέση:

$$v.H^T=0 \text{ (αριθμητική modulo-2)} \quad (8)$$

όπου το v αναπαριστά την κωδική λέξη και το H^T αναπαριστά τον ανάστροφο πίνακα του Η. Όπως γίνεται εύκολα αντιληπτό από τα παραπάνω, ο πίνακας Η είναι 36 ένας πίνακας με διαστάσεις (n-k) × n του οποίου τα στοιχεία είναι "0" και "1", εφόσον αναφερόμαστε σε δυαδική πληροφορία.

1.3.4 Απόσταση και Βάρος Hamming

Σε οποιαδήποτε κωδική λέξη ορίζεται ένας συγκεκριμένος αριθμός ο οποίος καλείται βάρος Hamming (Hamming weight). Η ποσότητα αυτή αντιστοιχεί στον αριθμό των μη μηδενικών στοιχείων της κωδικής λέξης. Ως απόσταση Hamming (Hamming distance) d, μεταξύ δύο κωδικών λέξεων ορίζεται ο αριθμός των θέσεων στις οποίες διαφέρουν οι δύο λέξεις.

Για παράδειγμα, έστω οι κώδικες λέξεις v = 11010 και u = 10111. Παρατηρούμε ότι οι λέξεις αυτές διαφέρουν στα δυαδικά ψηφία a0, a2 και a3, όπου το bit a0 αντίστοιχα στο ελάχιστης σημασίας bit (Least Significant Bit, LSB), δηλαδή σε τρεις θέσεις, συνεπώς η απόσταση Hamming είναι d = 3. Ως ελάχιστη απόσταση (minimum distance) dmin, ενός κώδικα δομής ορίζεται η ελάχιστη απόσταση Hamming μεταξύ δύο ζευγών κωδικών λέξεων του κώδικα. Η ελάχιστη απόσταση αποτελεί σημαντική παράμετρο καθώς καθορίζει την ικανότητα ανίχνευσης και διόρθωσης σφαλμάτων του κώδικα.

Συγκεκριμένα, σε ένα γραμμικό κώδικα δομής με ελάχιστη απόσταση dmin αποδεικνύεται ότι σε κάθε κωδικό λέξη υπάρχει η δυνατότητα να ανιχνευθούν **s ≤ (dmin - 1)** σφάλματα και να διορθωθούν αντίστοιχα **t ≤ (dmin - 1)/2** σφάλματα.

ΚΕΦΑΛΑΙΟ 2 LDPC ΚΩΔΙΚΕΣ

2.1. Τρόποι αναπαράστασης των LDPC κωδίκων

Οι Κώδικες - Χαμηλής - Πυκνότητας - Ελέγχου - Ισοτιμίας (LDPC CODES) μπορούν να αναπαρασταθούν με δύο τρόπους. Όπως το σύνολο των κωδίκων δομής, δύνανται να περιγραφούν μέσω πινάκων. Υπάρχει όμως και μία εναλλακτική μέθοδος απεικόνισης, η οποία χρησιμοποιεί γράφους Tanner.

2.1.1 Αναπαράσταση με πίνακα

Εφόσον οι κώδικες που μελετάμε ανήκουν στην κατηγορία των γραμμικών κωδίκων δομής (Linear Block Codes), προκύπτουν από έναν πίνακα G διαστάσεων K×n, ο οποίος λέγεται γεννήτορας πίνακας, ενώ οι αριθμοί K και n αντιστοιχούν στον αριθμό ψηφίων του προς μετάδοση μηνύματος και της κωδικής λέξης αντίστοιχα. Όπως έχει αναφερθεί και στο προηγούμενο κεφάλαιο, ο πίνακας αυτός δημιουργείται από ένα σύνολο K γραμμικώς ανεξάρτητων n-διαστάσεων διανυσμάτων, $g_0,g_1,\ldots,g_{K-1}$. Ο γεννήτορας πίνακας συνδέει το προς μετάδοση μήνυμα c με την κωδική λέξη v, αφού κάθε κωδική λέξη γράφεται ως εξής:

$$v = c \cdot G \quad (9)$$

Ο γεννήτορας πίνακας έχει τη μορφή $O = [P\ I_K]$, δηλαδή αποτελείται από τον Kx(n-K) πίνακα P και από τον KxK μοναδιαίο πίνακα I_K. Γενικά όμως οι γραμμικοί κώδικες δομής, περιγράφονται κυρίως από τον Πίνακα Ελέγχου Ισοτιμίας (Parity Check Matrix), H. Ο πίνακας αυτός έχει τη μορφή

$H = [I_{n-k}\ P^T]$, δηλαδή προκύπτει από το συνδυασμό του μοναδιαίου (n - K) x (n - K) I_{n-k} πίνακα με τον ανάστροφο του πίνακα P, διάστασης (n-K) x K. Όπως γίνεται εύκολα αντιληπτό, ο πίνακας H έχει διάσταση (n - K) x n. Συγκεκριμένα, το πλήθος των γραμμών του αντιστοιχεί στο πλήθος των πλεοναζόντων ψηφίων ελέγχου (Check Bits) που εισάγονται με την κωδικοποίηση, ενώ ο αριθμός των στηλών του ισούται με τον αριθμό των ψηφίων από τον οποίο αποτελείται μία κωδική λέξη. Ο πίνακας H πραγματοποιεί n – K ελέγχους ισοτιμίας σε κάθε κωδική λέξη που φτάνει στον αποκωδικοποιητή.

Οι κώδικες Χαμηλής Πυκνότητας Ελέγχου Ισοτιμίας, LDPC αποτελούν μία συγκεκριμένη κατηγορία γραμμικών κωδίκων δομής, της οποίας το βασικό χαρακτηριστικό συνίσταται στην χαμηλή πυκνότητα του πίνακα ελέγχου ισοτιμίας σε μη μηδενικά στοιχεία. Αυτό σημαίνει ότι ο πίνακας H της συγκεκριμένης κατηγορίας κωδίκων αποτελείται κυρίως από μηδενικά στοιχεία και μόνο από έναν πολύ μικρό αριθμό μονάδων. Από το χαρακτηριστικό αυτό προκύπτει και η ονομασία των συγκεκριμένων κωδίκων **(χαμηλής πυκνότητας).**

Ακολούθως, δίνεται ένας πίνακας χαμηλής πυκνότητας ελέγχου ισοτιμίας για έναν κώδικα, δηλαδή για κώδικα με μεταδιδόμενη πληροφορία αποτελούμενη από 4 ψηφία και με 4 ψηφία ελέγχου.

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Δύο σημαντικά μεγέθη που πρέπει να λαμβάνονται υπόψη σε έναν πίνακα ελέγχου ισοτιμίας είναι το πλήθος των μη μηδενικών στοιχείων σε κάθε γραμμή του πίνακα, και το πλήθος των μη μηδενικών στοιχείων σε κάθε

στήλη του πίνακα Τα μεγέθη αυτά ονομάζονται βαθμός γραμμής Wr και βαθμός στήλης Wc αντίστοιχα. Για να μπορεί να χαρακτηριστεί ένας πίνακας ως χαμηλής-πυκνότητας πίνακας ,θα πρέπει να ικανοποιούνται οι συνθήκες:

$$Wc << n \text{ και } Wr << m \quad (10)$$

Πιο αναλυτικά, θα πρέπει ο αριθμός των στοιχείων 1 σε μία στήλη του πίνακα να είναι κατά πολύ μικρότερος από το πλήθος n των στηλών, δηλαδή από το μήκος της κωδικής λέξης, και αντίστοιχα ο αριθμός των στοιχείων 1 σε μία γραμμή του πίνακα να είναι κατά πολύ μικρότερος από το πλήθος m των γραμμών, δηλαδή από το μήκος του προς μετάδοση μηνύματος. Για την ικανοποίηση των ανωτέρω συνθηκών, ο πίνακας ελέγχου ισοτιμίας πρέπει να είναι πολύ μεγάλος. Συνεπώς, ο πίνακας του παραδείγματος δεν μπορεί να χαρακτηριστεί ως πίνακας χαμηλής πυκνότητας, απλά χρησιμοποιείται για την κατανόηση των χαρακτηριστικών ενός τέτοιου πίνακα.

2.1.2 Γραφική αναπαράσταση

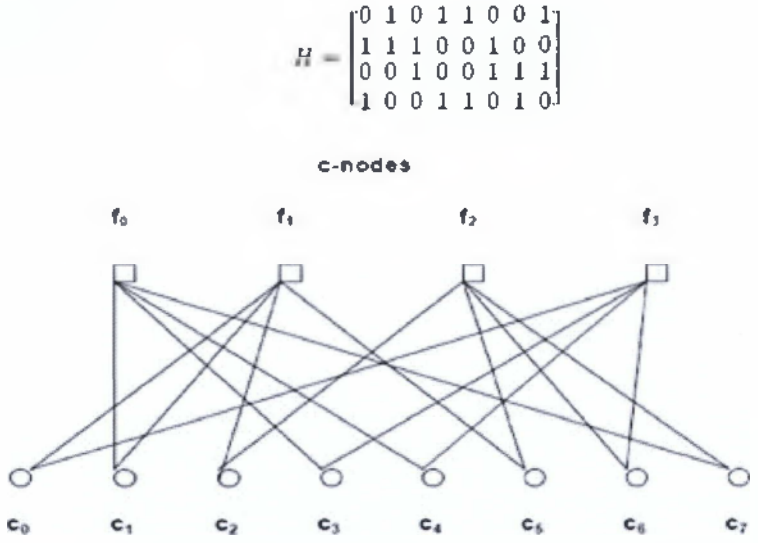
Ο δεύτερος τρόπος αναπαράστασης των LDPC κωδίκων προτάθηκε από τον Tanner, ο οποίος εισήγαγε τους γράφους Tanner. Οι γράφοι Tanner (Tanner Graphs) εκτός από το γεγονός ότι παρέχουν μία πλήρη περιγραφή του κώδικα, βοηθούν επίσης στην επεξήγηση του αλγόριθμου αποκωδικοποίησης που χρησιμοποιείται. Οι γράφοι Tanner ανήκουν στην κατηγορία των διμερών γράφων. Ένας γράφος ονομάζεται «διμερής» όταν οι κόμβοι του χωρίζονται σε δύο ομάδες και μόνο κόμβοι διαφορετικών ομάδων μπορούν να ενωθούν μεταξύ τους. Οι δύο τύποι κόμβων που υπάρχουν σε ένα γράφο Tanner ονομάζονται κόμβοι μεταβλητών (variable nodes), οι οποίοι συνήθως αναφέρονται ως v-nodes και κόμβοι ελέγχου

(check nodes), οι οποίοι συνήθως αναφέρονται ως c-nodes.Ο γράφος Tanner ενός κώδικα σχεδιάζεται ακολουθώντας το εξής κανόνα:

Ο i-οστός κόμβος ελέγχου συνδέεται με τον j-οστό κόμβο μεταβλητών μόνο όταν το στοιχείο h_{ij} του πίνακα ελέγχου ισοτιμίας, Η είναι ισο με 1.

Μπορούμε να συμπεράνουμε ότι ο γράφος περιλαμβάνει m = n - K κόμβους ελέγχου, έναν για κάθε ψηφίο ελέγχου, και n κόμβους μεταβλητών, έναν για κάθε ψηφίο της κωδικής λέξης. Επιπλέον, οι m γραμμές του πίνακα Η, ορίζουν m συνδέσεις κόμβων ελέγχου και οι n στήλες ορίζουν n συνδέσεις κόμβων μεταβλητών.

Στο σχήμα που ακολουθεί παρουσιάζεται ο γράφος του κώδικα που αντιστοιχεί στον πίνακα Η του προηγούμενου σχήματος.



Σχήμα 3. Γράφος Tanner και ο αντίστοιχος πίνακας ελέγχου ισοτιμίας

Παρατηρούμε ο κόμβος ελέγχου F_0 συνδέεται με τους κόμβους μεταβλητών οι, c_1, c_3 και c_4 αφού τα στοιχεία h_{01}, h_{03}, h_{04} και h_{07} του πίνακα Η έχουν τιμή 1. Με τον ίδιο τρόπο, ο κόμβος f_1 συνδέεται με τους c_0, c_1, c_2 και c_4 αφού $h_{10} = h_{11} = h_{12} = h_{14} = 1$. Ομοίως ο c-node f_2 συνδέεται με τους v-nodes c_2, c_5, c_6 και c_7, διότι $h_{20} = h_{25} = h_{26} = h_{27} = 1$ και τέλος, ο κόμβος f_3 συνδέεται με τους κόμβους c_0, c_3, c_4 και c_6 δηλώνοντας ότι τα h_{30}, h_{33}, h_{34} και h_{36} στοιχεία έχουν τιμή 1.

Γνωρίζοντας ότι σε κάθε κώδικα δομής πρέπει να ισχύει η σχέση $v \cdot H^T = 0$, ώστε να είναι δυνατή η ανίχνευση και διόρθωση σφαλμάτων, παρατηρούμε ότι οι τιμές των ψηφίων της πληροφορίας (variable nodes), που συνδέονται στον ίδιο κόμβο ελέγχου (check nodes), πρέπει να δίνουν μηδενικό άθροισμα. Όλα τα ψηφία που συνδέονται σε έναν κόμβο ελέγχου έχουν modulo 2 άθροισμα ίσο με το μηδέν ή, ισοδύναμα, έχουν αποτέλεσμα μηδέν στην πράξη XOR.

2.2 Κανονικοί (Regular) και Μη κανονικοί (Irregular) κώδικες

Οι κώδικες LDPC ταξινομούνται σε δύο κατηγορίες, τους ομαλούς (regular) και τους ανώμαλους (irregular) κώδικες. Ένας **κανονικός** κώδικας LDPC χαρακτηρίζεται από την ύπαρξη ενός πίνακα ελέγχου ισοτιμίας, Η, ο οποίος περιέχει ακριβώς W_c μη μηδενικά στοιχεία σε κάθε του στήλη αλλά και $W_r = W_c \cdot (n/m)$ μη μηδενικά στοιχεία σε κάθε γραμμή του. Στο παράδειγμα που περιγράφεται από το σχήμα , ο LDPC κώδικας είναι ομαλός, με βαθμό στήλης $W_c = 2$ και βαθμό γραμμής $W_r = 2 \cdot (8/4) = 4$.

Πληροφορίες για την ομαλότητα του κώδικα μπορούμε να πάρουμε και από τον γράφο Tanner. Ο κώδικας είναι ομαλός όταν ο αριθμός των εισερχόμενων γραμμών είναι ίδιος για όλους τους κόμβους μεταβλητών και επίσης για όλους τους κόμβους ελέγχου. Ανατρέχοντας στον γράφο του

προηγούμενου παραδείγματος, παρατηρούμε ότι σε κάθε c-node εισέρχονται 2 γραμμές σύνδεσης, ενώ σε κάθε c-node εισέρχονται σταθερά 4 γραμμές σύνδεσης. Αντίθετα, αν το πλήθος των μη μηδενικών στοιχείων των γραμμών ή των στηλών του πίνακα δεν είναι σταθερό, τότε ο πίνακας ονομάζεται **μη κανονικός (irregular).** Στην περίπτωση αυτή, οι παράμετροι W_c και W_r αποτελούν συναρτήσεις του πλήθους των στηλών και των γραμμών αντίστοιχα και τέτοια σημειολογία δεν χρησιμοποιείται εδώ. Έτσι, αντί για τους συγκεκριμένους συμβολισμούς, χρησιμοποιούνται τα πολυώνυμα κατανομής βαθμού κόμβων μεταβλητών (αντιστοιχία με τον βαθμό στήλης W_c) και βαθμού κόμβων ελέγχου (αντιστοιχία με τον βαθμό γραμμής W_r) (variable node and check node degree distribution polynomials), τα οποία συμβολίζονται ως **λ(χ)** και **ρ(χ)** αντίστοιχα.
Στο πολυώνυμο:

$$\lambda(\chi) = \sum_{d=1}^{d_v} \lambda_d \cdot \chi^{d-1} \qquad (11)$$

ο όρος d_v υποδηλώνει το μέγιστο βαθμό των variable nodes, ο οποίος αντιστοιχεί στον βαθμό στήλης W_c για τους ομαλούς κώδικες, και ο λ_d υποδηλώνει τον αριθμό στηλών βαθμού d.

