Αριστοτελειο Πανεπιστημιο Θεσσαλονικης
Πολυτεχνικη Σχολη
Τμημα Ηλεκτρολογων Μηχανικων και Μηχανικων Η/Υ

Προγραμματισμός κώδικα επανάληψης-συσσώρευσης (repeat-accumulate) και προσομοίωσή του σε περιβάλλον AWGN

Διπλωματικη Εργασια

του

ΠΑΝΟΥΡΓΙΑ Δ. ΧΡΗΣΤΟΥ

Επιβλέπων: Χρήστος Ε. Δημάκης
Επίκουρος Καθηγητής

Θεσσαλονίκη, Μάρτιος 2018

Στην Αγγελική

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον Επίκουρο καθηγητή κ. Χρήστο Ε. Δημάκη, για την επίβλεψη αυτής της διπλωματικής εργασίας καθώς και για την ευκαιρία που μου έδωσε να την εκπονήσω. Επίσης ευχαριστώ ιδιαίτερα τον υποψήφιο διδάκτορα Κωνσταντή Αρκουδογιάννη για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε καθ΄ όλη τη διάρκεια της εκπόνησης αυτής της εργασίας. Τέλος θα ήθελα να ευχαριστήσω τον πατέρα μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφερε όλα αυτά τα χρόνια.

Περιεχόμενα

Περίληψη

Στη σύγχρονη εποχή, οι ψηφιακές τηλεπικοινωνίες αποτελούν μια τεχνολογία που αποκτά συνεχώς περισσότερες πρακτικές εφαρμογές (LTE-A, Wifi, DVB-S). Αυτό οφείλεται, σε σημαντικό βαθμό, στη χρήση τεχνικών κωδικοποίησης καναλιού. Μέσω αλγορίθμων επαναληπτικής αποκωδικοποίησης (iterative decoding algorithms), έχει καταστεί δυνατή η λειτουργία των σύγχρονων τηλεπικοινωνιακών συστημάτων κοντά στο όριο χωρητικότητας, οδηγώντας σε γρήγορη και αξιόπιστη επικοινωνία. Το αποτέλεσμα είναι η χρήση τους να γίνεται ολοένα και πιο διαδεδομένη, σε πληθώρα εφαρμογών.

Στόχος της διπλωματικής αυτής εργασίας, είναι ο προγραμματισμός των LDPC Repeat-Accumulate κωδίκων με βάση το πρότυπο DVB-S2 σε Matlab, η καταγραφή και η παρουσίαση των επιδόσεών τους, καθώς και η πρόταση για περαιτέρω διερεύνησή τους μέσω προγραμματισμού σε GPU με βάση την αρχιτεκτονική CUDA.

Abstract

In the modern age, digital communications constitute a technology with an increasing number of practical applications (LTE-A, Wifi, DVB-S). This is due, to a significant extent, to the utilization of channel coding schemes. Through iterative decoding algorithms, it has been made possible for modern digital communication systems to operate close to their capacity bound, leading to fast and reliable communication. As a result, their use is being more and more wide, in a multitude of applications.

This Master thesis aims to the programming in Matlab of the LDPC Repeat-Accumulate codes described in the DVB-S2 standard, to the measurement and presentation of their performance and to the proposition of their further investigation via their programming on a GPU, by using the CUDA architecture.

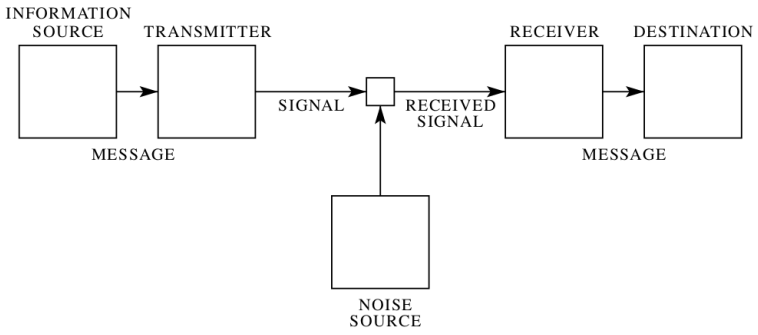
Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Ο σκοπός κάθε συστήματος επικοινωνίας είναι η μετάδοση πληροφορίας από την πηγή σε κάποιον προορισμό μέσω ενός καναλιού. Το πρωτεύον πρόβλημα αυτής της διαδικασίας, ορίστηκε από τον Claude E. Shannon στην καθοριστική εργασία του το 1948, ως η αναπαραγωγή σε ένα σημείο ενός μηνύματος που επιλέγεται σε άλλο σημείο, κατά το δυνατόν ακριβής. Ο σχεδιασμός του τηλεπικοινωνιακού συστήματος πρέπει να είναι τέτοιος, ώστε αυτό να λειτουργεί για κάθε δυνατή επιλογή μηνύματος πληροφορίας, καθώς τη στιγμή του σχεδιασμού το ακριβές αυτό μήνυμα δεν είναι γνωστό [27].

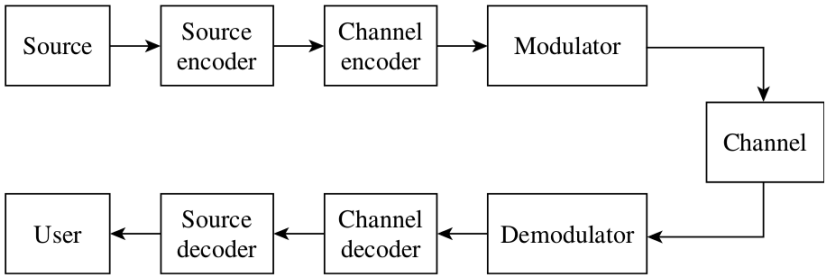


Σχήμα 1.1: Σχηματικό διάγραμμα τυπικού τηλεπικοινωνιακού συστήματος

Στο παραπάνω σχήμα (Σχήμα 1.1), φαίνεται το μπλοκ διάγραμμα ενός τυπικού τηλεπικοινωνιακού συστήματος, καθώς και τα βασικά μέρη από τα οποία αποτελείται: την πηγή πληροφορίας, τον πομπό, όπου γίνεται μετατροπή του μηνύματος πληροφορίας σε σήμα κατάλληλο για μετάδοση, το κανάλι, μέσω του οποίου γίνεται η μετάδοση, τον δέκτη, όπου γίνεται λήψη του σήματος και ανάκτηση του αρχικού μηνύματος και τέλος, τον τελικό προορισμό του μηνύματος πληροφορίας.

Τα κύρια μέρη από τα οποία αποτελείται το τηλεπικοινωνιακό σύστημα που περιγράφηκε, φαίνονται αναλυτικότερα στο Σχήμα 1.2. Το στάδιο της κωδικοποίησης πηγής

(source encoding) δεν μας απασχολεί, οπότε και θεωρείται πως από τον κωδικοποιητή πηγής εξέρχεται μια διακριτή ακολουθία από ανεξάρτητα και ομοιόμορφα κατανεμημένα (independent & identically distributed - i.i.d.) σύμβολα, τα οποία εισάγονται στον κωδικοποιητή καναλιού (channel encoder).



Σχήμα 1.2: Κύρια μέρη του τηλεπικοινωνιακού συστήματος

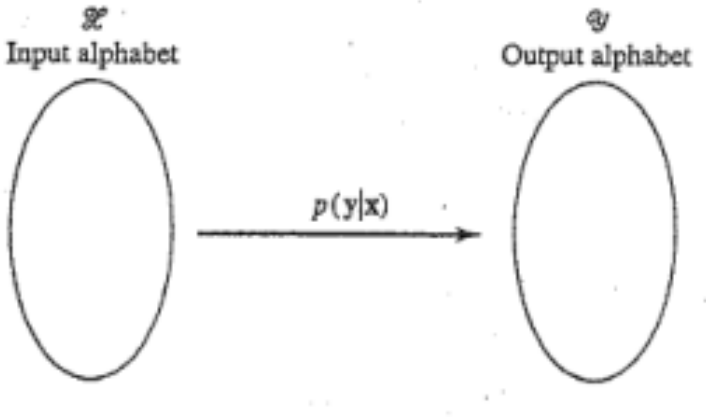
Το κανάλι επικοινωνίας

Σε αυτό το σημείο θα γίνει μια συνοπτική παρουσίαση του τηλεπικοινωνιακού καναλιού σε σχέση και με το συγκεκριμένο μοντέλο που προσομοιώθηκε.

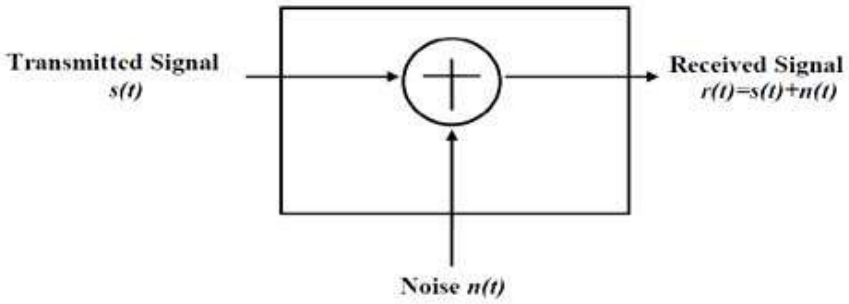
Ως κανάλι επικοινωνίας μπορεί να θεωρηθεί οποιοδήποτε μέσο μέσα στο οποίο η πληροφορία μπορεί να αποθηκευτεί ή μέσω του οποίου μπορεί να μεταδοθεί. Στα πλαίσια του τηλεπικοινωνιακού συστήματος, το τηλεπικοινωνιακό κανάλι ορίζεται ως ο χώρος που μεσολαβεί ανάμεσα στον πομπό και τον δέκτη και στον οποίο λαμβάνει χώρα η μετάδοση πληροφορίας. Σε θεωρητικό επίπεδο, το τηλεπικοινωνιακό κανάλι αποτελεί τη μαθηματική αφαίρεση των ειδών και της έντασης του θορύβου που αλλοιώνει τη μεταδιδόμενη πληροφορία.