Ομοίως, στο πολυώνυμο:

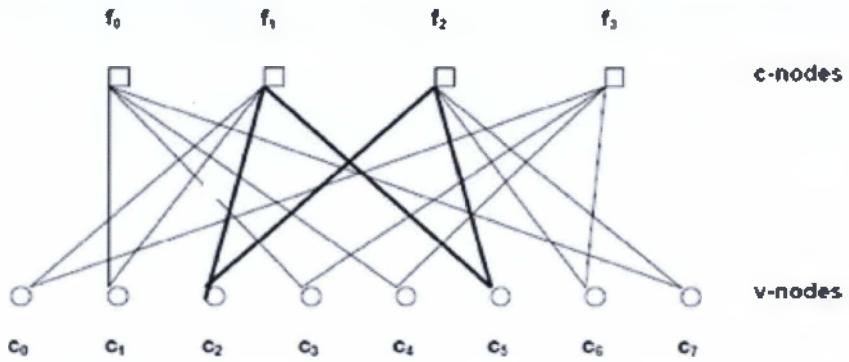
$$\rho(\chi) = \sum_{d=1}^{d_v} \rho_d \cdot \chi^{d-1} \qquad (12)$$

ο όρος d_v υποδηλώνει το μέγιστο βαθμό των variable nodes, ο οποίος αντιστοιχεί στον βαθμό γραμμής W_r για τους ομαλούς κώδικες, και ο ρ_d υποδηλώνει τον αριθμό γραμμών βαθμού d.

Ένα ακόμα μέγεθος που χαρακτηρίζει τους LDPC κώδικες προκύπτει από τη γραφική αναπαράσταση των κωδίκων και είναι ο **κύκλος** (cycle) ή

βρόχος (loop) μήκους ***l***, ο οποίος αποτελεί ένα μονοπάτι με συνολικά ***l*** ακμές, του οποίου η αφετηρία και το τέλος συμπίπτουν.

Στο παράδειγμα που περιγράφεται σε αυτό το κεφάλαιο, παρατηρούμε ο γράφος Tanner που προέκυψε έχει κύκλο μήκους 4 , ο οποίος παρουσιάζεται με έντονες γραμμές στο σχήμα που ακολουθεί.



Σχήμα 4. Κύκλος μήκους 4

Το ελάχιστο δυνατό μήκος ενός διμερούς γράφου είναι ο κύκλος μήκους-4. Όπως παρουσιάζεται και στο παραπάνω σχήμα, ο κώδικας του παραδείγματος περιέχει κύκλο τέτοιου είδους. Κύκλοι αυτής της μορφής, δηλαδή μήκους-4, δηλώνουν την παρουσία τους στον πίνακα ελέγχου ισοτιμίας Η, σαν τέσσερις μονάδες στις γωνίες κάποιου υπο-πίνακα του Η. Οι μικροί κύκλοι, όπως και ο κύκλος του παραδείγματος, προτείνεται να αποφεύγονται καθώς υποβαθμίζουν την ποιότητα της αποκωδικοποίησης. Το μήκος του μικρότερου κύκλου ενός κώδικα είναι εκείνο που μας ενδιαφέρει κυρίως, και για το λόγο αυτό αποτελεί χαρακτηριστικό του κώδικα και ονομάζεται girth.

2.3 Αλγόριθμοι Κωδικοποίησης

2.3.1 Το πρόβλημα της κωδικοποίησης

Σε έναν γραμμικό συστηματικό κώδικα δομής, με πίνακα ελέγχου ισοτιμίας H ο οποίος έχει πλήθος γραμμών m και πλήθος στηλών n, υπάρχουν 2^{n-m} ($= 2^k$) έγκυρες κωδικές λέξεις και χρησιμοποιείται πηγή με μπλοκ μεγέθους k = n-m. Το πρόβλημα της κωδικοποίησης έγκειται στην αντιστοίχηση των n-m ψηφίων της πληροφορίας σε n ψηφία της κωδικής λέξης. Επειδή ο κώδικας είναι συστηματικός, τα πληροφοριακά ψηφία (information Bits) αποτελούν υποσύνολο της κωδικής λέξης, ενώ τα υπόλοιπα m ψηφία, είναι τα ψηφία ελέγχου (Parity Bits). Έτσι λοιπόν, η δομή των συστηματικών κωδίκων καθιστά πιο εύκολο τον εντοπισμό των ψηφίων ελέγχου από τον δέκτη. Απαιτείται λοιπόν η εκτέλεση δύο βημάτων:

α) Καθορισμός του τμήματος της κωδικής λέξης του συστηματικού κώδικα που αποτελεί την πληροφορία της πηγής και του τμήματος που αποτελεί τα ψηφία ισοτιμίας.

β) Εύρεση του τρόπου με τον οποίο θα υπολογιστούν τα ψηφία ελέγχου (check bits), δεδομένων των n-m ψηφίων της πηγής. Η απαιτούμενη διαδικασία μπορεί να πραγματοποιηθεί με δύο μεθόδους.

2.3.2 Μέθοδος κωδικοποίησης για πυκνούς πίνακες (Dense Encoding Method)

Ο πίνακας ελέγχου ισοτιμίας H διαιρείται σε δύο υπό-πίνακες, τον **A,** ο οποίος έχει διάσταση mxm και αποτελεί το αριστερό τμήμα, και τον πίνακα **B,** ο οποίος έχει διάσταση mx(n-m) και αποτελεί το δεξί τμήμα. Η μέθοδος αυτή προϋποθέτει ότι έχουν γίνει οι κατάλληλες ενέργειες στον πίνακα A ώστε να είναι μη ιδιάζων, να μην έχει δηλαδή μηδενική ορίζουσα (*detA ≠ 0*). Ο πίνακας **H** λοιπόν παίρνει τη μορφή:

$$H = [A\ B] \quad (13)$$

Ομοίως, η κωδική λέξη **v,** διαιρείται σε m ψηφία ελέγχου, **c,** και σε n-m ψηφία της προς μετάδοση πληροφορίας, **s.** Έτσι η εξίσωση ελέγχου ισοτιμίας $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$ μετατρέπεται στην εξίσωση:

$$[c\ \ s] \cdot \begin{bmatrix} A \\ B \end{bmatrix} = 0 \qquad \text{(αριθμητική modulo-2)} \qquad (14)$$

από την οποία προκύπτει η σχέση:

$$Ac + Bs = 0 \quad (15)$$

Τελικά καταλήγουμε στη σχέση:

$$c = A^{-1} \cdot B \cdot s \quad (16)$$

η οποία προσφέρει το διάνυσμα των ψηφίων ελέγχου ισοτιμίας.

Για την εύρεση του διανύσματος c, μπορούμε αρχικά να υπολογίσουμε την ποσότητα $A^{-1} \cdot B$ και στη συνέχεια να υπολογίσουμε τα ψηφία ελέγχου ο πολλαπλασιάζοντας τα ψηφία της πηγής με τον πίνακα που προέκυψε από τον προηγούμενο υπολογισμό. Η εν λόγω διαδικασία απαιτεί υπολογιστικό χρόνο ανάλογο της ποσότητας $m \cdot (n - m)$.

2.3.3 Μεικτή μέθοδος κωδικοποίησης (A mixed encoding method)

Βασική υπόθεση στη δεύτερη περίπτωση αποτελεί το γεγονός ότι θεωρούμε αραιό τον πίνακα H = [A B] και κατά συνέπεια και τον πίνακα B. Στους LDPC κώδικες το πλήθος των μη μηδενικών στοιχείων σε μία γραμμή είναι σταθερό, τουλάχιστον κατά μέσο όρο, ανεξάρτητα από την ποσότητα n. Ο υπολογισμός της ποσότητας $c = A^{-1} \cdot B \cdot s$ είναι ταχύτερος αν πραγματοποιηθεί σε δύο βήματα.

α) Υπολογισμός του $z = B \cdot s$, διαδικασία που απαιτεί υπολογιστικό χρόνο ανάλογο του M, εκμεταλλευόμενοι το πλεονέκτημα του πίνακα B, ο οποίος είναι αραιός.
β) Υπολογισμός της ποσότητας $c = A^{-1} \cdot z$, σε χρόνο ανάλογο του m^2. Ο συνολικός χρόνος που απαιτείται είναι της τάξεως του m^2, ο οποίος καθιστά τη μέθοδο αυτή ταχύτερη από την προηγούμενη, η οποία απαιτούσε χρόνο ανάλογο του $m \cdot (n - m)$, κυρίως σε περιπτώσεις που ισχύει $m < n-m$, όταν δηλαδή ο ρυθμός κώδικα (code rate) είναι μεγαλύτερος του 1/2.

Υπολογισμός του $c = A^{-1} \cdot z$

Για τον υπολογισμό των ψηφίων ελέγχου ισοτιμίας, απαιτείται η επίλυση του συστήματος $c = A^{-1} \cdot z$. Για την εξασφάλιση ταχύτερων αποτελεσμάτων δεν χρησιμοποιείται η κλασσική διαδικασία επίλυσης του συστήματος. Πιο αναλυτικά, εφαρμόζεται μία λιγότερο χρονοβόρα διαδικασία, η οποία ονομάζεται ανάλυση σε τριγωνικούς πίνακες **L** και **U (LU Decomposition).** Η σχέση που συνδέει τον πίνακα **A** με τους L και U είναι η $A = L \cdot U$. Ο πίνακας L είναι ένας **κάτω τριγωνικός** πίνακας (lower triangular), δηλαδή

ένας πίνακας που τα στοιχεία πάνω από την κύρια διαγώνιό του είναι μηδενικά, ενώ ο U είναι **άνω τριγωνικός** (upper triangular), περιέχει δηλαδή μηδενικά στοιχεία κάτω από την κύρια διαγώνιό του.

Το πρώτο τμήμα της διαδικασίας επίλυσης του συστήματος, είναι η εύρεση του άνω τριγωνικού πίνακα, U, και του κάτω τριγωνικού πίνακα L. Στη συνέχεια επιλύεται το σύστημα $\mathbf{L \cdot y = z}$ για την εύρεση του $\mathbf{y = L^{-1} \cdot z}$, με αντικατάσταση προς τα εμπρός (forward substitution). Τέλος, επιλύεται το σύστημα $\mathbf{U \cdot c = y}$ με πίσω αντικατάσταση (backward substitution), του οποίου η λύση παρέχει και το διάνυσμα των ψηφίων ελέγχου, $\mathbf{c = U^{-1} \cdot y = U^{-1} \cdot L^{-1} \cdot z}$. Ωστόσο, και στις παραπάνω προσεγγίσεις, η πολυπλοκότητα κωδικοποίησης είναι πολύ μεγάλη, $O(n^2)$, και καθιστά αδύνατη η χρήση τους ιδιαίτερα για μεγάλου μήκους λέξεις όπως εκείνες που χρησιμοποιούνται σε νέες τηλεπικοινωνιακές εφαρμογές, όπως δορυφορικό Digital video Broadcast (DVB) DVB-S2, IEEE 802.3an (10GBASE-T) και IEEE 802.16 (WiMAX). Έτσι, προκύπτει η πρόκληση για εύρεση ενός αποδοτικού αλγορίθμου κωδικοποίησης με γραμμική πολυπλοκότητα χρόνου, αλλά και γραμμική πολυπλοκότητα χώρου για την επίλυση του προβλήματος κωδικοποίησης των παραπάνω εφαρμογών.

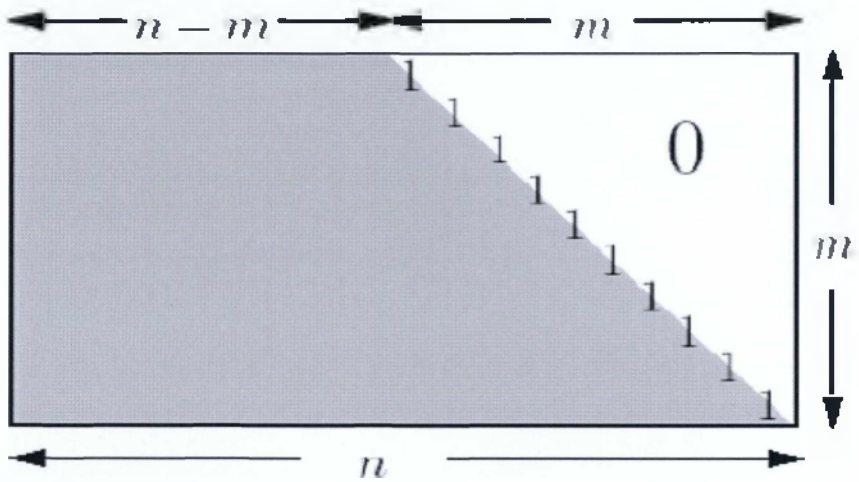
2.3.4 RU LDPC - Αλγόριθμος Κωδικοποίησης

Όπως αναφέρθηκε στην αρχή του κεφαλαίου, οι κώδικες LDPC είναι γραμμικοί κώδικες δομής και η κωδικοποίηση τέτοιων κωδίκων ακολουθεί την σχέση:

$$H X^T = 0 \quad (17)$$

Όπου X είναι η κωδική λέξη και H είναι ο πίνακας ελέγχου ισοτιμίας (Parity check matrix). Μια απευθείας μέθοδος κωδικοποίησης απαιτεί 3 βασικά βήματα:

1. Gaussian Elimination, για τη μετατροπή του πίνακα H σε μια μορφή κάτω τριγωνικό (Lower Triangular).



Σχήμα 5. Ο πίνακας ελέγχου ισοτιμίας σε μορφή κάτω τριγωνικό.

2. Διαχωρισμό του διανύσματος X σε information bits και parity bits, X= [s, p_1, p_2], όπου **s** είναι το διάνυσμα των information bits, p_1 και p_2 που μαζί αποτελούν το διάνυσμα των parity bits.
3. Τελευταίο βήμα, η επίλυση της εξίσωσης, χρησιμοποιώντας αντικατάσταση προς τα μπρος (forward substitution).

Η πολυπλοκότητα της παραπάνω μεθόδου είναι πολύ μεγάλη αφού μόνο για το 1° βήμα, η πολυπλοκότητα της Gaussian Elimination είναι $O(n^3)$. Επιθυμητό είναι η πολυπλοκότητα να είναι $O(n^2)$, για την ακρίβεια, $O(n^2 \frac{r(1-r)}{2})$, όπου r είναι ο ρυθμός του κώδικα (code rate), επομένως, έπρεπε να βρεθεί μια καλύτερη και πιο αποδοτική μέθοδος για την επίλυση των

παραπάνω βημάτων ή ακόμα, να βρεθεί καλύτερος αλγόριθμος που να πλησιάζει την γραμμική πολυπλοκότητα. Για το λόγο αυτό, ο **Richardson** και ο **Urbanke** εκμεταλλεύτηκαν την πολύτιμη ιδιότητα και χαρακτηριστικό του πίνακα H, την **αραιότητα** του πίνακα (sparsity of H matrix) .

Ο **Richardson** και ο **Urbanke (RU)** βρήκαν ότι στις περισσότερες περιπτώσεις, για τους κώδικες που χρησιμοποιούσαν, η πολυπλοκότητα κωδικοποίησης μπορεί να είναι γραμμική ή τετραγωνική ($2^{ου}$ βαθμού) αλλά ελεγχόμενη και αρκετά βολική. Για παράδειγμα, για ένα ομαλό κώδικα με μήκος λέξης, παρόλο η πολυπλοκότητα παραμένει τετραγωνική, ο απαιτούμενος πραγματικός όμως αριθμός των επεξεργαστικών στοιχείων **(XOR),** είναι O(n) αντί για $0.017^n n^2$, και αφού ο 0.017^2 είναι μικρός αριθμός, η πολυπλοκότητα του κωδικοποιητή παραμένει ελεγχόμενη για μεγάλου μήκους λέξεις. Ο RU αλγόριθμος είναι από τους πρώτους αλγορίθμους κωδικοποίησης που είχαν ως σκοπό, τη επίλυση της πολυπλοκότητας του κωδικοποιητή και εύρεσης μια αποδοτική μέθοδο κωδικοποίησης.

Η υψηλή αποδοτικότητα του κωδικοποιητή επιτυγχάνεται από την σωστή εκμετάλλευση της αραιότητας του πίνακα ελέγχου ισοτιμίας **H,** και από τον αλγόριθμο που θα μπορούσε να εφαρμοστεί πάνω σε οποιοδήποτε **«αραιό»** πίνακα H. Ο αλγόριθμος χρησιμοποιεί ένα δυαδικό πίνακα, και εφαρμόζεται γενικά σε πίνακες H των οποίων τα ορίσματα ανήκουν στο πεδίο F με την προϋπόθεση ότι οι γραμμές του πίνακα είναι γραμμικά ανεξάρτητες. Επομένως ο αλγόριθμος που υλοποιεί τον κωδικοποιητή θα ανιχνεύει αν δεν ισχύει αυτή ιδιότητα, θα ανιχνεύει δηλαδή την γραμμική εξάρτηση μεταξύ των γραμμών του πίνακα, έτσι υποχρεωτικά, θα πρέπει να χρησιμοποιηθεί άλλος πίνακας H ή, να ελαχιστοποιηθεί ο αριθμός των πλεοναζόντων γραμμών στον πίνακα κατά τη διάρκεια της κωδικοποίησης.

Θεωρώντας ένα πίνακα ελέγχου ισοτιμίας $H_{m\times n}$ πάνω στο πεδίο F, και εξ ορισμού, ο συσχετισμένος κώδικας αποτελείται από το σύνολο των διανυσμάτων n στοιχείων X έτσι ώστε :

$$H\,x^T = 0 \quad (18)$$

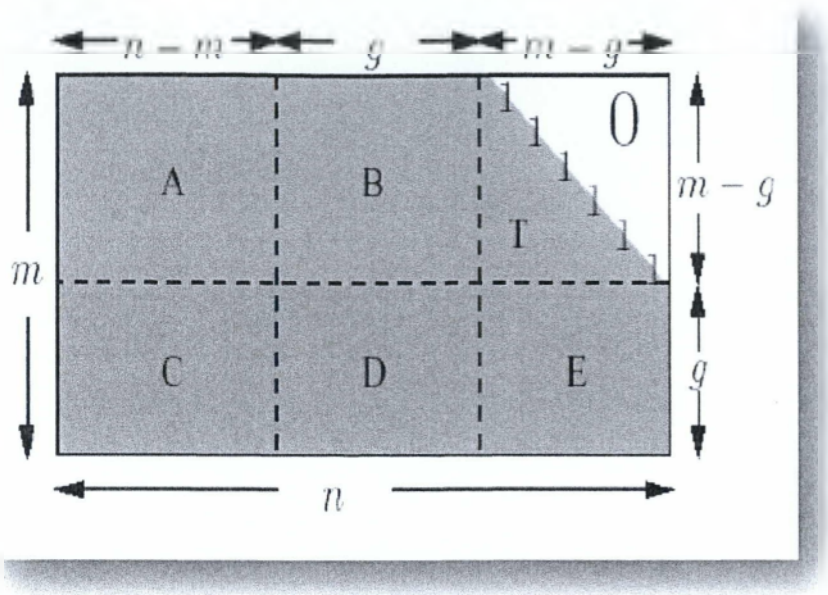
Ακολουθώντας τα 3 βήματα που αναφέρονται στη προηγούμενη παράγραφο και θεωρώντας χ = (s,p) όπου s ϵ F^{n-m} και p ϵ F^m υλοποιούμε ένα συστηματικό κωδικοποιητή ως, εξής: Συμπληρώνουμε το διάνυσμα **s** με τα επιθυμητά information σύμβολα μήκους (n-m).

1. Προσδιορισμός των m συμβόλων ελέγχου ισοτιμίας χρησιμοποιώντας αντικατάσταση προς τα πίσω (back_substitution), πιο αναλυτικά, $\forall l \epsilon$ [m], υπολογίζουμε:

$$\mathbf{p}_l = \sum_{j=1}^{n-m} H_{l,j} \quad S_j + \sum_{j=1}^{l-1} H_{l,j+n-m} \quad P_j \qquad (19)$$

Η μετατροπή του πίνακα Η στη επιθυμητή μορφή απαιτεί $O(n^3)$ πράξεις, οι οποίες γίνονται off-line ως προ-επεξεργασία, ενώ πραγματική κωδικοποίηση απαιτεί $O(n^2)$ αφού μετά από την προ-επεξεργασία, ο πίνακας Η δεν είναι πια **αραιός.**

Υποθέτοντας ότι μπορούμε να πετύχουμε τη μετατροπή του πίνακα Η, αλλάζοντας μονό την σειρά των γραμμών και των στηλών του πίνακα (ROW AND COLUMN PERMUTATIONS), έτσι ώστε να έχει τη μορφή που φαίνεται στο Σχήμα 6. Ο πίνακας παραμένει αραιός και είναι σε μια προσεγγιστική μορφή κάτω τριγωνικού πίνακα (approximate lower triangular form) .



Σχήμα 6. Η καινούρια μορφή του πίνακα ελέγχου ισοτιμίας, παραμένει η μορφή του κάτω τριγωνικού πίνακα.

Η καινούρια μορφή του πίνακα Η γίνεται:

H=(A B T
C D E) (20)

Όπου ο Α έχει διαστάσεις(m-g)*(n-m), ο Β είναι (m-g)*g, ο C είναι g*(n-m), ο D είναι g*g, ο Τ είναι (m-g)*(m-g) και ο Ε είναι g*(m-g) και χαρακτηριστικό όλων των πινάκων αυτών είναι η αραιότητα, και ο πίνακας Τ είναι κάτω τριγωνικός και έχει μόνο άσσους «1» κατά μήκος του διαγωνίου του.

διαγωνίου του. Θέτουμε, Φ := -ET^{-1}B +D και υποθέτουμε, για αρχή, ότι είναι non-singular (η ορίζουσα det(Φ) μη μηδενική), έτσι από την εξίσωση προκύπτει :

$$p_1^T = -\Phi^{-1}(-ET^{-1}A + C)s^T \qquad (21)$$

Εφόσον ο πίνακας -Φ^{-1}(-ET^{-1}A + C) έχει διαστάσεις g*(n-m) και έχει προυπολογιστεί off-line, έτσι η πολυπλοκότητα για τον υπολογισμό το p_1 εκτιμάται με O(g*(n-m)).

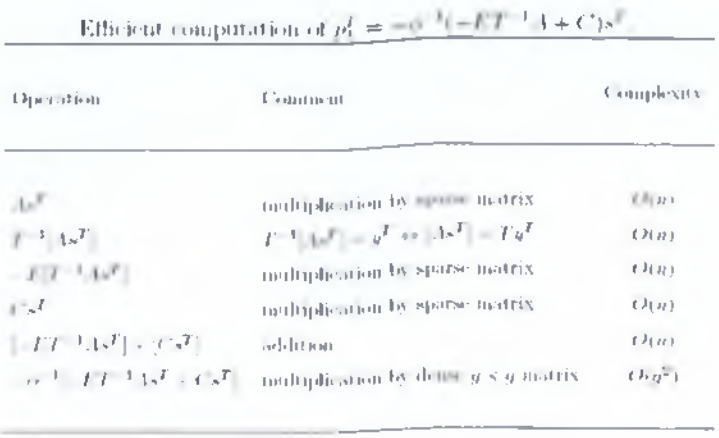
Η πολυπλοκότητα υπολογισμού του p_1 θα μπορούσε να μειωθεί, όπως δείχνεται παρακάτω στον πίνακα, και αυτό αντί του προ-υπολογισμού του -Φ^{-1}(-ET^{-1}A + C) και μετά να πολλαπλασιαστεί με το s^T, θα μπορούσε το p_1 να υπολογιστεί, σπάζοντας την εξίσωση σε μικρότερα βήματα.

- Αρχικά υπολογίζουμε το As^T, πολλαπλασιασμός με πολυπλοκότητα O(n) αφού ο πίνακας A είναι αραιός **(spare).**
- Έπειτα, πολλαπλασιάζουμε το αποτέλεσμα με το T^{-1}, ισχύει όμως ότι: (T^{-1} [As^T] = y^T) = ([As^T] = Ty^T) και αυτό επιτυγχάνεται με πολυπλοκότητα O(n) χρησιμοποιώντας αντικατάσταση προς τα πίσω (back substitution), αφού ο πίνακας T είναι κάτω τριγωνικός και αραιός.
- Τα υπόλοιπα βήματα είναι αρκετά απλά, και δείχνουν ότι η συνολική πολυπλοκότητα για τον προσδιορισμό το p_1 είναι **O(g*(m-n)).**

Με παρόμοιο τρόπο, σύμφωνα με την εξίσωση , προκύπτει:

$$p_2^T = -T^{-1}(As^T + Bp_1^T) \quad (22)$$

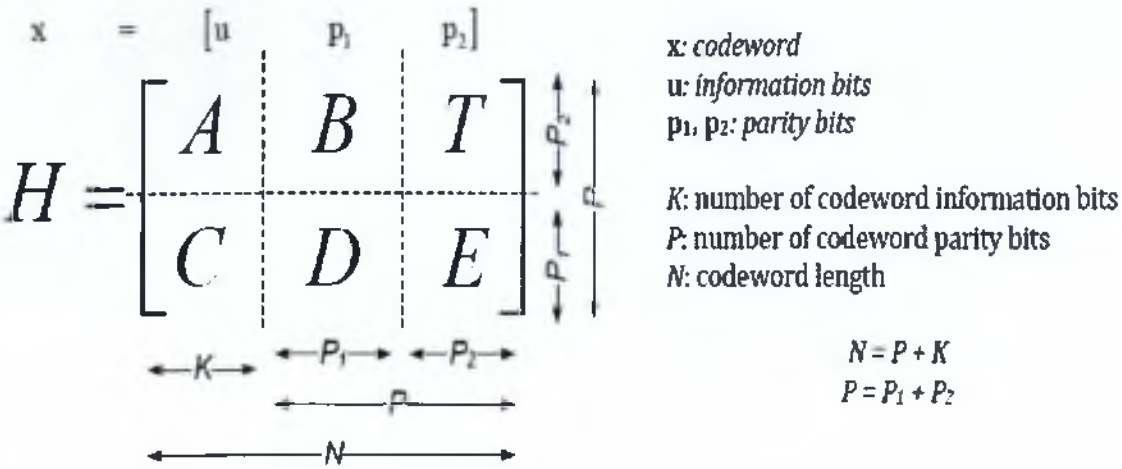
και η πολυπλοκότητα για τον προσδιορισμό του p_2 εκτιμάται με O(n).



Σχήμα 7. Πολυπλοκότητα υπολογισμού του ψηφίων του p_1.

2.3.5 Προσεγγιστική υλοποίηση του κωδικοποιητή RU.

Χρησιμοποιώντας τον αλγόριθμο του R&U, οδηγούμαστε σε μια υλοποίηση του κωδικοποιητή. Το σχήμα παρακάτω, δείχνει την δομή του πίνακα H σύμφωνα με τον αλγόριθμο R&U. Παρακάτω δίνονται τα βήματα για τον υπολογισμό των parity bits.



Σχήμα 8. Δομή του πίνακα Η, βάση τον αλγόριθμο R&U.

Το πρώτο διάνυσμα των parity bits P_1 , προκύπτει από τους information bits σύμφωνα με την παρακάτω σχέση:

$$p_1^T = [(D + E \cdot T^{-1} \cdot B)^{-1} \cdot (C + E \cdot T^{-1} \cdot A)] \cdot u^T \quad (23)$$

Για προσδιοριστούν τα υπόλοιπα parity bits (διάνυσμα P_2), υπάρχουν 2 επιλογές.

(a) Η πρώτη είναι να χρησιμοποιηθεί το διάνυσμα Pι που προκύπτει από τη σχέση:

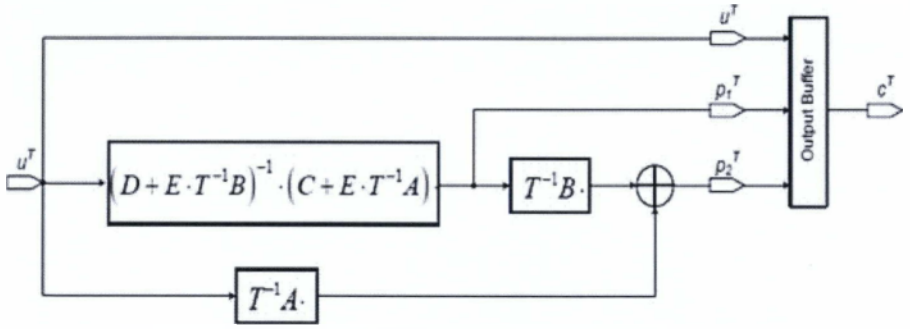
$$p_2^T = (T^{-1} \cdot A) \cdot u^T + (T^{-1} \cdot B) \cdot p_1^T \quad (24)$$

(b) Η δεύτερη επιλογή είναι να υπολογιστούν κατευθείαν από τους **information bits u,** σύμφωνα με την παρακάτω σχέση

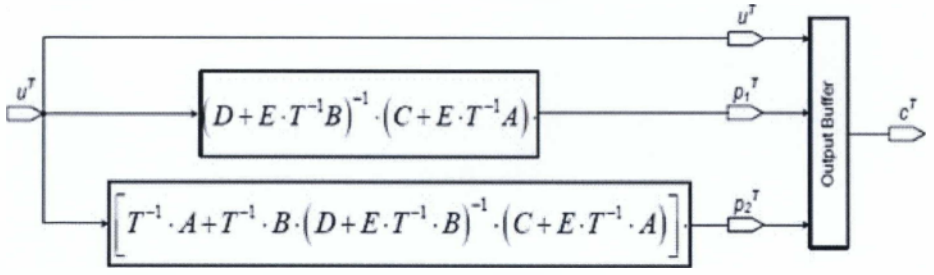
$$p_2^T = [(T^{-1} \cdot A) + (T^{-1} \cdot B) \cdot (D + E \cdot T^{-1} \cdot B)^{-1}(C + E \cdot T^{-1} \cdot A)] \cdot u^T$$

(25)

Στο σχήμα που ακολουθεί φαίνεται η αρχιτεκτονική και για τις 2 αυτές επιλογές υπολογισμού. Η πολυπλοκότητα της διαδικασίας κωδικοποίησης σε χρόνο αλλά και σε χώρο, εξαρτάται από την πυκνότητα των πινάκων του κωδικοποιητή, πόσο αραιοί είναι.