Η είσοδος και η έξοδος των καναλιών που συναντάμε πολλές φορές στην πράξη είναι διακριτά σήματα. Σε αυτήν την περίπτωση, που οι μεταβλητές εισόδου-εξόδου είναι πεπερασμένες ή άπειρα αριθμήσιμες, το κανάλι καλείται *διακριτό* (discrete channel). Το διακριτό κανάλι, το σχηματικό διάγραμμα του οποίου φαίνεται στο Σχήμα 1.3, ορίζεται ως εξής:

Ορισμός 1.1. *Ένα διακριτό κανάλι, συμβολίζεται ως* $(\mathcal{X},\ p(y \mid x),\ \mathcal{Y})$*, και αποτελείται από δύο πεπερασμένα σύνολα (finite sets)* $\mathcal{X}$ *και* $\mathcal{Y}$*, καθώς και μια συλλογή συναρτήσεων μάζας πιθανότητας* $p(y \mid x)$*, μία για κάθε* $x \in \mathcal{X}$*, τέτοιες ώστε, για κάθε* x*,* y *να ισχύει* $p(y \mid x) \geq 0$ *και* $\sum_y p(y \mid x) = 1 \ \ \forall x$*, όταν* $\mathcal{X}$ *είναι η είσοδος και* $\mathcal{Y}$ *η έξοδος του καναλιού.*
[8], [24]



Σχήμα 1.3: Το διακριτό κανάλι



Σχήμα 1.4: Κανάλι AWGN θορύβου

Όπως φαίνεται στο Σχήμα 1.4, κατά τη διέλευση του σήματος $s(t)$ από το κανάλι, προστίθεται ο θόρυβος που αποτελεί αναπόδραστη αιτία υποβάθμισης του σήματος πληροφορίας. Ο θόρυβος εκτός από προσθετικός, μοντελοποιείται ως λευκός (ίση πυκνότητα ισχύος σε όλο το φάσμα συχνοτήτων) και Gaussian (ακολουθεί την κανονική ή γκαουσσιανή κατανομή στο πεδίο του χρόνου) θόρυβος (Additive White Gaussian Noise, AWGN).

Εν γένει, ο θόρυβος ευρείας ζώνης μπορεί να προέρχεται από πολλές φυσικές πηγές. Το Θεώρημα Κεντρικού Ορίου ορίζει πως το άθροισμα πολλών τέτοιων τυχαίων διαδικασιών, τείνει στην Γκαουσσιανή (Κανονική) κατανομή ((convergence in distribution)) [17].

Ο θόρυβος AWGN χρησιμοποιείται ευρέως ως μοντέλο καναλιού χειροτέρευσης της επικοινωνίας. Παρ΄όλο που το μοντέλο δε λαμβάνει υπ΄όψη το fading, την επιλογή στη συχνότητα, τη διασπορά ή τη μη-γραμμικότητα, εντούτοις αποτελεί απλό μαθηματικό εργαλείο για την επόπτευση του τηλεπικοινωνιακού συστήματος, πριν ληφθούν υπ΄όψη τα παραπάνω φαινόμενα. Αποτελεί ακριβές μοντέλο για τις διαστημικές επικοινωνίες,

ενώ για τις επίγειες επικοινωνίες, χρησιμοποιείται στο να δίνει το πλαίσιο προσομοίωσης του καναλιού πριν προστεθούν οι διάφορες μορφές επιπλέον υποβαθμίσεων.

1.1 Το θεώρημα Shannon

Αναφέρθηκε πως κύριος στόχος του τηλεπικοινωνιακού συστήματος, είναι η γρήγορη μετάδοση πληροφορίας μέσω του καναλιού επικοινωνίας και η αξιόπιστη ανάκτησή της στο δέκτη. Ένα σημαντικό συμπέρασμα της Θεωρίας Πληροφοριών είναι ότι, ακόμη και με την παρουσία θορύβου, είναι δυνατό να επιτευχθεί αξιόπιστη μετάδοση, αρκεί ο ρυθμός αποστολής πληροφορίας να είναι μικρότερος από ένα δεδομένο ανώφλι. Το παρακάτω θεώρημα, αποδείχθηκε από τον Shannon και αποτελεί θεμελιώδες θεώρημα της Θεωρίας Πληροφοριών:

Θεώρημα 1.1. *(Θεώρημα Κωδικοποίησης Ενθόρυβου Καναλιού) Ο ρυθμός μετάδοσης πληροφορίας μέσω ενός τηλεπικοινωνιακού συστήματος έχει μέγιστο C, που καλείται* ***χωρητικότητα καναλιού****.*

- *Όταν ισχύει:*

$$R \leq C \tag{1.1}$$

  ,όπου R ο ρυθμός αποστολής της πληροφορίας, τότε υπάρχει τεχνική κωδικοποίησης καναλιού, έτσι ώστε να είναι δυνατή η μετάδοση με αυθαίρετα μικρό σφάλμα.

- *Στην αντίθετη περίπτωση η αξιοπιστία της επικοινωνίας δεν μπορεί να ελεγχθεί.*

Αξίζει να σημειωθεί πως, ενώ το θεώρημα Shannon αποδεικνύει, για οποιοδήποτε ρυθμό μικρότερο της χωρητικότητας την ύπαρξη τεχνικών κωδικοποίησης, κατάλληλων ώστε να υπάρχει οσοδήποτε μικρό σφάλμα κατά την επικοινωνία, η απόδειξη δεν είναι κατασκευαστική (constructive), δηλαδή δεν προτείνεται κάποιος συγκεκριμένος κώδικας. Ακόμη, από το θεώρημα προκύπτει πως ο βασικός περιορισμός που θέτει ο θόρυβος σε ένα κανάλι επικοινωνίας αφορά το ρυθμό μετάδοσης δεδομένων και όχι την αξιοπιστία της επικοινωνίας [24], [27].

1.2 Τυχαίοι κώδικες - Πολυπλοκότητα

Ορισμός 1.2. *Κώδικας $C(n, M)$*

Ένας κώδικας $C(n, M)$ μήκους n και πληθυκότητας M για ένα διακριτό κανάλι που προκύπτει από τον Ορισμό 1.1, αποτελείται από τα εξής στοιχεία:

- *Ένα σύνολο δεικτών* $\{1, 2, \ldots, M\}$

- *Μια συνάρτηση κωδικοποίησης* $X^n : \{1, 2, \ldots, M\} \to \mathcal{X}^n$ *η οποία παράγει τις κωδικές λέξεις* $x^n(1), x^n(2), \ldots, x^n(M)$. *Το σύνολο των κωδικών λέξεων, καλείται κωδικό βιβλίο*

- *Μια συνάρτηση αποκωδικοποίησης* $g : \mathcal{Y}^n \to \{1, 2, \ldots, M\}$ *η οποία λειτουργεί ως ένας ντετερμινιστικός κανόνας, που αποδίδει μια πρόβλεψη σε κάθε ληφθέν διάνυσμα*

Στο εξής, όταν αναφέρεται κώδικας, θα εννοείται το κωδικό βιβλίο του, το οποίο θα συμβολίζεται ως $C(n, M)$. Θα χρειαστεί να σημειωθεί, όπως αναφέρθηκε και στην προηγούμενη παράγραφο πως, ενώ το θεώρημα Shannon αποδεικνύει την ύπαρξη κωδίκων που πλησιάζουν τη χωρητικότητα, δεν προτείνει κάποια συγκεκριμένη επιλογή βέλτιστου κώδικα.

Η ρητή κατασκευή (explicit construction) "καλών" κωδίκων αποτελεί δύσκολο έργο. Αντί αυτής, η τυπική προσέγγιση βασίζεται στην πιθανοτική μέθοδο (probabilistic method) και προβλέπει πως σε ένα σύνολο κωδίκων οι οποίοι έχουν κατασκευαστεί με τυχαίο τρόπο, η πιθανότητα να υπάρχουν καλοί κώδικες εντός του είναι θετική. Για την ακρίβεια, κάνοντας χρήση κάποιας ανισότητας συγκέντρωσης (π.χ. ανισότητα Markov), αποδεικνύεται πως η πιθανότητα να επιλεγεί ένας καλός κώδικας μέσα από αυτό το σύνολο τείνει στο 1, καθώς το μήκος του κώδικα n μεγαλώνει. Στη συνέχεια θα οριστεί ένας τυχαίος κώδικας που ικανοποιεί το Θεώρημα 1.1 και θα εξεταστεί ο τρόπος κατασκευής του και η πολυπλοκότητα την οποία εισάγει η χρήση του.

1.2.1 Τυχαίοι κώδικες

Ένα σώμα πεπερασμένου αριθμού στοιχείων (finite field) $\mathbb{F}_q$, είναι ένα σύνολο από στοιχεία των οποίων το άθροισμα και το γινόμενο, είναι επίσης στοιχεία του συνόλου. Επίσης, η πρόσθεση κι ο πολλαπλασιασμός στοιχείων του ικανοποιούν την αντιμεταθετική, προσεταιριστική, και επιμεριστική ιδιότητα. Το πλήθος των στοιχείων ενός σώματος συμβολίζεται ως $|\mathbb{F}_q|$. Ευρέως χρησιμοποιούμενο είναι το *δυαδικό* σώμα $\mathbb{F}_2$ το οποίο αποτελείται από τα στοιχεία {0,1}, επομένως ισχύει $|\mathbb{F}_2| = 2$. Το σώμα αυτό υιοθετείται και σε όλη την έκταση της παρούσας εργασίας.

Χρησιμοποιώντας το συμβολισμό του Ορισμού 1.2, ο *ρυθμός κώδικα* $C(n, M)$ ορίζεται από την παρακάτω εξίσωση:

$$R = \frac{\log_2 M}{n} \quad (bits/transmission) \tag{1.2}$$

[8]. Ο ρυθμός κώδικα αποτελεί το ρυθμό αποστολής της πληροφορίας ανν τα μηνύματα είναι ισοπίθανα.