Σχήμα 9a. Η αρχιτεκτονική της επιλογής (a)



Σχήμα 9b. Η αρχιτεκτονική της επιλογής (b)

2.4 Αποκωδικοποίηση LDPC

2.4.1 Ο γράφος Tanner

Ένα χρήσιμο εργαλείο αναπαράστασης ενός LDPC κώδικα είναι ο γράφος Tanner ένας διμερής γράφος που αναπαριστά τις εξισώσεις ισοτιμίας (parity equations) που προκύπτουν από τον πίνακα Η. Ο πίνακας ελέγχου ισοτιμίας Η οφείλει την ονομασία του στο γεγονός ότι εκτελεί m=n-k ξεχωριστούς ελέγχους ισοτιμίας ανά λαμβανόμενη, κωδικοποιημένη λέξη. Ο γράφος Tanner που αντιστοιχεί σε έναν (n, k) κώδικα αποτελείται από n κόμβους bit (bit nodes), από m=n-k κόμβους ελέγχου (check nodes) και από έναν αριθμό ακμών ανάμεσα στις δύο αυτές κατηγορίες κόμβων.

Κάθε κόμβος bit αναπαριστά ένα bit της κωδικοποιημένης λέξης. Κάθε κόμβος ελέγχου αναπαριστά έναν έλεγχο ισοτιμίας. Ακμή μεταξύ κόμβου bit και κόμβου ελέγχου υπάρχει αν και μόνο αν υπάρχει "1" στην αντίστοιχη θέση του Η. Πιο φορμαλιστικά, εάν $H_{ij}=1$ τότε και μόνο τότε υπάρχει σύνδεση ανάμεσα στον κόμβο bit j και στον κόμβο ελέγχου i. Σε ένα γράφο Tanner, κύκλος μήκους ν ονομάζεται ένα μονοπάτι που περιλαμβάνει ν διαφορετικές ακμές και αρχίζει και τελειώνει στον ίδιο κόμβο. Το ελάχιστο μήκος κύκλου αναφέρεται και ως *girth*. Οι καλύτερες επιδόσεις των LDPC κωδίκων παρατηρούνται όταν συνδυάζονται με belief-propagation αλγόριθμο αποκωδικοποίησης, όπου η πιθανότητα κάθε bit διαδίδεται μέσω των ακμών και χρησιμοποιείται ως εξωτερική πληροφορία για την ανανέωση της πιθανότητας άλλων bits. Έχει δειχθεί ότι όσο πιο 'εξωτερική' είναι αυτή η πληροφορία τόσο αυξάνει η επίδοση του κώδικα.

Σε περίπτωση κύκλου μικρού μήκους, η πληροφορία επιστρέφει σχετικά γρήγορα στον αρχικό κόμβο περιέχοντας μικρό ποσό εξωτερικής πληροφορίας κάτι που δεν βοηθά στην αποδοτικότητα του κώδικα. Κατά συνέπεια, το ελάχιστο μήκος κύκλου αποτελεί σημαντικό σχεδιαστικό παράγοντα για τους LDPC κώδικες και αρκετή ερευνητική προσπάθεια καταβάλλεται για την κατασκευή κωδίκων με μεγάλο girth. Από τα παραπάνω κάποιος θα μπορούσε να οδηγηθεί στο συμπέρασμα ότι η ιδανική περίπτωση θα ήταν γράφος δενδρικής δομής, χωρίς κύκλους δηλαδή. Είναι αλήθεια ότι σε τέτοια περίπτωση ο belief-propagation (BP) επαναληπτικός αλγόριθμος αποκωδικοποίησης που θα εξηγηθεί στη συνέχεια τερματίζει πάντα μετά από πεπερασμένο αριθμό επαναλήψεων. Παρά ταύτα, κώδικες με μη κυκλικούς γράφους εμφανίζουν χαμηλές τιμές BER εξαιτίας της μικρής ελάχιστης απόστασης τους: η ελάχιστη απόσταση τους είναι δύο για ρυθμούς κώδικα $R > 1/2$ ($k/n > 1/2$). Έτσι, η παρουσία κύκλων σε αποδοτικούς από όλες τις απόψεις LDPC κώδικες είναι επιτακτική. Αυτό που μπορεί και πρέπει να γίνεται είναι η αφαίρεση κύκλων μικρού μήκους (4, 6, 8 κλπ) ή τουλάχιστον η κατά το δυνατόν ελάττωση τους.

Τελειώνοντας την ανάλυση μας για την σχέση του γράφου Tanner ενός LDPC κώδικα με τις ιδιότητες που αυτός εμφανίζει, αξίζει να αναφέρουμε ότι η αρνητική επιρροή κύκλων μικρού μήκους μειώνεται για αυξανόμενο μέγεθος κώδικα και μειώνεται δραστικά για μεγάλα μεγέθη κώδικα (> 1000 bits).

2.4.2 Ο αλγόριθμος Belief Propagation (BP)

Στη συνέχεια, θα περιγράψουμε τον βασικό αλγόριθμο αποκωδικοποίησης LDPC κωδίκων: τον αλγόριθμο belief-propagation (BP), γνωστό και ως sum-product αλγόριθμο. Στην πλειονότητα τους οι χρησιμοποιούμενοι αλγόριθμοι αποκωδικοποίησης LDPC κωδίκων είναι παραλλαγές ή απλοποιήσεις του βασικού αλγορίθμου που θα παρουσιάσουμε παρακάτω. Στόχος του αλγορίθμου είναι να προσδιορίσει κατά βέλτιστο τρόπο την «εκ των υστέρων» (a posteriori) πιθανότητα κάθε bit να είναι 0 ή 1, γνωρίζοντας το σήμα που έχει λάβει ο δέκτης, τα χαρακτηριστικά του κώδικα που σε αυτή την περίπτωση εκφράζονται ως εξισώσεις ισοτιμίας καθώς και τα χαρακτηριστικά του τηλεπικοινωνιακού διαύλου της διαδρομής από τον πομπό στο δέκτη. Σε αυτό το σημείο να τονίσουμε ότι ο αλγόριθμος BP βρίσκει την καλύτερη εκτίμηση για κάθε bit της κωδικής λέξης που καταφθάνει, αλλά όχι απαραίτητα την καλύτερη εκτίμηση για την κωδική λέξη ως σύνολο. Αυτό είναι συνέπεια του ότι πρόκειται για a-posteriori αλγόριθμο σε αντίθεση με τους αλγόριθμους μεγίστης πιθανοφάνειας, όπως ο αλγόριθμος Viterbi, που βελτιστοποιούν την αποκωδικοποίηση ολόκληρου του λαμβανόμενου κωδικοποιημένου διανύσματος.

Σε πρώτη φάση λοιπόν, ενημερώνονται οι τιμές Qij^x και τίθενται ίσες με τις εκ των προτέρων εκτιμήσεις για τα ληφθέντα σύμβολα (bits). Η πιθανότητα το j-οστό bit να είναι x (x=0 ή 1) συμβολίζεται fj^x. Οι εκφράσεις για τις παραπάνω πιθανότητες εξαρτώνται από το είδος του διαύλου και προκύπτουν από τη μοντελοποίηση που του έχουμε κάνει. Για παράδειγμα, όπως θα δείξουμε και στην προσομοίωση μας, σε περίπτωση διαύλου με προσθετικό λευκό θόρυβο Gauss (AWGN) χρησιμοποιούμε εκφράσεις που βασίζονται σε γκαουσιανές συναρτήσεις πυκνότητας πιθανότητας.

Μετά από τη φάση αρχικοποίησης, αρχίζει η ανταλλαγή πληροφορίας μεταξύ των κόμβων bit και των κόμβων ελέγχου. Χρησιμοποιούμε το συμβολισμό Rij^x για να εκφράσουμε την πιθανότητα να ικανοποιείται η i-οστή εξίσωση ελέγχου ισοτιμίας (hi) δεδομένου ότι ο κόμβος bit j βρίσκεται στην κατάσταση x (dj^x). Αυτή η πιθανότητα προκύπτει από την πληροφορία που στέλνεται από κάθε κόμβο ελέγχου που είναι συνδεδεμένος με τον κόμβο bit j προς τον κόμβο bit j. Τελικά, η πιθανότητα να ικανοποιείται η hi βρίσκεται ως:

$$\mathbf{P}(\mathbf{h_i}|\mathbf{d_j} = \mathbf{x}) = \sum_{\mathbf{d}:\mathbf{d_j}=\mathbf{x}} \mathbf{P}(\mathbf{h_i}|\mathbf{d}) * \mathbf{P}(\mathbf{d}|\mathbf{d_j} = \mathbf{x}) \quad (26)$$

Αυτή η πιθανότητα υπολογίζεται για όλα τα κωδικοποιημένα διανύσματα d για τα οποία ικανοποιείται η i-οστή εξίσωση ελέγχου ισοτιμίας, με την προϋπόθεση ότι ο κόμβος bit j βρίσκεται στην κατάσταση x. Συνοψίζοντας, μπορούμε να πούμε ότι ο BP αλγόριθμος για αποκωδικοποίηση LDPC κωδίκων έχει την παρακάτω γενική περιγραφή:

• **Αρχικοποίηση**:

Υπολογίζονται οι εκ των προτέρων πιθανότητες κατάστασης κάθε bit με βάση τη ληφθείσα τιμή του σήματος που αντιστοιχεί σε κάθε bit και τα χαρακτηριστικά του διαύλου, έστω fj^x . Αυτές οι τιμές χρησιμοποιούνται για την αρχικοποίηση των συντελεστών Qij^x, fj^x. Επιπλέον, καθορίζεται ο μέγιστος δυνατός αριθμός επαναλήψεων του αλγορίθμου αποκωδικοποίησης, έστω max_iter.

• **Επαναληπτική Αποκωδικοποίηση**:

Βήμα 1: Ροή πληροφορίας από κόμβους bit προς κόμβους ελέγχου. Αποστολή προς κόμβους ελέγχου των συντελεστών Qij^x. Ενημέρωση των συντελεστών Rij^x .

$$R_{ij}^{x} = \sum_{\mathbf{d}:\mathbf{d}_j=x} \mathbf{P}(\mathbf{h}_i|\mathbf{d}) * \prod_{k\in N(i)\setminus j} Q_{ik}^{dk} \quad (27)$$

Βήμα 2: Ροή πληροφορίας από κόμβους ελέγχου προς κόμβους bit
Αποστολή προς κόμβους ελέγχου των συντελεστών Rij^x. Ενημέρωση των συντελεστών Qij^x:

$$Q_{ij}^{x} = a_{ij} * f_j^x * \prod_{k\in M(j)\setminus i} R_{kj}^{x} \quad (28)$$

Ο συντελεστής f_j^x αναφέρεται στην εκ των προτέρων πιθανότητα ο κόμβος dj να βρίσκεται στην κατάσταση x, ενώ η σταθερά a_{ij} προκύπτει από την απαίτηση να ισχύει η κανονικοποιημένη συνθήκη (άθροισμα όλων των πιθανοτήτων ίσο με 1):

$$\sum_x Q_{ij}^{x}=1 \quad (29)$$

Ενημέρωση της πιθανότητας κατάστασης κάθε bit. Με βάση το ποια τιμή είναι πιο πιθανή, κάθε bit παίρνει την τιμή 0 ή 1, οπότε παίρνουμε μια εκτίμηση για το όλον διάνυσμα, έστω vector_est, βάση της σχέσης:

$$\hat{d}_j = \mathbf{argmax}_{x} \mathbf{f}_j^x \prod_{k \epsilon M(j)} \mathbf{R}_{kj}^x \quad (30)$$

Βήμα 3: Έλεγχος εγκυρότητας

Σε περίπτωση που ο αριθμός επαναλήψεων φτάσει την τιμή max_iter, ο αλγόριθμος τερματίζεται. Εάν vector_est x HT = 0, ο αλγόριθμος σταματά και το vector_est θεωρείται έγκυρο διάνυσμα, διαφορετικά ο αλγόριθμος αρχίζει μια νέα επανάληψη αρχίζοντας από το Βήμα 1. Ο παρακάτω πίνακας δίνει τους βασικούς συμβολισμούς:

ΣΥΜΒΟΛΟ	ΕΡΜΗΝΕΙΑ
Q_{ij}^x	Πιθανότητα ο κόμβος bit j να είναι στην κατάσταση x, όπως προκύπτει από την πληροφορία που στέλνουν οι συνδεδεμένοι με τον j κόμβοι ελέγχου ισοτιμίας εξαιρουμένου του κόμβου i
R_{ij}^x	Πιθανότητα να ικανοποιείται η i-οστή εξίσωση ελέγχου ισοτιμίας δεδομένου ότι ο κόμβος bit j βρίσκεται στην κατάσταση x
f_j^x	A priori πιθανότητα το j-οστό bit να είναι x

2.4.2.1. Ένα παράδειγμα εκτέλεσης BP αλγορίθμου

Σε αυτή την ενότητα θα παρουσιάσουμε ένα αριθμητικό παράδειγμα που βασίζεται στην παραπάνω παρουσίαση μας όσον αφορά την κωδικοποίηση και αποκωδικοποίηση LDPC κωδίκων. Έτσι, θα γίνει περισσότερο κατανοητή η λειτουργία του BP αλγορίθμου (στη βιβλιογραφία συναντάται και ως sum-product αλγόριθμος). Ο κώδικας μας χαρακτηρίζεται από τους παρακάτω πίνακες H, G:

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Όπως αναφέραμε, πρωταρχική σχεδίαση γίνεται μόνο για τον Η, οπότε ο G είναι απόρροια του Η με διαδικασία που αναφέρεται σε προηγούμενο εδάφιο. Όπως προκύπτει από τις διαστάσεις του Η (8 x 12), κάθε ακωδικοποίητο μήνυμα έχει μήκος m=12-8=4 και κάθε κωδικοποιημένο διάνυσμα έχει μήκος n=12.