Η κατασκευή τυχαίων κωδίκων επί του σώματος $\mathbb{F}_2$, ακολουθεί την εξής διαδικασία:

Ορισμός 1.3. *(Τυχαίο σύνολο Shannon - Shannon Random Ensemble)*

Το σύνολο όλων των κωδίκων $C(n, M)$ μήκους n και πληθυκότητας M αποτελείται από $|\mathbb{F}|_2^{nM}$ δυνατούς κώδικες, καθώς υπάρχουν nM βαθμοί ελευθερίας στην επιλογή του κωδικού βιβλίου. Στα στοιχεία του συνόλου προσδίδεται η ομοιόμορφη κατανομή πιθανότητας. [25],[27]

1.2.2 Πολυπλοκότητα

Οι (τυχαίοι) κώδικες που προκύπτουν από τον Ορισμό 1.3 ικανοποιούν και το θεώρημα Shannon για μεγάλα κωδικά μήκη n, είναι δηλαδή ικανοί να λειτουργήσουν κοντά στη χωρητικότητα, με αυθαίρετα μικρό σφάλμα. Παρά το ότι η τυχαία κωδικοποίηση αποτελεί την προφανή λύση στο πρόβλημα κωδικοποίησης, υστερεί σε δυνατότητα πρακτικής υλοποίησης, καθώς δεν λαμβάνει υπ΄όψη την πολυπλοκότητα αποθήκευσης του κώδικα (πολυπλοκότητα περιγραφής) και την πολυπλοκότητα κωδικοποίησης και αποκωδικοποίησης.

Στην περίπτωση ενός κώδικα που προκύπτει από τον Ορισμό 1.3, είναι φανερό πως η πολυπλοκότητα αποθήκευσης αυξάνει εκθετικά συναρτήσει του μήκους του κώδικα, καθώς απαιτούνται $nM = n2^{Rn}$ bits για την αποθήκευση του κωδικού βιβλίου $C(n, M)$ στον κωδικοποιητή. Επιπλέον, δεδομένου ενός μηνύματος, ο κωδικοποιητής θα πρέπει να διατρέξει το αποθηκευμένο κωδικό βιβλίο για να αποφασίσει τη μεταδιδόμενη κωδικολέξη. Ακόμη ο αποκωδικοποιητής θα πρέπει να έχει αποθηκευμένη κάθε δυνατή κωδική λέξη και (όπως ήδη αναφέρθηκε) να εκτελέσει αναζήτηση για τον έλεγχο και την απόφαση του μηνύματος πληροφορίας που στάλθηκε. Παρατηρείται συνεπώς, πως με την αύξηση του μήκους n η πολυπλοκότητα καθίσταται απαγορευτική.

Συνοψίζοντας, διαπιστώνεται πως το πρόβλημα έγκειται στη δυνατότητα προσέγγισης της χωρητικότητας με *πρακτικό* τρόπο.

1.2.3 Πρακτικοί κωδικοποιητές/αποκωδικοποιητές

Τη δυνατότητα αυτή, δίνουν κώδικες που διακρίνονται από αλγεβρικές δομικές ιδιότητες και ορίζονται από συγκεκριμένους κανόνες για την κωδικοποίηση και την αποκωδικοποίηση.

Θα αποδειχθεί στη συνέχεια της εργασίας, πως ακολουθώντας τη συγκεκριμένη διαδικασία, η πολυπλοκότητα κωδικοποίησης και αποκωδικοποίησης απλοποιείται σημαντικά, και καταλήγει να είναι τετραγωνικής ($\mathcal{O}(n^2)$) ή και γραμμικής ($\mathcal{O}(n)$) τάξης. Οι αλγεβρικές δομικές ιδιότητες προσδίδονται στους κώδικες ακριβώς για το σκοπό αυτό. Θα οριστούν επίσης έννοιες όπως οι αραιοί πίνακες, η επαναληπτική αποκωδικοποίηση και τα γραφήματα Tanner ως εργαλεία για την περιγραφή τέτοιων κωδίκων.

1.3 Δομή εργασίας

Η εργασία ακολουθεί την εξής δομή: στο Κεφάλαιο 2 θα αναλυθούν οι αλγεβρικές δομικές ιδιότητες των κωδίκων καναλιού, ώστε να καταστεί πρακτική η κωδικοποίηση και αποκωδικοποίησή τους και θα οριστούν οι γραμμικοί μετασχηματισμοί για την εκτέλεσή τους, οι πίνακες **G** και **H**. Στο Κεφάλαιο 3 θα οριστούν και θα περιγραφούν οι κώδικες επανάληψης - συσσώρευσης (repeat-accumulate) ως υποομάδα τόσο της κωδικοποίησης turbo όσο και των LDPC κωδίκων και θα εξεταστεί ο μηχανισμός και η πολυπλοκότητα κωδικοποίησης και αποκωδικοποίησής τους. Στο Κεφάλαιο 4 θα περιγραφεί αναλυτικά η προσομοίωση που έγινε και θα παρουσιαστούν τα αποτελέσματά της. Τέλος στο Κεφάλαιο 5 θα παρουσιαστούν τα συμπεράσματα που προέκυψαν από την προσομοίωση, καθώς και οι δυνατότητες περαιτέρω μελέτης και πρόσθετες επεκτάσεις, στηριζόμενες στη γλωσσα C++ και τον προγραμματισμό σε GPU μέσω της αρχιτεκτονικής CUDA.

Κεφάλαιο 2

Γραμμικοί Μπλοκ Κώδικες

Στο κεφάλαιο αυτό, παρουσιάζονται οι βασικές αλγεβρικές δομικές ιδιότητες που προσδίδονται στους κώδικες καναλιού, οι οποίες διακρίνουν και τους κώδικες πάνω στους οποίους στηρίζεται η παρούσα εργασία και που θα παρουσιαστούν στο επόμενο κεφάλαιο. Όπως αναφέρθηκε στο εισαγωγικό κεφάλαιο, το σώμα που υιοθετείται είναι το $\mathbb{F}_2$. Συνεπώς, θεωρείται πως η έξοδος της πηγής πληροφορίας είναι μια διακριτή ακολουθία i.i.d. δυαδικών ψηφίων, η οποία καλείται *ακολουθία πληροφορίας*. Η μετάδοση λαμβάνει χώρα μέσω διακριτού καναλιού, όπως αυτό ορίστηκε στον Ορισμό 1.1. Υπενθυμίζεται ακόμη πως η έκφραση «κώδικας $C(n, M)$», αναφέρεται στο κωδικό βιβλίο του κώδικα.

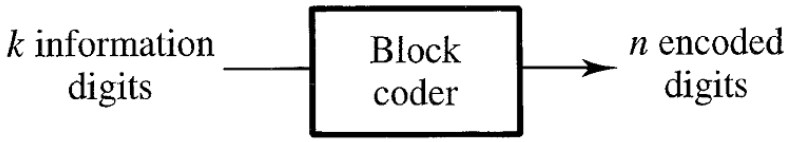
2.1 Μπλοκ κώδικες

Ξεκινώντας τη μελέτη των δομών που αναφέρθηκαν παραπάνω, ορίζεται η κλάση των κωδίκων καναλιού, οι κώδικες *μπλοκ* (block codes).

Η αρχή λειτουργίας της μπλοκ κωδικοποίησης, συνίσταται στην κατάτμηση της ροής πληροφορίας σε σταθερά μπλοκ $\mathbf{u} = (u_0, u_1, ..., u_{k-1})$ μήκους k και στην απεικόνισή της πριν την είσοδο του καναλιού, μέσω του κωδικοποιητή καναλιού, στην κωδική λέξη $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$ μήκους n, με $n > k$. Η απεικόνιση αυτή είναι ανεξάρτητη από τα προηγούμενα μπλοκ, δηλαδή δεν υπάρχει μνήμη από ένα μήνυμα, σε ένα άλλο επόμενο [24].

2.1.1 Ρυθμός δυαδικού κώδικα

Στο προηγούμενο κεφάλαιο ορίστηκε ο ρυθμός του κώδικα ως $R = \frac{\log_2 M}{n}$. Επίσης ισχύει πως η πληθυκότητα (cardinality) του μπλοκ κώδικα C στο $\mathbb{F}_2$, το σύνολο δηλαδή των δυνατών κωδικών λέξεων του κωδικού βιβλίου, είναι $M = 2^k$. Εφαρμόζοντας τις



Σχήμα 2.1: Σχηματικό διάγραμμα ενός μπλοκ κώδικα

παραπάνω σχέσεις, προκύπτει ο λόγος

$$R = \frac{\log_2 2^k}{n} = \frac{k}{n} \tag{2.1}$$

ο οποίος καλείται *ρυθμός* του μπλοκ κώδικα (*code rate*) και αντιπροσωπεύει τον μέσο όρο πληροφορίας που αποστέλλεται με κάθε κωδικό bit, ανν τα bits πληροφορίας είναι ανεξάρτητα και ισοπίθανα (i.u.d.). Προφανώς ισχύει $0 < R < 1$.

Όπως αναφέρθηκε, ο κώδικας αντιστοιχίζει κάθε μήνυμα πληροφορίας $\mathbf{u}$, στην κωδικολέξη $\mathbf{v}$, μήκους n, με $n > k$. Τα $n - k$ επιπλέον bits που προστίθενται στο μήνυμα από τον κωδικοποιητή καναλιού καλούνται *πλεονασματικά* (redundant) bits.

Τα πλεονασματικά bits δεν περιέχουν επιπλέον πληροφορία και ο σκοπός τους είναι να δώσουν δυνατότητες *ανίχνευσης* και *διόρθωσης* σφαλμάτων, που προκύπτουν κατά τη μετάδοση στο κανάλι, λόγω θορύβου ή/και παρεμβολών. Το πώς διαμορφώνονται αυτά τα πλεονασματικά bits, ώστε ο μπλοκ κώδικας να έχει ικανοποιητικές δυνατότητες ανίχνευσης ή/και διόρθωσης, αποτελεί μείζων ζήτημα της σχεδίασης μηχανισμών κωδικοποίησης [26].

2.2 Γραμμικοί μπλοκ κώδικες

Επιπρόσθετα από την δόμηση σε κωδικά blocks, είναι αναγκαίο να προσδώσουμε δομικές ιδιότητες στον κώδικα $C(n, M)$, ώστε οι παραπάνω διαδικασίες να καταστούν πρακτικά υλοποιήσιμες. Μια τέτοια δομική ιδιότητα είναι η *γραμμικότητα*.