Για κάθε τέτοιο κώδικα ισχύει:

$G*H^T = 0$ (31)

$c* H^T = m*G * H^T = 0$ (32)

$c = m*G$ (33)

Συνεπώς, ο ρυθμός κώδικα, ένα πολύ σημαντικό μέγεθος για κάθε σχήμα κωδικοποίησης, είναι 1/3. Παρατηρούμε επίσης ότι ο συγκεκριμένος LDPC κώδικας είναι μη-κανονικός. Αξίζει επίσης να σημειώσουμε ότι οι χρησιμοποιούμενοι στην πράξη πίνακες έχουν πολύ μεγαλύτερο μέγεθος, όμως σε αυτή την ενότητα μας ενδιαφέρει η κατανόηση της λειτουργίας της κωδικοποίησης και της αποκωδικοποίησης και όχι τόσο η αποδοτική σχεδίαση κώδικα που θα μας απασχολήσει στην προσομοίωση μας. Για αυτό το παράδειγμα, θεωρούμε μήνυμα m=(1 0 0 0).

Το m κωδικοποιείται ως c, με βάση τη γνωστή μας σχέση $c = m \cdot G$, επομένως προκύπτει η κωδ. λέξη c=(1 1 1 1 1 0 0 0 1 0 0 0). Το κωδικοποιημένο διάνυσμα ακολούθως διαμορφώνεται προς μετάδοση με BPSK διαμόρφωση, οπότε σε πολική μορφή μεταδίδεται το παρακάτω διάνυσμα: t = (+1 +1 +1 +1 +1 –1 –1 –1 +1 –1 –1 –1).

Θεωρώντας ότι το σήμα διέρχεται από δίαυλο που εισάγει προσθετικό λευκό θόρυβο Gauss τυπικής απόκλισης σ=0.8 και ως αποτέλεσμα της μετάδοσης και της διαδικασίας δειγματοληψίας φτάνει στην είσοδο του αποκωδικοποιητή το ακόλουθο διάνυσμα r:

r=(+1.31 +2.65 +0.74 +2.17 +0.59 -0.83 -0.39 -1.75 +1.49 +0.40 -0.92 +1.07)

Εάν χρησιμοποιούνταν αποκωδικοποιητής σκληρής απόφασης (hard decision), ο οποίος θα αποκωδικοποιούσε όλες τις θετικές τιμές ως ψηφίο «1» και όλες τις αρνητικές τιμές ως ψηφίο «0», τότε το αποκωδικοποιημένο διάνυσμα θα ήταν d = (1 1 1 1 1 0 0 0 1 1 0 1).

Ο δίαυλος επικοινωνίας λοιπόν θα είχε εισάγει δύο λάθη στις θέσεις 10, 12 όπως φαίνεται από τη σύγκριση των διανυσμάτων c, d. Αφού ο δίαυλος μας είναι τύπου AWGN, η έκφραση για την εκ των προτέρων πιθανότητα κατάστασης κάθε bit του ληφθέντος διανύσματος θα δίνεται από την γκαουσιανή συνάρτηση πυκνότητας πιθανότητας. Η πιθανότητα f_j^0 το bit στη θέση j του διανύσματος να είναι 0 και η πιθανότητα f_j^1 το bit στη θέση j του διανύσματος να είναι 1 δίνονται από τις σχέσεις:

$$\mathbf{f_j^0 = \frac{1}{\sqrt{2\pi\sigma}} e^{-(r_j+1)^2/2\sigma^2}} \quad (34)$$

$$\mathbf{f_j^1 = \frac{1}{\sqrt{2\pi\sigma}} e^{-(r_j-1)^2/2\sigma^2}} \quad (35)$$

Όπως φαίνεται από τις παραπάνω σχέσεις, πρέπει να γνωρίζουμε την τυπική απόκλιση σ του διαύλου. Υπάρχουν διάφορες τεχνικές με τις οποίες ο δέκτης μπορεί να κάνει εκτίμηση της τυπικής απόκλισης, όπως εύρεση της παραμέτρου σ μέσω της μεθόδου ελαχιστοποίησης του τετραγωνικού σφάλματος (MMSE). Στη συνέχεια θα θεωρούμε ότι το σύστημα του δέκτη γνωρίζει την τιμή της σταθεράς σ.

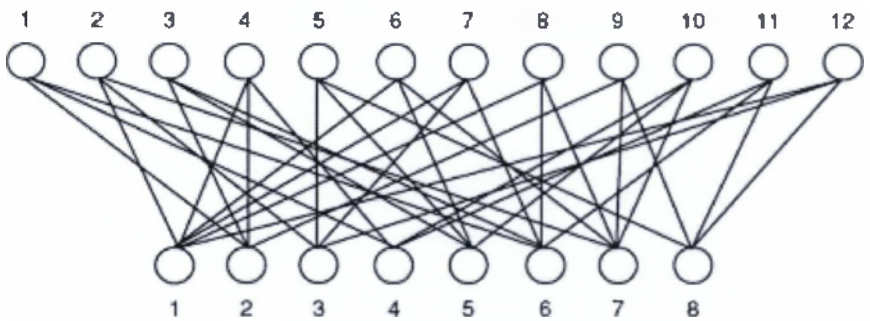
Με βάση τις παραπάνω σχέσεις τώρα για τα f_j^0 και f_j^1, παραθέτουμε τον παρακάτω Πίνακα με τις τιμές τους για j = 1,2,..,12 και σ = 0.8.

j	1	2	3	4	5	6	7	8	9	10	11	12
r	+1.3129	+2.6584	+0.7413	+2.1745	+0.5981	−0.8323	−0.3962	−1.7586	+1.4905	+0.4084	−0.9290	+1.0765
t	+1	+1	+1	+1	+1	−1	−1	−1	+1	−1	−1	−1
f_j^0	0.0076	0.0000	0.0467	0.0002	0.0678	0.4878	0.3751	0.3181	0.0039	0.1059	0.4967	0.0172
f_j^1	0.4619	0.0582	0.4733	0.1697	0.4396	0.0362	0.1088	0.0013	0.4132	0.3794	0.0272	0.4964

Πίνακας 1. Τιμές των fj^0 και fj^1 μετά την αρχικοποίηση του BP αλγορίθμου

Σε αυτό το σημείο έχει ολοκληρωθεί η φάση αρχικοποίησης του αλγορίθμου, η αρχικοποίηση δηλαδή των συντελεστών $Qij^x = f_j^x$ (i=1,...,8 και j=1,...,12)

Ο γράφος Tanner του παραδείγματος αυτού είναι:



Σχήμα 10. Κόμβοι bit (επάνω) και ελέγχου ισοτιμίας/parity check (κάτω)

Γνωρίζοντας ότι σε κάθε κώδικα δομής πρέπει να ισχύει η σχέση $v \cdot H^T = 0$, ώστε να είναι δυνατή η ανίχνευση και διόρθωση σφαλμάτων, παρατηρούμε ότι οι τιμές των ψηφίων της πληροφορίας (*bit ή variable nodes)*, που συνδέονται στον ίδιο κόμβο ελέγχου *(parity check nodes*), πρέπει να δίνουν μηδενικό άθροισμα.

Έτσι προκύπτουν οι εξισώσεις ισοτιμίας του γράφου. Οι εξισώσεις ισοτιμίας του γράφου είναι μία για κάθε κόμβο ισοτιμίας, δηλαδή:

(Η ανά ψηφίο πράξη XOR συμβολίζεται ως $\oplus$)

$C_2 \oplus C_4 \oplus C_6 \oplus C_7 \oplus C_8 \oplus C_{12} = 0$ (36)

$C_1 \oplus C_3 \oplus C_4 \oplus C_9 = 0$ (37)

$C_2 \oplus C_5 \oplus C_7 \oplus C_{12} = 0$ (38)

$C_1 \oplus C_4 \oplus C_{10} \oplus C_{11} = 0$ (39)

$C_3 \oplus C_5 \oplus C_6 \oplus C_{10} = 0$ (40)

$C_1 \oplus C_3 \oplus C_7 \oplus C_8 \oplus C_{11} = 0$ (41)

$C_2 \oplus C_6 \oplus C_8 \oplus C_9 \oplus C_{10} = 0$ (42)

$C_5 \oplus C_9 \oplus C_{11} \oplus C_{12} = 0$ (43)

Η πρώτη εξίσωση προκύπτει επειδή ο κόμβος ελ. ισοτιμίας είναι συνδεμένος με τους bit κόμβους 2, 4, 6, 7, 8,12 και ανάλογα προκύπτουν και οι υπόλοιπες εξισώσεις.

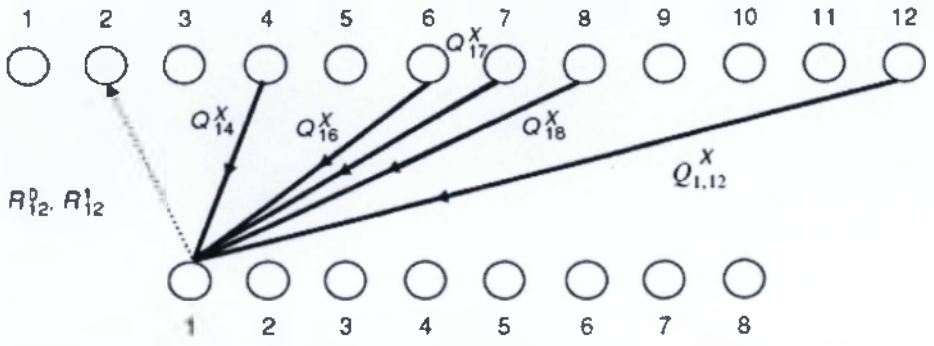
Βήμα 1 (Θυμίζουμε ότι i=1,...,8 και j=1,...,12)

Στο βήμα αυτό γίνεται η ανάθεση τιμών στους συντελεστές Rij^x με βάση τους συντελεστές Qij^x. Κάθε συντελεστής Rij^x εκφράζει την πιθανότητα να ικανοποιείται η εξίσωση ισοτιμίας i δεδομένου ότι ο κόμβος bit j βρίσκεται στην κατάσταση x. Έτσι, στο παράδειγμα μας, η τιμή R_{12}^0 εκφράζει την πιθανοτική εκτίμηση να ικανοποιείται η εξίσωση (1) όταν το bit στη θέση 2 του διανύσματος έχει την τιμή 0. Παρατηρούμε ότι η παραπάνω εξίσωση ικανοποιείται για οποιοδήποτε συνδυασμό των bits c4, c6, c7, c8, c12, στον οποίο ο αριθμός των άσσων είναι άρτιος όπως προκύπτει από τους κανόνες της modulo-2 αριθμητικής. Θα υπολογίσουμε λοιπόν την πιθανότητα R_{12}^0 αθροίζοντας την πιθανότητα να ισχύει κάθε ένας από τους προαναφερθέντες συνδυασμούς.

Για κάθε έναν από αυτούς τους συνδυασμούς, π.χ. τα bits c4, c6, c7, c8, c12 να είναι όλα 0, η πιθανότητα του συνδυασμού ισούται με το γινόμενο των πιθανοτήτων κάθε bit να είναι στην κατάσταση που εξετάζουμε λόγω ανεξαρτησίας. Όπως ήδη ξέρουμε, η πιθανότητα το bit j να είναι στην κατάσταση x εκφράζεται μέσω του συντελεστή Qij^x. Είμαστε σε θέση τώρα να υπολογίσουμε την τιμή R_{12}^0. Με παρόμοια ανάλυση προκύπτει ότι η τιμή R_{12}^1 προκύπτει για όλους τους συνδυασμούς με περιττό αριθμό άσσων. Στη συνέχεια αποδίδουμε π.χ. την ανταλλαγή πληροφορίας που συντελείται για τον υπολογισμό του R_{12}^0:

$R_{12}^0 = Q_{1,4}^0 Q_{1,6}^0 Q_{1,7}^0 Q_{1,8}^0 Q_{1,12}^0 + Q_{1,4}^0 Q_{1,6}^0 Q_{1,7}^0 Q_{1,8}^1 Q_{1,12}^1 + Q_{1,4}^0 Q_{1,6}^0 Q_{1,7}^1 Q_{1,8}^0 Q_{1,12}^1 + Q_{1,4}^0 Q_{1,6}^0 Q_{1,7}^1 Q_{1,8}^1 Q_{1,12}^0 + Q_{1,4}^0 Q_{1,6}^1 Q_{1,7}^0 Q_{1,8}^0 Q_{1,12}^1 + Q_{1,4}^0 Q_{1,6}^1 Q_{1,7}^0 Q_{1,8}^1 Q_{1,12}^0 + Q_{1,4}^0 Q_{1,6}^1 Q_{1,7}^1 Q_{1,8}^0 Q_{1,12}^0 + Q_{1,4}^0 Q_{1,6}^1 Q_{1,7}^1 Q_{1,8}^1 Q_{1,12}^1 + Q_{1,4}^1 Q_{1,6}^0 Q_{1,7}^0 Q_{1,8}^0 Q_{1,12}^1 + Q_{1,4}^1 Q_{1,6}^0 Q_{1,7}^0 Q_{1,8}^1 Q_{1,12}^0 + Q_{1,4}^1 Q_{1,6}^0 Q_{1,7}^1 Q_{1,8}^0 Q_{1,12}^0 + Q_{1,4}^1 Q_{1,6}^0 Q_{1,7}^1 Q_{1,8}^1 Q_{1,12}^1 + Q_{1,4}^1 Q_{1,6}^1 Q_{1,7}^0 Q_{1,8}^0 Q_{1,12}^0 + Q_{1,4}^1 Q_{1,6}^1 Q_{1,7}^0 Q_{1,8}^1 Q_{1,12}^1 + Q_{1,4}^1 Q_{1,6}^1 Q_{1,7}^1 Q_{1,8}^0 Q_{1,12}^1 + Q_{1,4}^1 Q_{1,6}^1 Q_{1,7}^1 Q_{1,8}^1 Q_{1,12}^0$.

Στο παρακάτω σχήμα δίνεται σχηματικά η ροή πληροφορίας κατά τον υπολογισμό των R_{12}^0, R_{12}^1



Σχήμα 11. Ροή πληροφορίας κατά τον υπολογισμό των τον υπολογισμό των R_{12}^{0}, R_{12}^{1}

Βήμα 2: Μετά τον υπολογισμό όλων των συντελεστών $R_{i,j}^{0}$, $R_{i,j}^{1}$ μπορούμε να υπολογίσουμε την πρώτη εκτίμηση για το διάνυσμα d. Αυτό γίνεται με βάση τον τύπο

Για παράδειγμα για το bit 1 παίρνουμε

$$\hat{d}_1 = \begin{Bmatrix} \hat{0} : f_1^0 * R_{21}^0 * R_{41}^0 * R_{61}^0 \\ \hat{1} : f_1^1 * R_{21}^1 * R_{41}^1 * R_{61}^1 \end{Bmatrix} \quad (44)$$

και όποια πιθανότητα είναι μεγαλύτερη, αυτή αποφασίζει για το αν το bit θα είναι «0» ή «1».

Κάνοντας την ίδια διαδικασία για όλα τα bits, προκύπτει η λέξη d=(1 1 1 1 0 1 1 0 1 0 0 0).

Υπολογίζονται οι νέοι συντελεστές Qij^x από την σχέση

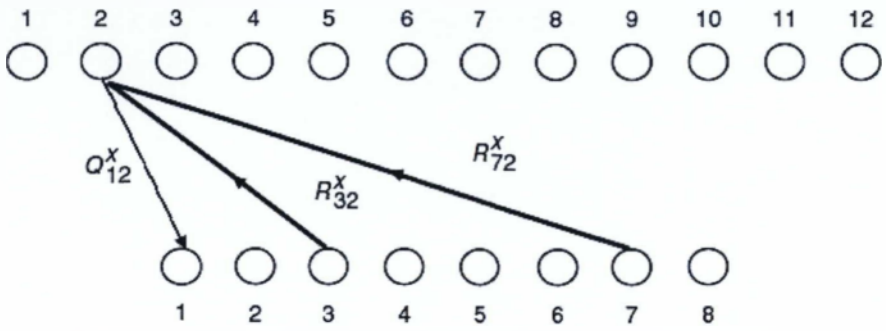
$$Q_{ij}^x = a_{ij} * f_j^x * \prod_{k \in M(j) \setminus i} R_{kj}^x \quad (45)$$

Για παράδειγμα

$$Q_{12}^0 = a_{12} * f_2^0 * R_{32}^0 * R_{72}^0 \text{ και } Q_{12}^1 = a_{12} * f_2^1 * R_{32}^1 * R_{72}^1 \quad (46)$$

Η σταθέρα a_{12} θα πρέπει να ικανοποιεί την σχέση $Q_{12}^0 + Q_{12}^1 = 1$ επομένως

$$a_{12} = \frac{1}{f_2^0 * R_{32}^0 * R_{72}^0 + f_2^1 * R_{32}^1 * R_{72}^1} \quad (47)$$



Σχήμα 12. Ροή πληροφορίας κατά τον υπολογισμό των τον υπολογισμό των Q_{12}^0, Q_{12}^1

Βήμα 3: (Έλεγχος εγκυρότητας και τερματισμού) Παρατηρούμε ότι η αποκωδ. λέξη d διαφέρει σε 3 θέσεις από την λέξη c=(1 1 1 1 1 0 0 0 1 0 0 0) που στάλθηκε από τον κωδικοποιητή. Ο έλεγχος εγκυρότητας είναι ο υπολογισμός του συνδρόμου που δίνεται από την σχέση:

$$\mathbf{d * H^T = 0} \quad (48)$$

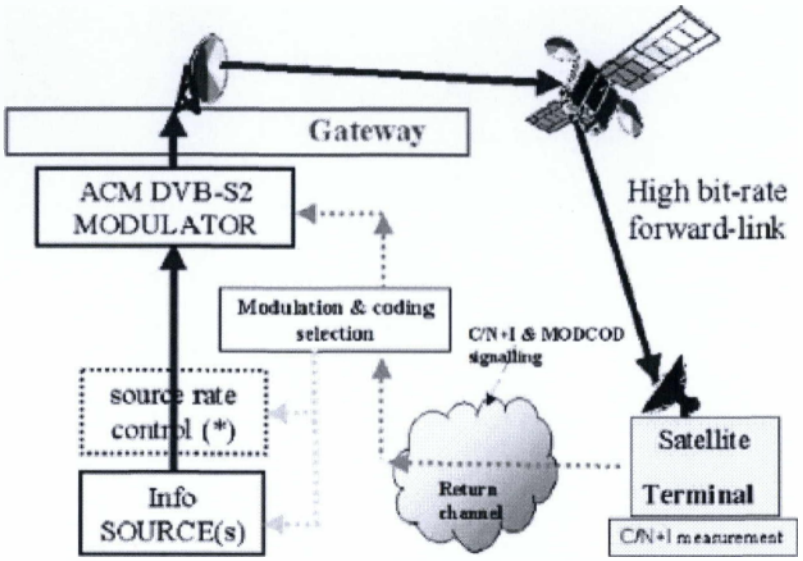
Η παραπάνω σχέση ικανοποιείται για κάθε αποκωδικοποιούμενη λέξη που ανήκει στις έγκυρες λέξεις. Επομένως αυτή η σχέση δεν θα ικανοποιείται στην $1^η$ επανάληψη και ο αλγόριθμος πηγαίνει στο Βήμα 1 για την $2^η$ επανάληψη.

Στην $2^{η}$ επανάληψη, σε αντίθεση με την $1^{η}$ επανάληψη, οι συντελεστές Qij^{x} δεν εμπεριέχουν μόνο πληροφορία διαύλου αλλά και πληροφορία κώδικα. Αυτό το χαρακτηριστικό προφανώς θα ενυπάρχει και στις επόμενες επαναλήψεις. Προχωρώντας στα βήματα της δεύτερης επανάληψης, υπολογίζουμε, μέσω των γενικών σχέσεων που ισχύουν κάθε φορά, τις τιμές των συντελεστών Rij^{0}, Rij^{1}, Qij^{0}, Qij^{1}.
Στην συνέχεια υπολογίζονται τα εκτιμώμενο αποκωδικοποιημένο διάνυσμα στο τέλος της δεύτερης απανάληψης είναι d=(1 1 1 1 1 0 0 0 1 0 0 1) που διαφέρει από το αρχικό κωδικοποιημένο διάνυσμα c=(1 1 1 1 1 0 0 0 1 0 0 0) μόνο στο τελευταίο bit. Αυτή η διαφορά προκαλεί μη μηδενικό σύνδρομο για το d, οπότε μια νέα επανάληψη αρχίζει, η τρίτη κατά σειρά.

Κατά την τρίτη λοιπόν επανάληψη ο αλγόριθμος BP καταφέρνει να διορθώσει όλα τα σφάλματα που είχε εισάγει ο δίαυλος και να δώσει το μήνυμα d=(1 1 1 1 1 0 0 0 1 0 0 0)=c. Ο αποκωδικοποιητής τότε οδηγεί στην έξοδο το αποκωδικοποιημένο μήνυμα m=(1 0 0 0), που ταυτίζεται με το αρχικό μήνυμα πληροφορίας.

2.5 LDPC κώδικες για το DVB-S2

Το DVB-S2 αποτελεί τη δεύτερη γενιά του, ιδιαίτερα διαδεδομένου, προτύπου DVB-S (Digital Video Broadcasting-Satellite). Το νέο αυτό πρότυπο έχει σχεδιασθεί για να εξυπηρετεί μία μεγάλη ποικιλία ευρυζωνικών υπηρεσιών και εφαρμογών, όπως εμπορική τηλεόραση, υψηλής ευκρίνειας τηλεόραση (HDTV), διαδραστικές υπηρεσίες (Internet), DSNG (Digital Satellite News Gathering), διανομή TV σε επίγειους πομπούς VHF/UHF, διανομή περιεχομένου (content delivery) και υπηρεσίες κορμού Internet. Ένα γενικό λειτουργικό διάγραμμα δίνεται στο σχήμα που ακολουθεί.



Σχήμα 13. DVB-S2

Όσον αφορά την κωδικοποίηση, να επισημάνουμε αρχικά ότι σε συστήματα ευρυεκπομπής (broadcasting) όπως το DVB-S2 η τεχνική ARQ απορρίπτεται για πρακτικούς λόγους. Έτσι, τέτοια συστήματα χρησιμοποιούν υποχρεωτικά πρόσθια διόρθωση λαθών (FEC). Μάλιστα, λόγω των ιδιαίτερων απαιτήσεων του DVB-S2 αλλά και λόγω των σημαντικών απωλειών που εισάγει ο δορυφορικός δίαυλος η επίδοση του χρησιμοποιούμενου κώδικα διόρθωση λαθών πρέπει να είναι εξαιρετική.

Μετά απο διεξοδικές συγκρίσεις, μέσω προσομοιώσεων, μεταξύ αλυσιδωτών συνελικτικών κωδίκων, Turbo κωδίκων και LDPC κωδίκων επελέγησαν τελικά οι LDPC κώδικες. Φαίνεται λοιπόν και ο πρωτεύον ρόλος της προσομοίωσης στα σύγχρονα τηλεπικοινωνιακά συστήματα. Ο κώδικας που τελικά υιοθετήθηκε έχει μήκος κωδικής λέξης n=64800 bits και αυτό γιατί όπως γνωρίζουμε οι εξαιρετικές επιδόσεις των LDPC κωδίκων επιτυγχάνονται για πολύ μεγάλες τιμές του μήκους της κωδικής λέξης. Παράλληλα, υπάρχει η δυνατότητα επιλογής από το σύστημα

εναλλακτικού LDPC κώδικα με μήκος n=16200 bits. Αυτό το μικρότερο μέγεθος κώδικα συνεπάγεται και μικρότερη καθυστέρηση κατά τις διαδικασίες κωδικοποίησης / αποκωδικοποίησης και επιλέγεται για εφαρμογές πραγματικού χρόνου, δηλαδή για εφαρμογές ευαίσθητες ως προς την καθυστέρηση. Για τις υπόλοιπες εφαρμογές χρησιμοποιείται το μπλοκ των 64800 bits που επιτυγχάνει συγκριτικά καλύτερες επιδόσεις για ίδιο σηματοθορυβικό λόγο. Επιπλέον, για πληρέστερη περιγραφή της μονάδας FEC του προτύπου DVB-S2 να σημειώσουμε ότι χρησιμοποιείται και εξωτερικός κώδικας (outer code) BCH για τη διασφάλιση κατώτατου ορίου σφαλμάτων . Σαν αποτέλεσμα των παραπάνω, με τη χρήση των LDPC κωδίκων έγινε εφικτή η αύξηση της επίδοσης κατά 35% σε σύγκριση με το προϋπάρχον DVB-S1 όπου χρησιμοποιούνταν συνελικτικοί και Reed-Solomon (RS) κώδικες.

Συνοψίζοντας, μπορούμε να πούμε ότι βασικοί στόχοι του νέου αυτού προτύπου ήταν:

- Βέλτιστη επίδοση
- Πλήρης ευελιξία
- Ανεκτή πολυπλοκότητα δέκτη

2.6 Πλεονεκτήματα και Μειονεκτήματα

Όπως θα δούμε και στη συνέχεια, ο πίνακας ελέγχου ισοτιμίας καθορίζει το κύκλωμα και κατά συνέπεια την πολυπλοκότητα του κωδικοποιητή. Πιο συγκεκριμένα, η πολυπλοκότητα είναι ανάλογη των μη μηδενικών στοιχείων του πίνακα ισοτιμίας. Όσο λιγότεροι, επομένως, είναι οι άσσοι του πίνακα ισοτιμίας, τόσο απλούστερος ο κωδικοποιητής, γεγονός που φανερώνει το σημαντικό πλεονέκτημα των κωδικών LDPC ως προς την υλοποίηση, έναντι άλλων μπλοκ κωδικών ίδιου μεγέθους. Ένα

ακόμα σημαντικό πλεονέκτημα των κωδίκων LDPC είναι το γεγονός πως οι χρησιμοποιούμενοι αλγόριθμοι αποκωδικοποίησης, πέρα από τη διόρθωση σφαλμάτων, προσφέρουν σαφή και έγκυρη ανίχνευση των περιπτώσεων εκείνων που η αποκωδικοποίηση αποτυγχάνει. Ενώ θεωρητικά είναι πιθανό, με κατάλληλη επιλογή κώδικα, μπορούμε να ελαττώσουμε κατά πολύ την πιθανότητα η αποκωδικοποίηση να καταλήξει σε κωδική λέξη η ο οποία να είναι έγκυρη, χωρίς όμως να είναι εκείνη που μεταδόθηκε.

Ένα εξαιρετικής σημασίας πλεονέκτημα των κωδίκων LDPC είναι οι πολύ καλές επιδόσεις τους. Ενώ μέχρι πριν λίγα χρόνια οι κώδικες Turbo ήταν εκείνοι οι οποίοι κατείχαν τα πρωτεία όσον αφορά τις επιδόσεις, πλέον οι κώδικες LDPC φαίνεται να τους ανταγωνίζονται επάξια. Υποστηρίζουν ρυθμούς μετάδοσης, οι οποίοι πλησιάζουν τη χωρητικότητα καναλιού (όριο Shannon).

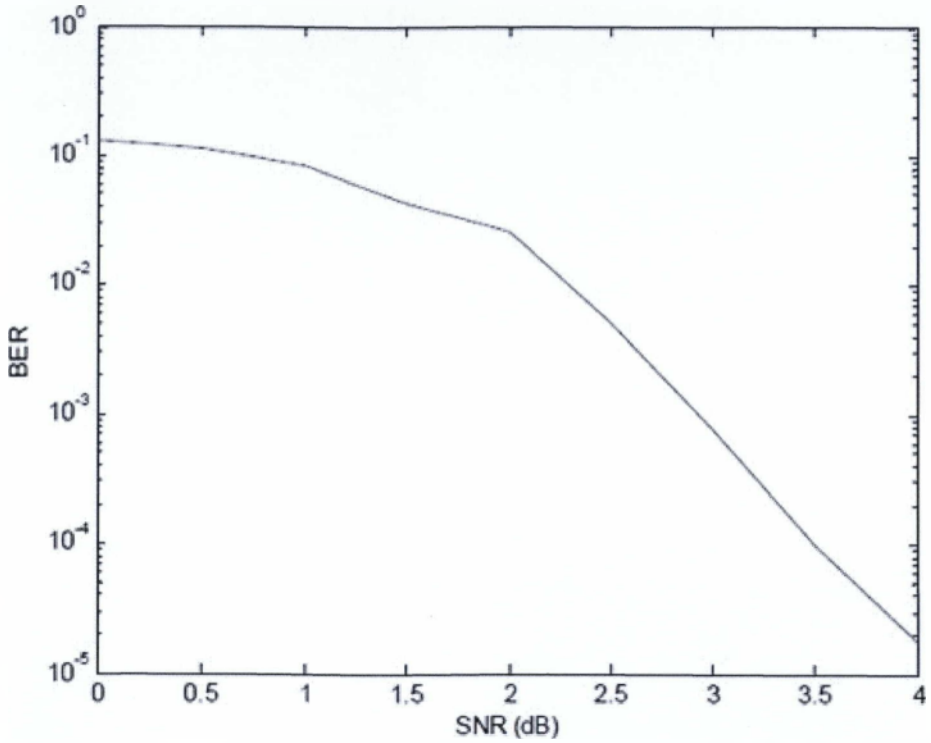
Το βασικό αρνητικό στοιχείο είναι πως για να επιτευχθεί αυτός ο ρυθμός μετάδοσης πρέπει να χρησιμοποιηθεί κώδικας πολύ μεγάλου μήκους. Αυτό σημαίνει πως απαιτείται πολύ μεγάλος πίνακας ελέγχου ισοτιμίας. Όσο αραιός και να είναι ένας τόσο μεγάλος πίνακας, ο αριθμός των μη μηδενικών του στοιχείων είναι εξαιρετικά μεγάλος, με αποτέλεσμα η διαδικασία κωδικοποίησης αλλά και αποκωδικοποίησης να είναι εξαιρετικά απαιτητική από άποψη πολυπλοκότητας, ιδιαίτερα στον κωδικοποιητή όπου απαιτείται 'κατά κάποιο τρόπο' η αποθήκευση του γεννήτορα πίνακα.

ΚΕΦΑΛΑΙΟ 3 LDPC ΓΙΑ ΤΟ AWGN

3.1 SNR απόδοση του κώδικα LDPC (256,128) στο AWGN

Πρώτο βήμα, αλλά πολύ ουσιώδες, για την εργασία μας ήταν η κατασκευή του πίνακα ελέγχου ισοτιμίας H. Οι διαστάσεις του H είναι 128 x 256, κατά συνέπεια ο κώδικας μας είναι LDPC (256, 128). Ο πίνακας είναι **κανονικός (regular)** έχοντας 6 άσσους ανά γραμμή και 3 άσσους ανά στήλη. Όλα τα αποτελέσματα που θα ακολουθήσουν αφορούν τον ίδιο πίνακα. Ο μέγιστος αριθμός επαναλήψεων είναι 80 (max_iter=80).