Ορισμός 2.4. *Ένας δυαδικός κώδικας μήκους n καλείται (n, k) γραμμικός μπλοκ κώδικας, ανν οι 2^k κωδικές του λέξεις σχηματίζουν έναν k-διάστατο υποχώρο του διανυσματικού χώρου V όλων των δυνατών n-άδων στο $\mathbb{F}_2$ [26].*

Ένας ισοδύναμος ορισμός είναι ο εξής:

Ορισμός 2.5. *Ένας κώδικας μπλοκ είναι γραμμικός αν κάθε γραμμικός συνδυασμός δύο κωδικών του λέξεων, είναι επίσης κωδική του λέξη. Στο $\mathbb{F}_2$, αν $\mathbf{c_i}$ και $\mathbf{c_j}$ οι δύο*

κωδικές λέξεις, τότε πρέπει $\mathbf{c_i} \oplus \mathbf{c_j}$ *να είναι επίσης κωδική λέξη του* C*, όπου με* $\oplus$ *συμβολίζεται η συνιστώσα-προς-συνιστώσα modulo-2 πρόσθεση [24].*

Η ακολουθία $\mathbf{0}$ αποτελεί κωδική λέξη κάθε γραμμικού block κώδικα, αφού μπορεί να προκύψει ως συνιστώσα-προς-συνιστώσα modulo-2 πρόσθεση μιας κωδικής λέξης με τον εαυτό της. Ακόμη φαίνεται πως η γραμμικότητα ενός κώδικα εξαρτάται αποκλειστικά από τις κωδικές του λέξεις και όχι από τον τρόπο με τον οποίο η ακολουθία πληροφορίας απεικονίζεται σε αυτές.

2.3 Αναπαράσταση με πίνακα

2.3.1 Γεννήτορας πίνακας, G

Σύμφωνα με τον ορισμό του γραμμικού μπλοκ κώδικα $C(n,k)$, αποδεικνύεται πως υπάρχουν k γραμμικά ανεξάρτητες κωδικές λέξεις, $\mathbf{g_0}, \mathbf{g_1}, ..., \mathbf{g_{k-1}}$, έτσι ώστε κάθε κωδική λέξη $\mathbf{v}$ να προκύπτει ως γραμμικός συνδυασμός τους. Οι k γραμμικά ανεξάρτητες αυτές κωδικές λέξεις λέγεται ότι σχηματίζουν μια *βάση* του C.

Με βάση αυτό, η κωδική λέξη προκύπτει ως εξής: έστω $\mathbf{u} = (u_0, u_1, ..., u_{k-1})$ το μήνυμα στην είσοδο του κωδικοποιητή. Η κωδικολέξη $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$ δίνεται από τον γραμμικό συνδυασμό των $\mathbf{g_0}, \mathbf{g_1}, ..., \mathbf{g_{k-1}}$ με συντελεστές τα k bits πληροφορίας, από την παρακάτω εξίσωση:

$$\mathbf{v} = u_0\mathbf{g_0} + u_1\mathbf{g_1} + ... + u_{k-1}\mathbf{g_{k-1}} \tag{2.2}$$

Οι συνιστώσες των k γραμμικά ανεξάρτητων κωδικολέξεων μπορούν να αναπαρασταθούν ως γραμμές ενός $k \times n$ πίνακα στο $\mathbb{F}_2$ ως εξής:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix} \tag{2.3}$$

Σε αυτή την περίπτωση, η κωδικολέξη $\mathbf{v}$ που αντιστοιχεί στο μήνυμα $\mathbf{u}$, δίνεται από τον πολλαπλασιασμό πινάκων:

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{G} \tag{2.4}$$

Ο πίνακας $\mathbf{G}$ καλείται *πίνακας γεννήτορας* (generator matrix) του (n,k) γραμμικού μπλοκ κώδικα C. Γενικά, ένας γραμμικός μπλοκ κώδικας δεν έχει μοναδική βάση.

Συνεπώς κάθε επιλογή βάσης του C δίνει διαφορετικό γεννήτορα πίνακα, οπότε προκύπτει πως ο πίνακας $\mathbf{G}$ δεν είναι μοναδικός για δεδομένο κώδικα. Ο βαθμός (rank) του πίνακα $\mathbf{G}$ είναι προφανώς ίσος με τη διάσταση του κώδικα C.

Παρατηρείται συνεπώς, πως μειώνεται σημαντικά η πολυπλοκότητα περιγραφής του κώδικα, αφού πλέον αρκεί ο κωδικοποιητής να αποθηκεύσει τις k γραμμές του πίνακα $\mathbf{G}$. Επιπρόσθετα, η εκτέλεση του πολλαπλασιασμού πινάκων που περιγράφει η 2.4 απαιτεί περίπου $2kn$ πράξεις, δηλαδή είναι τετραγωνικής πολυπλοκότητας ως προς το n - $\mathcal{O}(n^2)$.

2.3.2 Πίνακας ελέγχου ισοτιμίας, H

Ορισμός 2.6. *Ο δυικός κώδικας (dual code) του C, C_d δίνεται από τις παρακάτω n-άδες:*

$$C_d = \{\mathbf{w} \in V : \langle \mathbf{w}, \mathbf{v} \rangle = 0 \quad \forall\, \mathbf{v} \in C\}$$

, όπου με $\langle \cdot, \cdot \rangle$ συμβολίζεται το διανυσματικό εσωτερικό γινόμενο [26].

Από τον Ορισμό 2.6 προκύπτουν τα παρακάτω:

- Αν η βάση του C αποτελείται από k γραμμικά ανεξάρτητες κωδικές λέξεις, η βάση του C_d αποτελείται από $(n-k)$ γραμμικά ανεξάρτητες n-άδες.
- Έστω $\mathbf{h_0}, \mathbf{h_1}, ..., \mathbf{h_{n-k-1}}$ οι $(n-k)$ αυτές, γραμμικά ανεξάρτητες n-άδες. Συνεπάγεται πως κάθε n-άδα στον C_d προκύπτει ως γραμμικός συνδυασμός τους
- Ομοίως με τον πίνακα $\mathbf{G}$, οι παραπάνω γραμμικά ανεξάρτητες n-άδες μπορούν να αναπαρασταθούν ως γραμμές ενός πίνακα στο $\mathbb{F}_2$, που σχηματίζεται ως εξής:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h_0} \\ \mathbf{h_1} \\ \vdots \\ \mathbf{h_{n-k-1}} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,n-1} \end{bmatrix} \tag{2.5}$$

- Ο πίνακας $\mathbf{H}$ είναι γεννήτορας πίνακας του C_d, ακριβώς όπως ο $\mathbf{G}$ είναι του C. Λόγω αυτού και από την ιδιότητα της ορθογωνιότητας, προκύπτει η πολύ χρήσιμη εξίσωση $\mathbf{G} \cdot \mathbf{H^T} = \mathbf{0}$, όπου $\mathbf{0}$ μηδενικός $k \times (n-k)$ πίνακας.

Ακόμη, προκύπτει πως ο κώδικας C ορίζεται με μοναδικό τρόπο από τον πίνακα $\mathbf{H}$, επειδή κάθε δυαδική n-άδα $\mathbf{v}$ είναι κωδικολέξη του C ανν ισχύει $\mathbf{v} \cdot \mathbf{H^T} = \mathbf{0}$, δηλαδή

$$C = \{\mathbf{v} \in V : \mathbf{v} \cdot \mathbf{H^T} = \mathbf{0}\} \tag{2.6}$$

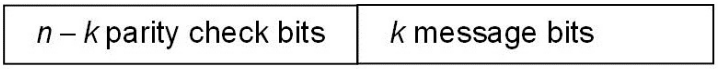
Ο πίνακας **H** καλείται πίνακας *ελέγχου ισοτιμίας* (parity check matrix) του κώδικα C. Από την παραπάνω σχέση ορισμού ενός γραμμικού κώδικα (εξίσωση 2.6), προέρχεται και το όνομα του **H**, αφού, στην πλευρά του αποκωδικοποιητή, ο **H** χρησιμοποιείται για την επαλήθευση του παραπάνω συστήματος ομογενών εξισώσεων, όταν το διάνυσμα **v**, αντικατασταθεί με το ληφθέν διάνυσμα.

Παρατηρείται επομένως, πως ο γραμμικός μπλοκ κώδικας C μπορεί να οριστεί πλήρως από δύο πίνακες, τον γεννήτορα ή τον πίνακα ελέγχου ισοτιμίας. Στη γενική περίπτωση, η κωδικοποίηση του C βασίζεται στον γεννήτορα πίνακα και ακολουθεί την εξίσωση 2.4, ενώ η αποκωδικοποίηση βασίζεται στον πίνακα ελέγχου ισοτιμίας.

Ο πίνακας ελέγχου ισοτιμίας μπορεί να περιγράψει πλήρως έναν κώδικα, και ορισμένες κατηγορίες γραμμικών μπλοκ κωδίκων κατασκευάζονται με βάση τον πίνακα **H**. Ο πίνακας **H** είναι *πλήρους βαθμού* (full rank), ανν ο βαθμός του είναι ίσος με το πλήθος των σειρών. Σε πολλές περιπτώσεις, ο πίνακας **H** δεν δίνεται σε full rank μορφή, δηλαδή το πλήθος των σειρών του είναι μεγαλύτερο από το rank του, $n-k$, ή ισοδύναμα, ορισμένες από τις σειρές του είναι γραμμικός συνδυασμός των $n-k$ γραμμικά ανεξάρτητων σειρών. Οι σειρές αυτές ονομάζονται *πλεονάζουσες* (redundant) σειρές.

Οι κώδικες Ελέγχου Ισοτιμίας Χαμηλής Πυκνότητας (Low Density Parity Check - LDPC), που θα εξεταστούν στη συνέχεια της εργασίας, διακρίνονται από πίνακες **H** που δεν είναι full rank [8], [26].