Η γραφική παράσταση δίνεται στο σχήμα 13. Αναλογιζόμενοι το μέγεθος του κώδικα πρόκειται για αρκετά καλή επίδοση. Σαφώς, έχουν κατασκευαστεί πολύ ισχυρότεροι LDPC κώδικες που όμως έχουν και πολύ μεγάλα μεγέθη κώδικα. Χαρακτηριστικά αναφέρουμε ότι το μήκος της κωδικής λέξης του LDPC κώδικα που έχει υιοθετηθεί στο πρότυπο DVBS2 είναι 64800 bits. Το τίμημα για τέτοια μεγέθη κώδικα είναι αυξημένες απαιτήσεις αποθηκευτικού χώρου, καθώς μιλάμε για πίνακες ελέγχου ισοτιμίας μεγάλων διαστάσεων, και αύξηση της πολυπλοκότητας τόσο της κωδικοποίησης όσο και της αποκωδικοποίησης. Με την αύξηση του σηματοθορυβικού λόγου παρατηρούμε και μείωση του ρυθμού λαθών. Σαν αριθμητικό παράδειγμα, βλέπουμε ότι για SNR 3.5 dB ο ρυθμός λαθών ανά ψηφίο είναι περίπου 10-4. Με άλλα λόγια, για αυτή την τιμή σηματοθορυβικού λόγου εμφανίστηκε ένα λάθος ανά 10000 ψηφία (bits) ή, αλλιώς, η εκτίμηση της προσομοίωσης μας για την πιθανότητα λάθους είναι 10-4. Ο κατακόρυφος άξονας (BER) δίνεται σε λογαριθμική κλίμακα.



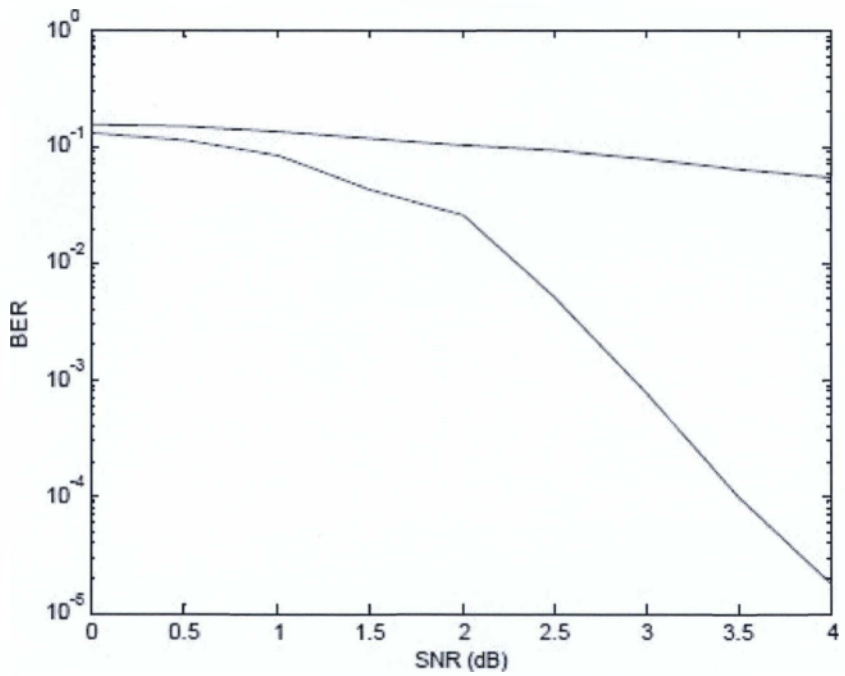
Σχήμα 13. Επίδοση του LDPC (256,128) για το AWGN και διαμόρφωση BPSK

3.2 SNR κέρδος από τη χρήση κωδικοποίησης

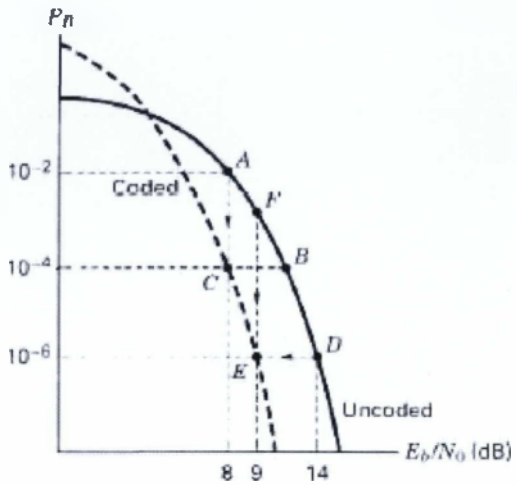
Το σενάριο προς προσομοίωση είναι η μελέτη και σύγκριση συστήματος που κάνει χρήση της κωδικοποίησης μας με σύστημα που δεν χρησιμοποιεί κωδικοποίηση (uncoded). Για την εγκυρότητα της σύγκρισης, και τα δύο συστήματα πρέπει να χρησιμοποιούν την ίδια τεχνική διαμόρφωσης και ο τηλεπικοινωνιακός δίαυλος να είναι ίδιου τύπου. Όπως

προαναφέρθηκε, προς το παρόν εξετάζουμε μόνο BPSK διαμόρφωση και δίαυλο AWGN.

Το σχήμα 14 δείχνει αυτή τη σύγκριση και μας δίνει τη δυνατότητα για εξαγωγή χρήσιμων συμπερασμάτων. Βασικά, σημειώνουμε το ολοένα και αυξανόμενο κέρδος από τη χρήση κωδικοποίησης με την αύξηση του σηματοθορυβικού λόγου. Για μικρές τιμές SNR η παρουσία του θορύβου σε σχέση με το σήμα μας είναι ιδιαίτερα έντονη. Κατά συνέπεια, η μεταδιδόμενη κωδική λέξη αλλοιώνεται σημαντικά δυσχεραίνοντας κατά πολύ το έργο του αποκωδικοποιητή για διόρθωση λαθών και έτσι μέχρι SNR ίσο με 1 dB δεν υπάρχει σημαντική βελτίωση της επίδοσης. Μάλιστα, από τη γενική και θεωρητική καμπύλη του σχήματος 15 προβλέπεται ακόμα και χειροτέρευση της επίδοσης με τη χρήση κωδικοποίησης για ακόμα μικρότερες τιμές του σηματοθορυβικού λόγου.



Σχήμα 14. Επίδοση του LDPC (256,128) για το AWGN (πράσινη γραμμή) σε σχέση με uncoded (μπλε γραμμή)



Σχήμα 15. Θεωρητική επίδοση με και χωρίς κωδικοποίηση σύμφωνα με την βιβλιογραφία

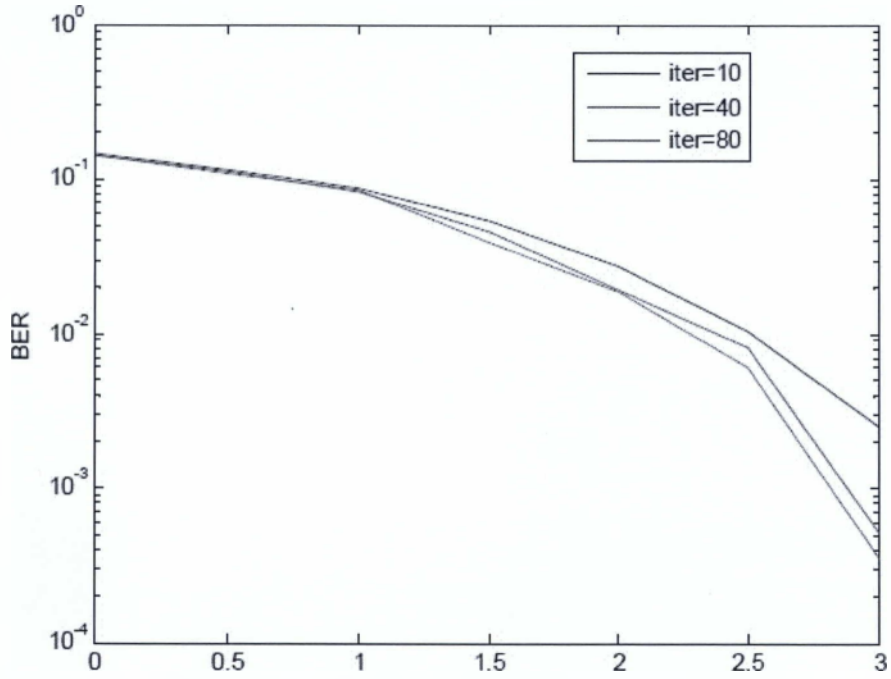
Βλέπουμε ότι για ικανοποιητικές τιμές SNR η κωδική λέξη καταφθάνει αλλοιωμένη δίνοντας τη δυνατότητα στον αποκωδικοποιητή να κάνει διόρθωση λαθών. Έτσι, παρατηρούμε κέρδος από τη χρήση κωδικοποίησης που με την αύξηση του σηματοθορυβικού λόγου γίνεται τεράστιο. Για παράδειγμα, η τιμή του BER για 4 dB που προέκυψε από την προσομοίωση είναι 0.0552 χωρίς χρήση κωδικοποίησης και 0.000017188 με χρήση κωδικοποίησης. Δηλαδή, με χρήση κωδικοποίησης καταφέραμε να έχουμε ρυθμό λαθών περίπου 3200 φορές μικρότερο από ότι χωρίς χρήση κωδικοποίησης.

3.3 SNR απόδοση του κώδικα LDPC (256,128) για διαφορετικό αριθμό επαναλήψεων

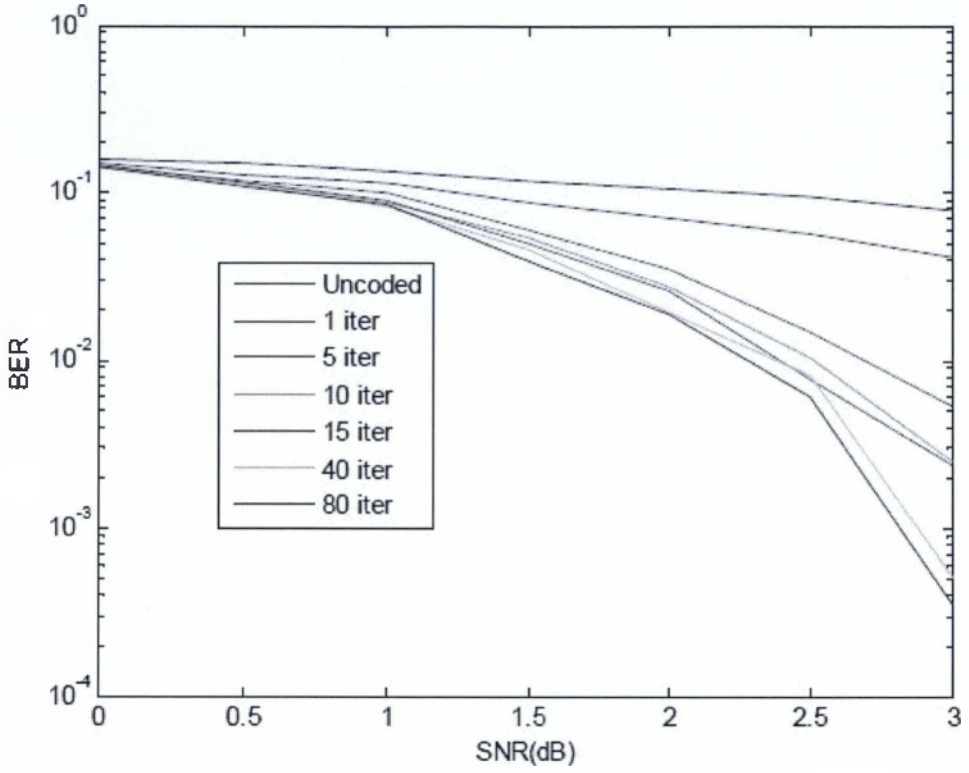
Θέλουμε να συγκρίνουμε την επίδοση του για διάφορες τιμές του μέγιστου αριθμού επαναλήψεων του επαναληπτικού αλγορίθμου αποκωδικοποίησης. Βεβαίως, για να γίνει κάτι τέτοιο έπρεπε να τροποποιηθεί ανάλογα ο εκτελέσιμος κώδικας των προηγούμενων προσομοιώσεων.

Το πρώτο αποτέλεσμα φαίνεται στο σχήμα 16 όπου φαίνεται η επίδοση του LDPC κώδικα για 10, 40 και 80 επαναλήψεις. Όπως αναμενόταν, μεγαλύτερος αριθμός επαναλήψεων οδηγεί σε καλύτερη επίδοση. Όμως, πρέπει να επισημάνουμε ότι πρώτον για χαμηλές τιμές SNR δεν υπάρχει ουσιαστική διαφοροποίηση και δεύτερον ότι οι 40 και οι 80 επαναλήψεις δεν διαφέρουν πολύ ως προς την επίδοση. Το πρώτο συμπέρασμα αιτιολογείται, όπως και προηγουμένως, από τη μεγάλη αλλοίωση που προκαλεί στην κωδική λέξη η μικρή τιμή σηματοθορυβικού λόγου. Έτσι, αν η αλλοίωση είναι μεγάλη, ξεφεύγουμε από τις διορθωτικές ικανότητες του κώδικα και οι περισσότερες επαναλήψεις δεν μπορούν να προσφέρουν ουσιαστικό έργο κάτι που δεν ισχύει για ικανοποιητικές τιμές σηματοθορυβικού λόγου. Στο Σχήμα 17 παρουσιάζουμε τη μελέτη επίδοσης για 1, 5, 10, 15, 40 και 80 επαναλήψεις καθώς και την επίδοση χωρίς χρήση κωδικοποίησης. Πάνω από έναν αριθμό επαναλήψεων δεν επιτυγχάνουμε σημαντική βελτίωση. Από αυτό το σχήμα βλέπουμε ότι το καλύτερο είναι αριθμό επαναλήψεων ίσο με 80. Σε αυτό τον LDPC κώδικα δηλαδή, ακόμα και 1000 επαναλήψεις να κάναμε δεν θα παρατηρούσαμε σημαντική βελτίωση σε σχέση με τις 80 επαναλήψεις. Είναι προφανές ότι ο αριθμός 80 επαναλήψεων δεν αποτελεί γενική συνταγή για τους LDPC κώδικες αφού προβλέπεται διαφορετική συμπεριφορά ανάλογα με το

μέγεθος κώδικα αλλά και με την κατασκευή του πίνακα ελέγχου ισοτιμίας. Παράλληλα, δίνουμε και τη δυνατότητα σύγκρισης και με σύστημα χωρίς κωδικοποίηση, παρατηρώντας ότι το σύστημα με κωδικοποίηση είναι καλύτερο ακόμα και με μία μόνο επανάληψη.