2.4 Κώδικες σε συστηματική μορφή



Σχήμα 2.2: Κωδική λέξη σε συστηματική μορφή

Μία ακόμη επιθυμητή ιδιότητα, είναι οι κωδικές λέξεις ενός γραμμικού μπλοκ κώδικα $C(n,k)$ να έχουν τη μορφή που φαίνεται στο Σχήμα 2.2, η οποία καλείται *συστηματική μορφή* (systematic form). Στη συστηματική μορφή, η κωδική λέξη χωρίζεται στο τμήμα μηνύματος (data part) και στο πλεονάζον τμήμα ελέγχου (parity part). Το τμήμα μηνύματος αποτελείται από τα k αναλλοίωτα ψηφία πληροφορίας και το πλεονάζον τμήμα ελέγχου, από τα $n-k$ bits ελέγχου ισοτιμίας. Ολόκληρος ο γραμμικός block κώδικας $C(n,k)$ βρίσκεται σε *συστηματική* μορφή, όταν όλα τα στοιχεία του κωδικού βιβλίου βρίσκονται σε συστηματική μορφή, ή ισοδύναμα, όταν μπορεί να οριστεί πλήρως από έναν $k \times n$ γεννήτορα πίνακα με την παρακάτω μορφή:

$$\mathbf{G} = [\mathbf{P} \;\; \mathbf{I}_k] = \left[\begin{array}{cccc|cccc} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} & 1 & 0 & \cdots & 0 \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & \cdots & 1 \end{array}\right] \tag{2.7}$$

$\mathbf{P}$ πίνακας; $k \times k$ $\mathbf{I}_k$ μοναδιαίος πίνακας

Όπως διαπιστώνεται και από την εξίσωση 2.7, ο γεννήτορας πίνακας $\mathbf{G}$, που για συντομία σημειώνεται ως $\mathbf{G} = [\mathbf{P} \;\; \mathbf{I}_k]$, αποτελείται από δύο υποπίνακες, έναν $k \times (n-k)$ υποπίνακα $\mathbf{P}$ στα αριστερά, ο οποίος καλείται υποπίνακας ελέγχου ισοτιμίας του $\mathbf{G}$ και έναν $k \times k$ μοναδιαίο πίνακα $\mathbf{I}_k$ στα δεξιά [26].

Η συστηματική μορφή του πίνακα $\mathbf{G}$ έχει σημαντικά πλεονεκτήματα καθώς σε κάθε κωδική λέξη, τα bits πληροφορίας παραμένουν αναλλοίωτα στις τελευταίες k θέσεις της. Μειώνεται συνεπώς η πολυπλοκότητα κωδικοποίησης, αφού χρειάζεται να υπολογιστούν μόνο τα σύμβολα στις θέσεις του πλεονάζοντος τμήματος ελέγχου.

Αντίστοιχα, η συστηματική μορφή του πίνακα ελέγχου ισοτιμίας διαμορφώνεται ως εξής:

$$\mathbf{H} = \left[\mathbf{I}_{n-k} \;\; \mathbf{P^T}\right] = \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{0,0} & p_{1,0} & \cdots & p_{k-1,0} \\ 0 & 1 & \cdots & 0 & p_{0,1} & p_{1,1} & \cdots & p_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \tag{2.8}$$

Προφανώς ισχύει και πάλι ότι $\mathbf{G} \cdot \mathbf{H^T} = \mathbf{0}$.

Θεωρείται και πάλι το μήνυμα πληροφορίας $\mathbf{u} = (u_0, u_1, ..., u_{k-1})$. Αν εφαρμοστεί η εξίσωση 2.4, όταν ο πίνακας $\mathbf{G}$ είναι σε συστηματική μορφή, η κωδική λέξη προκύπτει από τον παρακάτω πολλαπλασιασμό:

$$\begin{aligned} \mathbf{v} &= (u_0, u_1, \cdots, u_{k-1}) \cdot \mathbf{G} \\ &= (v_0, v_1, \cdots, v_{n-k-1}, v_{n-k}, \cdots, v_{n-1}) \end{aligned} \tag{2.9}$$

τα n στοιχεία της οποίας προκύπτουν ως εξής:

$$\begin{aligned} v_{n-k+i} &= u_i \quad \forall \;\; 0 \le i \le k-1 \\ v_j &= u_0 p_{0,j} + u_1 p_{1,j} + \cdots + u_{k-1} p_{k-1,j} \quad \forall \;\; 0 \le j \le n-k-1 \end{aligned} \tag{2.10}$$

Οι εξισώσεις 2.10 δείχνουν πως τα δεξιότερα k bits της κωδικής λέξης παραμένουν αναλλοίωτα τα k bits του μηνύματος πληροφορίας και τα υπόλοιπα $n-k$ αποτελούν γραμμικό συνδυασμό των bits πληροφορίας και ορίζονται πλήρως από τις $n-k$ στήλες του υποπίνακα $\mathbf{P}$, όπως αυτός ορίζεται στην εξίσωση 2.7. Η δεύτερη έκφραση, αποτελεί ένα σύστημα $n-k$ εξισώσεων οι οποίες ονομάζονται εξισώσεις ελέγχου ισοτιμίας.

Η συστηματική μορφή προσφέρει το πλεονέκτημα λιγότερων υπολογισμών κατά την κωδικοποίηση, καθώς το μήνυμα πληροφορίας παραμένει ατόφιο και απαιτείται υπολογισμός μόνο των bits του τμήματος ελέγχου, κάτι που οδηγεί σε μειωμένη πολυπλοκότητα περιγραφής. Ακόμη, προσφέρει ένα επιπλέον πλεονέκτημα κατά την αποκωδικοποίηση, μιας και αφού αποφασιστεί ποια είναι η αποσταλείσα κωδική λέξη, εξάγεται άμεσα το μήνυμα.

2.5 Κατανομή Βαρών - Ελάχιστη Απόσταση Hamming

Παρακάτω θα δοθούν οι ορισμοί μερικών βασικών παραμέτρων ενός κώδικα. Οι έννοιες που θα οριστούν, βοηθούν στην κατανόηση της δυνατότητας εντοπισμού και διόρθωσης σφαλμάτων των γραμμικών μπλοκ κωδίκων που θα συζητηθεί στη συνέχεια.

Ορισμός 2.7. *Απόσταση Hamming*
Έστω δύο n-άδες, $\mathbf{v}$ *και* $\mathbf{w}$*. Η απόσταση Hamming ορίζεται ως ο αριθμός των συνιστωσών στις οποίες διαφέρουν μεταξύ τους και συμβολίζεται με* $d(\mathbf{v}, \mathbf{w})$*.*

Ορισμός 2.8. *Βάρος Hamming*
Το βάρος Hamming μιας n-άδας, $\mathbf{v}$ *ορίζεται ως το πλήθος των μη μηδενικών στοιχείων της και συμβολίζεται με* $w(\mathbf{v})$

Από τους ορισμούς 2.7, 2.8 προκύπτει πως η απόσταση Hamming μεταξύ δύο n-άδων ισούται με το βάρος του αθροίσματός τους modulo-2, δηλ $d(\mathbf{v}, \mathbf{w}) = w(\mathbf{v} + \mathbf{w})$.

Ορισμός 2.9. *Η ελάχιστη απόσταση ενός κώδικα ορίζεται ως η ελάχιστη απόσταση Hamming μεταξύ δύο διαφορετικών κωδικών λέξεων, δηλαδή*

$$d_{min} = \min_{\mathbf{v},\mathbf{w}} d(\mathbf{v}, \mathbf{w}) \tag{2.11}$$

Ορισμός 2.10. *Το ελάχιστο βάρος ενός κώδικα, ορίζεται από την παρακάτω σχέση*

$$w_{min} = \min_{\mathbf{v}\neq\mathbf{0}} w(\mathbf{v}) \tag{2.12}$$

είναι δηλαδή, το ελάχιστο των βαρών των κωδικών λέξεων του κώδικα, αν εξαιρεθεί η μηδενική κωδική λέξη.

Από τα παραπάνω, καθώς και από την ιδιότητα της γραμμικότητας, προκύπτει το παρακάτω θεώρημα:

Θεώρημα 2.2. *Η ελάχιστη απόσταση ενός γραμμικού κώδικα είναι ίση με το ελάχιστο βάρος του. Η απόδειξη του θεωρήματος είναι η ακόλουθη:*

$$\begin{aligned} d_{min}(C) &= \min\{d(\mathbf{v},\mathbf{w}) : \mathbf{v},\mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\} \\ &= \min\{w(\mathbf{v}+\mathbf{w}) : \mathbf{v},\mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\} \\ &= \min\{w(\mathbf{x}) : \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\} \\ &= w_{min}(C). \end{aligned} \tag{2.13}$$

Μπορεί πλέον να οριστεί και η *κατανομή βάρους* του κώδικα C. Έστω A_i ο αριθμός των κωδικών λέξεων του $C(n,k)$ με βάρος Hamming i. Η κατανομή βάρους του C αποτελείται από τους αριθμούς $A_0, A_1, \cdots, A_n$ για $0 \leq i \leq n$. Ισχύουν οι παρακάτω σχέσεις:

$$\begin{aligned} A_0 + A_1 + \cdots + A_n &= 2^k \\ A_0 &= 1 \end{aligned} \tag{2.14}$$

Η κατανομή βάρους ενός κώδικα αποτελεί σημαντικό εργαλείο στον καθορισμό της πιθανότητας να συμβεί ένα μη-ανιχνεύσιμο σφάλμα κατά την αποκωδικοποίηση καθώς αποτελεί παράγοντα του άνω ορίου της πιθανότητας αυτής. Δίνει επίσης την εικόνα της κατανομής αποστάσεων μεταξύ των κωδικών λέξεων ενός κώδικα, σε σχέση με την κωδική λέξη $\mathbf{0}$, καθώς το A_i αποτελεί και τον αριθμό των κωδικών λέξεων που απέχουν από τη μηδενική κωδικολέξη, απόσταση i [26], [20], [22].

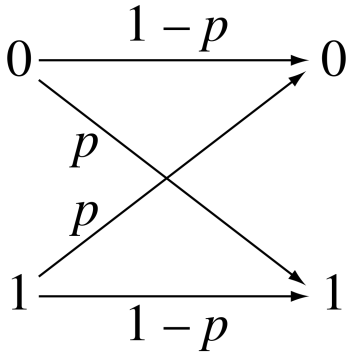
2.6 Ανίχνευση σφαλμάτων - Αποκωδικοποίηση - Πολυπλοκότητα

Έστω ένας γραμμικός μπλοκ κώδικας $C(n,k)$, που ορίζεται από τον πίνακα ελέγχου ισοτιμίας $\mathbf{H}$ και $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$, $\mathbf{r} = (r_0, r_1, \cdots, r_{n-1})$ η κωδική λέξη στην είσοδο του καναλιού και το λαμβανόμενο σήμα στο δέκτη αντίστοιχα.

Το διάνυσμα

$$\mathbf{e} = \mathbf{r} + \mathbf{v}$$

, όπου $e_i = r_i + v_i,\ 0 \leq j \leq n-1$ και η πρόσθεση είναι modulo-2, περιέχει 1 στις θέσεις στις οποίες η κωδικολέξη και το λαμβανόμενο διάνυσμα διαφέρουν. Συνεπώς, δίνει μια εικόνα των σφαλμάτων που υφίσταται η κωδική λέξη κατά τη διέλευσή της από το διακριτό κανάλι και καλείται πρότυπο σφάλματος (error pattern).



Σχήμα 2.3: Το δυαδικό συμμετρικό κανάλι

Αν υποθέσουμε πως το κανάλι είναι *δυαδικό συμμετρικό* (Binary Symmetric Channel - BSC) κανάλι όπως στο Σχήμα 2.3, η πιθανότητα να συμβεί σφάλμα σε οποιαδήποτε από τις n θέσεις είναι η ίδια - p. Υπάρχουν $2^n - 1$ διαφορετικά μη-μηδενικά πρότυπα σφάλματος.

Για την ανίχνευση σφαλμάτων, ο αποκωδικοποιητής υπολογίζει τις παρακάτω $(n - k)$-άδες στο $\mathbb{F}_2$:

$$\mathbf{s} = (s_0, s_1, \cdots, s_{n-k-1}) = \mathbf{r} \cdot \mathbf{H^T} \tag{2.15}$$

Το διάνυσμα $\mathbf{s}$ καλείται *σύνδρομο* του $\mathbf{r}$ [26]. Σύμφωνα με την εξίσωση 2.15 και όσα έχουν ήδη αναφερθεί, το ληφθέν διάνυσμα θα είναι κωδικολέξη του C ανν $\mathbf{s} = 0$. Σε αντίθετη περίπτωση το ληφθέν διάνυσμα δεν είναι κωδική λέξη και σίγουρα περιέχει *σφάλματα μετάδοσης*.

Στην περίπτωση που ισχύει $\mathbf{s} = 0$, ο αποκωδικοποιητής θεωρεί το ληφθέν διάνυσμα κωδικολέξη του C χωρίς σφάλματα. Ωστόσο, στην περίπτωση που το $\mathbf{r}$ ανήκει στο κωδικό βιβλίο του C αλλά διαφέρει από την κωδική λέξη που στάλθηκε $\mathbf{v}$, ο αποκωδικοποιητής διαπράττει *σφάλμα αποκωδικοποίησης*. Αυτό συμβαίνει στην περίπτωση που το πρότυπο σφάλματος $\mathbf{e}$ ταυτίζεται με μία κωδική λέξη και καλείται *μη-ανιχνεύσιμο* πρότυπο σφάλματος. Υπάρχουν $2^k - 1$ μη-ανιχνεύσιμα πρότυπα σφάλματος [20].

2.6.1 Αποκωδικοποίηση

Ένας από τους σκοπούς της χρήσης κωδικοποίησης είναι η αύξηση της Ευκλείδειας απόστασης μεταξύ των μεταδιδόμενων σημάτων και κατά συνέπεια, η μείωση της πιθανότητας σφάλματος για δεδομένη ισχύ εκπομπής. Για σηματοδοσία BPSK ή QPSK με απεικόνιση Gray, αυτό ισοδυναμεί με την απαίτηση η απόσταση Hamming μεταξύ των κωδικών λέξεων να είναι η μεγαλύτερη δυνατή. Καθώς ο υπολογισμός της απόστασης Hamming μεταξύ κάθε κωδικολέξης είναι πρακτικά αδύνατη διαδικασία, η σύγκριση της επίδοσης κωδίκων γίνεται με βάση την ελάχιστη απόσταση d_{min} (ή με βάση το

ελάχιστο βάρος w_{min}). Συνεπάγεται πως κώδικες με μεγαλύτερη d_{min} έχουν συνήθως καλύτερες επιδόσεις [24].

Θεωρείται ο $C(n,k)$ γραμμικός block κώδικας με πίνακα ελέγχου ισοτιμίας $\mathbf{H}$, ελάχιστη απόσταση $d_{min}(C)$ και το ληφθέν διάνυσμα στο δέκτη, $\mathbf{r}$. Επιγραμματικά αναφέρεται πως, για αποκωδικοποίηση *μέγιστης πιθανοφάνειας* (maximum-likelihood decoding - MLD), η απόφαση για τη μεταδιδόμενη κωδικολέξη λαμβάνεται με βάση τη μεγιστοποίηση της υπο-συνθήκη πιθανότητας $P(\mathbf{r} \mid \mathbf{v})$. Για το κανάλι BSC του Σχήματος 2.3, αυτό ισοδυναμεί με τον υπολογισμό της απόστασης μεταξύ του $\mathbf{r}$ και κάθε κωδικής λέξης $\mathbf{v}$ και την επιλογή της κωδικής λέξης που ελαχιστοποιεί την απόσταση αυτή. Αυτός ο τρόπος αποκωδικοποίησης ονομάζεται *κοντινότερου γείτονα* (nearest-neighbor) και αποτελεί αποκωδικοποίηση πλήρους *διόρθωσης σφαλμάτων* (complete error-correction).

Η συγκεκριμένη μέθοδος αποκωδικοποίησης απαιτεί 2^k υπολογισμούς της απόστασης Hamming. Συνεπώς είναι αδύνατον να υλοποιηθεί πρακτικά για μεγάλα μήκη κώδικα. Στη συνέχεια θα δειχθούν μέθοδοι, για να πετύχουν εφάμιλλες επιδόσεις και ραγδαία μείωση της πολυπλοκότητας [26].

2.6.2 Πολυπλοκότητα

Τέλος, αναφορικά με την πολυπλοκότητα, αναφέρεται πως ο ορισμός της έννοιας της πολυπλοκότητας προσεγγίζεται δύσκολα. Ήδη από τη αναπαράσταση ενός γραμμικού μπλοκ κώδικα, φαίνεται πως η πολυπλοκότητα περιγραφής (το ποσό μνήμης που χρειάζεται για να αποθηκευτεί ενάς κώδικας), είναι το πολύ $\min\langle Rn^2, (1-R)\, n^2\rangle$ bits, όπου R ο ρυθμός του κώδικα. Επίσης όπως αναφέρθηκε ήδη, η κωδικοποίηση όπως προκύπτει από την εξίσωση 2.4, γίνεται σε τετραγωνικό χρόνο $\mathcal{O}(n^2)$ [25].

Η πολυπλοκότητα αποκωδικοποίησης MLD σε ένα κανάλι BSC έχει αναλυθεί από τον Berlekamp κ.α. [3]. Αποδεικνύεται πως η MLD αποκωδικοποίηση είναι NP-complete, συνεπώς θεωρείται απίθανο να βρεθεί γενικός αλγόριθμος για τέτοιου τύπου αποκωδικοποίηση με πολυωνυμική πολυπλοκότητα [5].

Κεφάλαιο 3

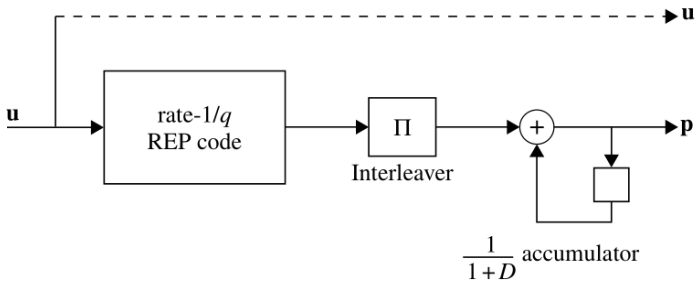
Κώδικες Επανάληψης-Συσσώρευσης (repeat - accumulate, RA)

Στο κεφάλαιο αυτό, παρουσιάζονται οι κώδικες Επανάληψης-Συσσώρευσης (repeat - accumulate, RA), η μελέτη της επίδοσης των οποίων αποτελεί και το αντικείμενο της παρούσας εργασίας.

Οι RA αποτελούν τους πρώτους κώδικες που βασίζονται σε συσσωρευτές και εφευρέθηκαν από τους D. Divsalar κ .ά. [9]. Ενώ έχουν απλή δομή, διακρίνονται από καλές επιδόσεις και πρακτική κωδικοποίηση, με μοναδικό μειονέκτημα ότι υστερούν από άποψη επιδόσεων στους χαμηλούς ρυθμούς (1/2 ή χαμηλότερου). Πλέον, χρησιμοποιούνται ήδη σε αρκετά πρότυπα τηλεπικοινωνιών (DVB-S2, IEEE802.16). Διακρίνονται ως μια ειδική κλάση των *σειριακά αλυσιδωτών* κωδίκων (serially concatenated codes - SC), στους οποίους ο εξωτερικός κώδικας είναι ένας επαναληπτικός κώδικας ρυθμού $1/q$ και ο εσωτερικός είναι ένας συνελικτικός κώδικας γεννήτορα $\frac{1}{1+D}$, ο οποίος δίνει το $mod-2$ άθροισμα του bit εισόδου με το προηγούμενο bit εξόδου. Παράγει δηλαδή το άθροισμα όλων των παρελθουσών εισόδων και για το λόγο αυτό καλείται και *συσσωρρευτής* (accumulator), στοιχείο που δίνει στους RA το όνομά τους.

Οι RA που μεταδίδουν τα bits πληροφορίας αυτούσια λέγονται συστηματικοί. Ένα σχηματικό διάγραμα ενός συστηματικού RA κώδικα, φαίνεται στο Σχήμα 3.1. Η αποκωδικοποίησή τους μπορεί να αντιμετωπισθεί είτε ως σειριακή turbo, είτε ως LDPC, τακτική που είναι και η πιο διαδεδομένη [26].