Σχήμα 16. Απόδοση του κώδικα για 10,40,80 επαναλήψεις



Σχήμα 17. Απόδοση του κώδικα για 1,5,10,15,40,80 επαναλήψεις

ΚΕΦΑΛΑΙΟ 4 ΣΥΜΠΕΡΑΣΜΑΤΑ

Βασικός στόχος της παρούσας εργασίας ήταν η μελέτη και προσομοίωση LDPC κωδίκων με σκοπό να εξαχθούν ποιοτικά αλλά και ποσοτικά συμπεράσματα για την συμπεριφορά τους. Κάτι τέτοιο έχει ιδιαίτερη αξία καθώς οι LDPC κώδικες χαρακτηρίζονται ως κώδικες αιχμής και βρίσκουν πρακτική εφαρμογή σε ολοένα και περισσότερα τηλεπικοινωνιακά συστήματα.

Συνοψίζοντας τη συμβολή της εργασίας μπορούμε να αναφερθούμε στα παρακάτω στοιχεία:

• Επισκόπηση των τεχνικών κωδικοποίησης καναλιού με έμφαση στην παρουσίαση των γραμμικών συμπαγών κωδίκων, κατηγορία στην οποία ανήκουν οι LDPC κώδικες. Ανάλυση των δυνατοτήτων των κωδίκων όσον αφορά τον εντοπισμό και τη διόρθωση λαθών και παρουσίαση τρόπων αξιολόγησης τεχνικών κωδικοποίησης.

• Μελέτη των LDPC κωδίκων και ανάλυση των αλγορίθμων αποκωδικοποίησης τους.

• Σχεδίαση και προσομοίωση ενός LDPC (256, 128) κώδικα που οδήγησε στην εξαγωγή πολύ χρήσιμων συμπερασμάτων σχετικά με τις ιδιότητες του. Εκτός από τις ιδιότητες του κώδικα, τα αποτελέσματα της προσομοίωσης μας επέτρεψαν να εξάγουμε συμπεράσματα σχετικά με τις διάφορες τεχνικές διαμόρφωσης και τους διάφορους τύπους διαύλου.

Βιβλιογραφία

[1] Claude E. Shannon, "A Mathematical Theory of Communications", The Bell System Techical Journal, 1948

[2] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error – Correcting Coding and Decoding: Turbo Codes", Proceedings of International Conference in Communications, May 1993

[3] "Channel Coding in Communication Networks: From Theory to Turbocodes", p. 272, edited by A. Glavieux, ISTE Ltd., 2007

[4] Lance C. Perez, Jan Seghers, Daniel J. Costello, Jr., "A Distance Spectrum Interpretation of Turbo Codes", IEEE Transactions on Information Theory, April 1996

[5] S. Dolinar, and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations", TDA Progress Report, August 1995

[6] Stewart N. Crozier, "New High – Spread High Distance Interleavers for Turbo-Codes", Proceedings of the 20th Biennial Symposium on Communications, May 2000

[7] S. Benedetto, and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes", IEEE Transactions on InformationTheory, March 1996

[8] Radford M. Neal, Sparse Matrix Methods and Probabilistic Inference Algorithms¡", IMA Program on Codes, Systems and Graphical Models, 1999

[9] Robert J. McEliece, Eugene R. Rodemich, and Jung-Fu Cheng, "The Turbo Decision Algorithm", 33rd

[10] M. Tanner, On Graph Constructions for LDPC Codes by Quasi- Cyclic Extension, in Information, Coding and Mathematics (M. Blaum, P. Farrell, and H. van Tilborg, eds.), pp. 209¡V220, Kluwer, June 2002.

[11] R. Garello, "The Turbo Code Minimum Distance Algorithm: New Optimization Techniques and Applications", to be submitted to IEEE Transactions on Communications

[12] E. Rosnes, and A. Ytrehus, "Improved Algorithms for the Determination

of Turbo-Code Weight Distributions", IEEE Transactions on Communications, January 2005

[13] Konstantinos V. Koutsouvelis, and Christos E. Dimakis, "A Low Complexity Algorithm for Generating Turbo Code S-Random Interleavers", Wireless Personal Communications, August 2008

[14] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", IEEE Transactions on Information Theory, January 1982

[15] R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA: MIT Press, 1963.

[16] MacKay, D. J. C. and Neal, R. M., "Near Shannon limit performance of low density parity check codes," Electron. Lett., vol. 33, no. 6, March 13, 1997.

Παράρτημα: Κώδικας Matlab

Στο παράρτημα της εργασίας παραθέτουμε στοιχεία από τον πηγαίο κώδικα Matlab η εκτέλεση του οποίου μας έδωσε τα αποτελέσματα της προσομοίωσης. Το βασικό αρχείο της εφαρμογής είναι τα **ldpc_main.m**. Παράλληλα, παρατίθενται και όλα τα **.m** αρχεία που καλούνται.

Η αλγοριθμική διαδικασία που ακολουθείται για την ανάπτυξη προσομοιωτή που μελετά την επίδοση του LDPC κώδικα δίνεται σε γενικές γραμμές παρακάτω υπό τη μορφή ψευδοκώδικα.

Ψευδοκώδικας

Load H % Φόρτωση του πίνακα ελέγχου ισοτιμίας
Get_G_from_H % Δημιουργία, με βάση τον H,του γεννήτορα πίνακα
SNRRange=[0:4:0.5] % Εύρος τιμών SNR, εδώ 0 έως 4 με βήμα 0.5
Packets_per_SNR=1000 % Προσομοιούμενα πακέτα ανά SNR
for SNRrange_index=1:length(SNRrange) % για κάθε τιμή του SNR
for packetnumber=1:Packets_per_SNR % για κάθε πακέτο (κωδική λέξη)
m=random_input_message %τυχαίο δυαδικό μήνυμα πληροφορίας
c=encode_LDPC(m, G) % κωδικοποιημένο διάνυσμα
t=modulate(c, modulation)_type) % διαδικασία διαμόρφωσης
r=after_channel(t, channel_type) % ληφθέν διάνυσμα ύστερα από τη μετάδοση
m_after_decoding=decode_LDPC(r, H) % μήνυμα πληροφορίας ύστερα από τη διαδικασία αποκωδικοποίησης
errors_for_this_SNR(packetnumber)=compare(m, m_after_decoding) % υπολογισμός λαθών για το συγκεκριμένο πακέτο (εσωτερικό for loop) και για τον τρέχοντα SNR(εξωτερικό for loop) συγκρίνοντας το αρχικό

μήνυμα πληροφορίας με το μήνυμα πληροφορίας ύστερα από τη διαδικασία αποκωδικοποίησης

end % τερματισμός εσωτερικού βρόχου

mean_error_rate(SNRrange_index)=mean(errors_for_this_SNR)

%εύρεση της μέσης τιμών σφαλμάτων για την τρέχουσα τιμή SNR

end % τερματισμός εξωτερικού βρόχου

plot(SNRrange, mean_error_rate); % γραφική παράσταση της επίδοσης, ο πίνακας **mean_error_rate** περιέχει τις μέσες τιμές σφαλμάτων ανά SNR

Στη συνέχεια, παραθέτουμε τον πηγαίο κώδικα της προσομοίωσης.

Πηγαίος Κώδικας Matlab

Αρχείο ldpc_main.m

```
% LDPC Code
clear all;
%load parity check matrix H
load 128x256regular.mat H
[rowsh,colsh]=size(H);
n=colsh ; % codeword length
k=n-rowsh ; %message length
A=H(1:n-k,1:n-k);
B=H(1:n-k,n-k+1:n) ;%H=[A B]
I=eye(k); %identity matrix
%I=eye(n-k);
load inverse_of_A.mat invA %inverse of A. over the binary field
p_aux=bin_multi(invA,B);
P=p_aux.';
%get the corresponding generator matrix G
G=[P I];
%run various simulations for various SNR (or Eb/No) values
number_of_Packets_per_SNR=1000;
SNRrange=[0:0.5:4];
mean_error_rate=zeros(1, length(SNRrange));
for SNRrange_index=1:length(SNRrange)
if (SNRrange(SNRrange_index)>3.5)
number_of_Packets_per_SNR=5000;
end
error_rate=zeros(1, number_of_Packets_per_SNR);
for packetnumber=1:number_of_Packets_per_SNR
```

```
m=randint(1,k); % random input message
c=bin_multi(m,G); %codeword production
t=bpsk(c); %transmitted vector after BPSK mod
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNR=SNRrange(SNRrange_index);
No=10^(-SNR/10);
sigma=sqrt(No/2);
%sprintf('SNR: %.4f dB \n',SNR)
r=awgn(t,SNR); %Gaussian noise
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
129
%decoding
%init
max_iter=80;
count_iter=0;
f1=1./(1+exp(-2*r/sigma^2));
f0=1-f1;
fnd=find(H==1);
[ii,jj]=ind2sub([rowsh,colsh],fnd);
len=length(ii);
%Mackay
Q0=zeros(rowsh,colsh);
Q1=zeros(rowsh,colsh);
dQ=zeros(rowsh,colsh);
dR=zeros(rowsh,colsh);
R0=zeros(rowsh,colsh);
R1=zeros(rowsh,colsh);
q0=zeros(1,n);
q1=zeros(1,n);
for i=1:len
Q0(ii(i),jj(i))=f0(jj(i));
end
for i=1:len
Q1(ii(i),jj(i))=f1(jj(i));
end
while (count_iter<max_iter)
dQ=Q0-Q1;
for i=1:len
dR(ii(i),jj(i))=produ(dQ,ii,jj,i); %HORIZONTAL STEP
end
for i=1:len
R0(ii(i),jj(i))=(1/2)*(1+dR(ii(i),jj(i)));
end
for i=1:len
R1(ii(i),jj(i))=(1/2)*(1-dR(ii(i),jj(i)));
```

```
end
% VERTICAL
for i=1:len
Q0(ii(i),jj(i))=f0(jj(i))*vert_prod(R0,ii,jj,i);
end
for i=1:len
Q1(ii(i),jj(i))=f1(jj(i))*vert_prod(R1,ii,jj,i);
end
%VERTICAL_END
for i=1:len
[Q0(ii(i),jj(i)),Q1(ii(i),jj(i))]=normalize(Q0(ii(i),jj(i)),Q1(ii(i),
jj(i)));
end
% a posteriori symbol probabilities q0,q1
for i=1:n
q0(i)= f0(i)*column_product(R0,ii,jj,i);
end
for i=1:n
q1(i)= f1(i)*column_product(R1,ii,jj,i);
[q0(i),q1(i)]=normalize(q0(i),q1(i));
end
vector_est=estimate(q0,q1,n);
130
vector_est;
if bin_multi(vector_est,H')==0
break;
end;
count_iter=count_iter+1;
end
disp('iterations: ');disp(count_iter);
vector_est;
error_counter=0;
final_msg=vector_est(n-k+1:n);
for p=1:k
if final_msg(p)~=m(p)
disp('error in position ');disp(p);
error_counter=error_counter+1;
end
end
%disp('iterations: ');disp(count_iter);
sprintf('Errors: %d %d dB \n',error_counter,
SNRrange(SNRrange_index))
error_rate(1, packetnumber)=error_counter/k;
end
sprintf('For SNR= %d \n mean error rate is ',
SNRrange(SNRrange_index))
mean_error_rate(1, SNRrange_index)=double(mean(error_rate));
```

```
mean_error_rate(1, SNRrange_index);
disp('new SNR');
end
plot(SNRrange, mean_error_rate);
title('LDPC – BPSK - AWGN ');
xlabel('SNR (dB)');
ylabel('BER')
```

Αρχείο bin_multi.m

```
function res=bin_multi(A,B)
% sum=0;
[ra,ca]=size(A); %get A dimensions
[rb,cb]=size(B); %get B dimensions
res=zeros(ra,cb);
if (ca==rb) ,
for i=1:ra
for k=1:cb
for j=1:ca
res(i,k)=xor(res(i,k),A(i,j)*B(j,k));
end;
end;
end;
elseif (ca~=rb)
disp('error in matrix multiplication');
else disp('...');
end;
```

Αρχείο column_product.m

```
% a posteriori symbol probs qX(j)=fX(j) * Π RX(i,j) , i e M(j) ,
X={0,1}
function cp=column_product(R,ii,jj,n)
cp=1;
for j=1:length(ii)
if jj(j)==n %same column
cp=cp*R(ii(j),jj(j));
end;
end;
```

Αρχείο estimate.m

```
% final vector estimation, in current iteration
% if q0(j)>q1(j) -> d(j)=0, else d(j)=1
function d=estimate(q0,q1,n)
for j=1:n
if q0(j)>q1(j)
```

```
d(j)=0;
else
d(j)=1;
end;
end;
```

Αρχείο f_minus.m

```
%look-up f-
Function lup = f_minus(x)
lup=abs(log(1-exp(-x)));
%look-up f+
function lup = f_plus(x)
lup=abs(log(1+exp(-x)));
end
```

Αρχείο myXor.m

```
%modulo-2 addition
function mxr=myXor (a , b)
if ((a==0) && (b==0)), mxr=0;
elseif ((a==0) && (b==1)), mxr=1;
elseif ((a==1) && (b==0)), mxr=1;
elseif ((a==1) && (b==1)), mxr=0;
else disp('error...');
end;
```

Αρχείο normalize.m

```
% normalize Q0,Q1 so that Q0(i,j)+Q1(i,j)=1
function [a,b]=normalize(inp1,inp2)
aux=1./(eps+(inp1+inp2));
a=inp1*aux;
b=inp2*aux;
```

Αρχείο bpsk.m

```
%BPSK modulation / mapping
function tr=bpsk(c)
for i=1:length(c)
if c(i)==0
tr(i)=-1;
else
tr(i)=c(i);
end
end
```

Αρχείο produ.m

```
% product calculation, sum-product algorithm
%δR(i,j)=Π δQ(i,j') , j' is_member N(i)\j
%HORIZONTAL STEP
function product=produ(dQ,ii,jj,i)
product=1;
for j=1:length(ii)
if ii(j)==ii(i) %same row
if jj(j)~=jj(i) % \j
product=product*dQ(ii(j),jj(j));
end;
end;
end;
```

Αρχείο sum_log.m

```
% product calculation, sum-product algorithm
%δR(i,j)=Π δQ(i,j') , j' is_member N(i)\j
%HORIZONTAL STEP, logarithmic decoding (product -> sum)
function sum=sum_log(LdQ,ii,jj,i)
sum=0;
for j=1:length(ii)
if ii(j)==ii(i) %same row
if jj(j)~=jj(i) % \j
sum=sum+LdQ(ii(j),jj(j));
end;
end;
end;
```

Αρχείο vert_prod.m

```
% VERTICAL STEP
%SUM-PRODUCT ALGORITHM, MACKAY'S SIMPLIFICATION
%Q0(i,j)=f0(j)* Π R0(i',j) , i' is_member M(j)\i
%same with Q1
function vp=vert_prod(R,ii,jj,i)
vp=1;
for j=1:length(ii)
if jj(j)==jj(i) %same column
```

137

```
if ii(j)~=ii(i) % \i
vp=vp*R(ii(j),jj(j));
end;
end;
end;
```

Αρχείο vertical_sum.m

```
% VERTICAL STEP
%SUM-PRODUCT ALGORITHM, MACKAY'S SIMPLIFICATION in logarithmic form
%LQ0(i,j)=Lf0(j)+ Σ LR0(i',j) , i' is_member M(j)\i
%same with LQ1
function vs=vertical_sum_log(R,ii,jj,i)
vs=0;
for j=1:length(ii)
if jj(j)==jj(i) %same column
if ii(j)~=ii(i) % \i
vs=vs+R(ii(j),jj(j));
end;
end;
end;
```