Στη συνέχεια του κεφαλαίου παρουσιάζονται συνοπτικά οι κώδικες που μπορούν να πλησιάσουν το θεωρητικό όριο χωρητικότητας Shannon (Θεώρημα 1.1), διατηρώντας παράλληλα τη δυνατότητα πρακτικής κωδικοποίησης και αποκωδικοποίησης. Αφού γίνει μια σύντομη αναφορά στη σχέση χωρητικότητας και σηματοθορυβικής σχέσης (SNR), παρουσιάζονται οι τρόποι θεώρησης των RA, είτε ως turbo, είτε ως LDPC. Κατόπιν



Σχήμα 3.1: Μπλοκ διάγραμμα ενός RA κώδικα

αφού αναλυθεί ο τρόπος κωδικοποίησης των LDPC (ως ο πιο συχνός τρόπος θεώρησης των RA), δίνεται πλήρως το coding scheme των RA, που δίνει και τη βάση πάνω στην οποία στηρίζεται η προσομοίωση που έγινε και θα αναλυθεί στο Κεφάλαιο 4.

3.1 Κώδικες που πλησιάζουν τη χωρητικότητα

Μέχρι και πολύ πρόσφατα, οι κώδικες που μπορούσαν να λειτουργήσουν κοντά στο θεωρητικό όριο χωρητικότητας που προέβλεπε το θεώρημα Shannon, ήταν κυρίως μη πρακτικοί κώδικες. Με την ανακάλυψη των turbo κωδίκων και την επανανακάλυψη των LDPC τη δεκαετία του ’90, έγινε δυνατό να αποδειχθεί η ικανότητα λειτουργίας τους κοντά στη χωρητικότητα, με πρακτικά υλοποιήσιμους κωδικοποιητές-αποκωδικοποιητές σε σχετικά χαμηλά bit error rates (BERs), όσον αφορά το κανάλι AWGN.

Παρ’όλο που η θεωρητική επίτευξη του ορίου χωρητικότητας απαιτεί κώδικες με άπειρο μήκος, στην πράξη αρκεί το μήκος του κώδικα να είναι αρκετά μεγάλο (π.χ. της τάξης των μερικών δεκάδων χιλιάδων). Η σχεδίαση τέτοιων κωδίκων χαρακτηρίζεται από 2 βασικά στοιχεία:

- Χρήση κωδίκων αποτελούμενων από απλά μέρη, συνενωμένα με τρόπο που να παράγεται μια ψευδο-τυχαία κατανομή βαρών και,
- Χρήση υποβέλτιστων επαναληπτικών αλγορίθμων αποκωδικοποίησης με ανταλλαγή soft πληροφορίας, των οποίων η πολυπλοκότητα αυξάνει γραμμικά με την αύξηση του μήκους κώδικα.

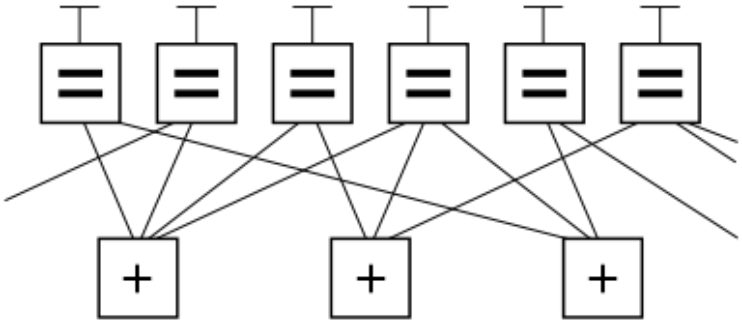
Μπορεί επίσης να δοθεί και ο παρακάτω ορισμός των κωδίκων που πλησιάζουν τη χωρητικότητα:

Ορισμός 3.11. *Capacity-approaching κώδικες*

Έστω μια ακολουθία από δυαδικούς γραμμικούς κώδικες (C_m)*, ρυθμού* R_m *και για κάθε κώδικα, οι κωδικές λέξεις μεταδίδονται ισοπίθανα μέσω ενός καναλιού με*

χωρητικότητα C. Η ακολουθία επιτυγχάνει κλάσμα $1-\epsilon$ της χωρητικότητας του καναλιού αν $\lim_{m\to\infty} R_m \geq (1-\epsilon)\cdot C$ και υπάρχει αλγόριθμος αποκωδικοποίησης για τον οποίο η πιθανότητα εσφαλμένου bit του κώδικα C_m τείνει στο μηδέν, όταν $m \to \infty$ [23].

Δύο πολύ γνωστές κλάσεις capacity-approaching κωδίκων είναι οι εξής: οι turbo ή turbo-like κώδικες, οι οποίοι συνίστανται από συνελικτικούς συστατικούς κώδικες (component codes) και οι κώδικες Ελέγχου Ισοτιμίας Χαμηλής Πυκνότητας (Low Density Parity Check - LDPC).



Σχήμα 3.2: Μέρος του γραφήματος παραγόντων ενός LDPC κώδικα

Και οι δύο κατηγορίες που αναφέρθηκαν χαρακτηρίζονται ως *Κώδικες σε Γραφήματα* (Codes on Graphs), διότι αναπαρίστανται από *γραφήματα παραγόντων* (factor graphs) πάνω στα οποία εκτελείται ο αλγόριθμος για την επαναληπτική αποκωδικοποίησή τους. Στο Σχήμα 3.2 παραθέτουμε μέρος του γραφήματος ενός LDPC κώδικα [26], [18], [7].

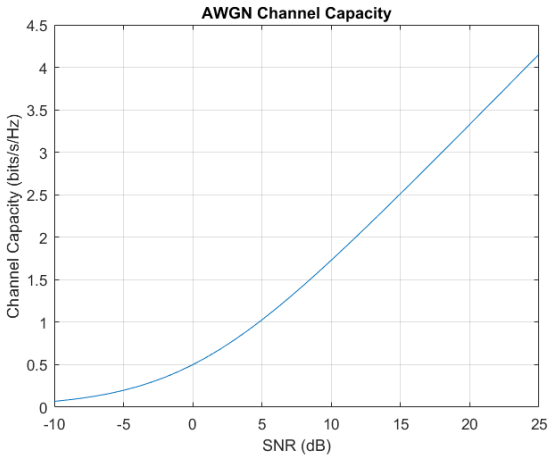
3.1.1 Η χωρητικότητα ως SNR

Σε αυτό το σημείο θα γίνει μια σύντομη αναφορά στη σχέση που συνδέει τη χωρητικότητα ενός καναλιού με τη σηματοθορυβική σχέση SNR που το χαρακτηρίζει.

Έστω τηλεπικοινωνιακό σύστημα που εισάγει AWGN θόρυβο. Η χωρητικότητα καναλιού αποδεικνύεται ότι δίνεται από την παρακάτω εξίσωση:

$$C = \frac{1}{2}\log_2(1+SNR) \quad \text{bit/μετάδοση} \tag{3.1}$$

Από την παραπάνω σχέση φαίνεται ότι η χωρητικότητα εξαρτάται μόνο από το SNR. Από εδώ και στο εξής, ως χωρητικότητα θα εννοούμε το ελάχιστο SNR (συνήθως ως E_b/N_0), που απαιτείται για την επίτευξη οσοδήποτε αξιόπιστης επικοινωνίας με δεδομένο ρυθμό C, όπως φαίνεται και από το Σχήμα 3.3.



Σχήμα 3.3: Χωρητικότητα AWGN καναλιού

3.2 RA ως turbo

Σε αυτή την παράγραφο, θα γίνει μια σύντομη αναφορά στους κώδικες turbo, την αποκωδικοποίησή τους, καθώς και την δυνατότητα των RA να χαρακτηριστούν ως τέτοιοι.

3.2.1 Κώδικες turbo

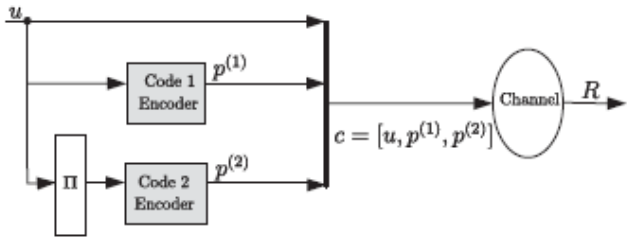
Οι κώδικες turbo ανακαλύφθηκαν από τους Berrou, Glavieux και Thitimajshima [4] το 1993 και αποτέλεσαν ριζοσπαστική προσέγγιση της κωδικοποίησης για διόρθωση σφαλμάτων. Αποτελούνται από τον παράλληλο συνδυασμό δύο συνελικτικών κωδίκων, οι οποίοι κατά την αποκωδικοποίηση διαμοιράζουν πληροφορία μεταξύ των αντίστοιχων αποκωδικοποιητών. Αναφέρεται πως ο κωδικοποιητής χρησιμοποιεί συνελικτικούς κωδικοποιητές [16], ενώ ο αποκωδικοποιητής χρησιμοποιεί BCJR αποκωδικοποιητές [2].

Ο κωδικοποιητής turbo περιέχει έναν *αναδιατάκτη* (interleaver), ο ρόλος του οποίου είναι να μεταθέτει κωδικές λέξεις μικρού βάρους από τον ένα κωδικοποιητή σε κωδικές λέξεις μεγάλου βάρους στον άλλο. Ο αναδιατάκτης αυτός διαθέτει ψευδοτυχαία χαρακτηριστικά.

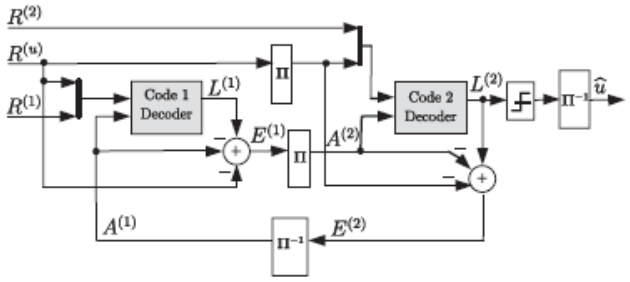
Η αποκωδικοποίηση στηρίζεται στην ανταλλαγή soft πληροφορίας μεταξύ δύο συστατικών αποκωδικοποιητών. Η πληροφορία αυτή έχει τη μορφή λογαριθμικού λόγου πιθανοτήτων (log-likelihood ratio, LLR) για καθένα από τα bits πληροφορίας. [4].

3.2.2 Κωδικοποίηση RA

Το συνολικό coding scheme των RA που προσομοιώθηκε θα αναλυθεί στη συνέχεια του κεφαλαίου. Ωστόσο, η αποκωδικοποίηση τους μπορεί να βασιστεί στην



Σχήμα 3.4: Κωδικοποιητής turbo



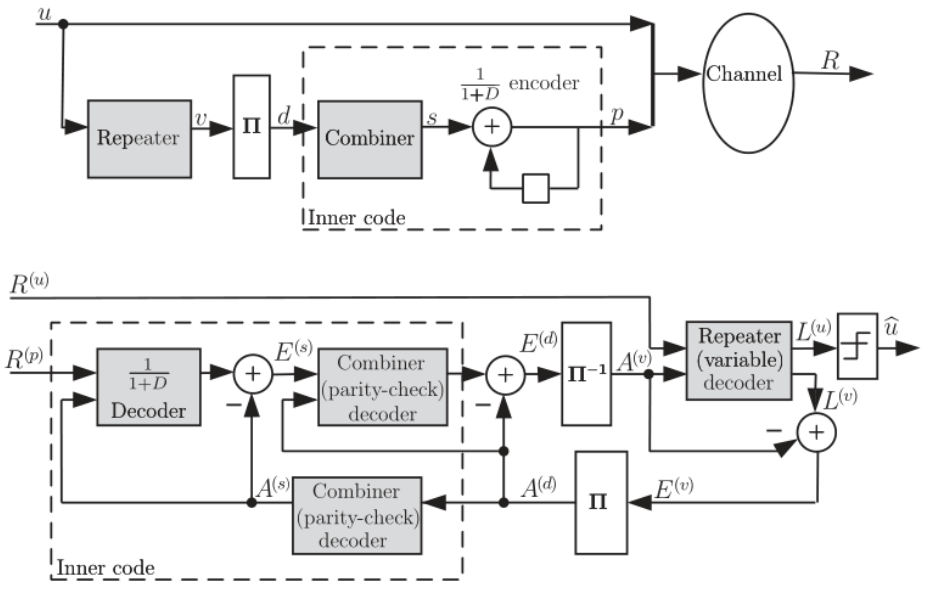
Σχήμα 3.5: Αποκωδικοποιητής turbo

turbo αποκωδικοποίηση των SC, όπου, όπως αναφέρθηκε ήδη, ο εξωτερικός κώδικας είναι repetition και ο εσωτερικός είναι ο accumulator. Για τους συστηματικούς RA κώδικες που περιέχουν συνδυαστή, ο εσωτερικός κώδικας είναι συνδυαστικός αποκωδικοποιητής συνδυαστή-συσσωρευτή (accumulator-combiner - AC) και ο REP αποκωδικοποιητής έχει ως είσοδο τα LLRs από το κανάλι [18].

Στο Σχήμα 3.6, φαίνεται η κωδικοποίηση και αποκωδικοποίηση turbo ενός RA κώδικα, βάσει της *αρχής turbo* (turbo principle) [15]. Σε κάθε επανάληψη της αποκωδικοποίησης, ο REP αποκωδικοποιητής λαμβάνει LLRs από το κανάλι αναφορικά με τα κωδικά bits και στις επόμενες επαναλήψεις, ως επιπλέον, a priori πληροφορία τα LLRs εξωγενούς πληροφορίας από τον AC αποκωδικοποιητή.

Στο επόμενο βήμα ο REP αποκωδικοποιητής υπολογίζει τα LLRs εξόδου. Η εξωγενής πληροφορία από τον REP αποκωδικοποιητή, παρεμβάλλεται από τον interleaver για την παραγωγή a-priori LLRs στον AC αποκωδικοποιητή. Στην τελική επανάληψη, λαμβάνεται απόφαση για τα bits πληροφορίας από τα τελικά LLRs που εξάγει ο REP αποκωδικοποιητής. Ο αλγόριθμος τελειώνει μετά από ένα προκαθορισμένο αριθμό I_{max} επαναλήψεων. Το block διάγραμμα διαμόρφωσης της κωδικής λέξης ενός RA κώδικα, δόθηκε ήδη στο Σχήμα 3.1. Τα bits πληροφορίας αντιγράφονται q φορές, περνούν από τον αναδιατάκτη, ομαδοποιούνται μέσω $mod-2$ πρόσθεσης σε block μήκους a και τέλος περνούν από ένα συνελικτικό κωδικοποιητή μνήμης -1.

Πιο συγκεκριμένα, το μήνυμα πληροφορίας μήκους k, $\mathbf{u} = (u_0, u_1, \cdots, u_{k-1})$, μετά την έξοδο από τον επαναληπτικό κωδικοποιητή, έχει τη μορφή:



Σχήμα 3.6: Κωδικοποιητής / turbo αποκωδικοποιητής συστηματικού RA

$$\begin{aligned}\mathbf{c} &= \left[c_1, c_2, \cdots, c_{qk}\right] \\ &= (\underbrace{u_0, u_0, \cdots, u_0}_{q}, \underbrace{u_1, u_1, \cdots, u_1}_{q}, \cdots, \underbrace{u_{k-1}, u_{k-1}, \cdots, u_{k-1}}_{q})\end{aligned} \tag{3.2}$$

, όπου $c_i = u_{f(i)}$ με $f(i) = \lceil i/q \rceil$ και $\lceil x \rceil$ τον αμέσως μεγαλύτερο ακέραιο από το x.

Κατόπιν, κατά την έξοδο από τον αναδιατάκτη $\Pi = [\pi_1, \pi_2, \cdots, \pi_{qk}]$, διαμορφώνεται μια μετάθεση της ακολουθίας $\mathbf{c}$ ως εξής:

$$\mathbf{d} = \left[d_1, d_2, \cdots, d_{qk}\right] = [c_{\pi_1}, c_{\pi_2}, \cdots, c_{\pi_{qk}}]$$

Ο συνδυαστής λαμβάνει την έξοδο του αναδιατάκτη, προσθέτει $(mod - 2)$ και ομαδοποιεί τα bits σε block μήκους a. Τα bits της ακολουθίας εξόδου του συνδυαστή $\mathbf{s}$, δίνονται από την εξίσωση:

$$s_i = d_{a(i-1)+1} \oplus d_{a(i-1)+2} \oplus \cdots \oplus d_{ai}, \quad i = 1, 2, \cdots, m \;\; m = kq/a \tag{3.3}$$

Στην έξοδο του συσσωρευτή, τα parity bits θα δίνονται από την εξίσωση:

$$\begin{aligned} p_0 &= 0 \\ p_i &= p_{i-1} \oplus s_i \quad i = 1, 2, \cdots, m \end{aligned} \tag{3.4}$$

Από την εξίσωση 3.4, προκύπτει πως η κωδική λέξη του συστηματικού RA κώδικα, έχει τη μορφή $\mathbf{v} = [u_0, u_1, \cdots, u_{k-1}, p_1, p_2, \cdots, p_m]$, οπότε το μήκος του κώδικα προκύπτει $n = k(1 + q/a)$ και ο ρυθμός $R = a/(a + q)$ [18].

3.3 RA ως LDPC

Οι κώδικες LDPC παρουσιάστηκαν για πρώτη φορά το 1962 από τον R. G. Gallager στη διδακτορική του διατριβή [14] και αποτελούν κώδικες διόρθωσης σφαλμάτων που ορίζονται από αραιό πίνακα ελέγχου ισοτιμίας. Παρά το ότι αποτελούν capacity-approaching κώδικες εγκαταλείφθηκαν για περίπου 30 χρόνια, με εξαίρεση τη δουλειά του Tanner [28] που εισήγαγε και τη γραφική αναπαράστασή τους, μέσω του γραφήματος Tanner, λόγω περιορισμών στην τεχνική τους υλοποίηση και επανανακαλύφθηκαν το 1996 από τους Mackay κ.α. [19]. Το κύριο χαρακτηριστικό τους, η χαμηλή πυκνότητα του πίνακα $\mathbf{H}$, είναι και αυτό που κάνει τους LDPC να επιδέχονται διάφορους αλγόριθμους επαναληπτικής αποκωδικοποίησης (iterative decoding). Παρ'ότι οι αλγόριθμοι επαναληπτικής αποκωδικοποίησης έχουν μη-βέλτιστη απόδοση, μπορούν να έχουν απόδοση που χαρακτηρίζεται "σχεδόν" βέλτιστη (near-optimal), σε εφαρμογές/error-rates ενδιαφέροντος. Πρόσφατα αποδείχτηκε από τους Chung, Richardson κ.α. η δυνατότητα των LDPC προσέγγισης του θεωρητικού ορίου Shannon κατά $0.0045\ dB$ [6].

Κάθε γραμμικός κώδικας έχει αναπαράσταση μέσω πίνακα ελέγχου-ισοτιμίας, ωστόσο για να είναι "αραιή" η αναπαράσταση πρέπει να πληρούνται ορισμένα κριτήρια. Ένας $m \times n$ πίνακας καλείται αραιός αν το ποσό από άσσους (1) στις γραμμές και στις στήλες του, το βάρος γραμμών (w_r) και στηλών (w_c), είναι πολύ μικρότερο από τις διαστάσεις του ($w_r \ll n$, $w_c \ll m$). Πρακτικά, θεωρείται πως αν λιγότερο από 1% των στοιχείων του πίνακα είναι άσσοι, τότε είναι αραιός. Αν το βάρος γραμμών και στηλών είναι σταθερό ο κώδικας LDPC λέγεται ομαλός, ενώ σε αντίθετη περίπτωση λέγεται ανώμαλος [29].

Γενικά, εκτός από την απαίτηση να είναι ο πίνακας $\mathbf{H}$ αραιός, οι LDPC δε διαφέρουν από τους μπλοκ κώδικες. Ωστόσο, η εύρεση ενός αραιού πίνακα ελέγχου ισοτιμίας για ένα δεδομένο μπλοκ κώδικα αποτελεί δύσκολη διαδικασία και της κατασκευής των LDPC προηγείται η κατασκευή του πίνακα $\mathbf{H}$ και ακολουθεί η σχεδίαση του κωδικοποιητή. Οι LDPC αποκωδικοποιούνται επαναληπτικά, κάνοντας χρήση γραφικής αναπαράστασης του πίνακα ελέγχου ισοτιμίας.

Έχει διαπιστωθεί πως και οι RA και οι LDPC ενέχουν πλεονέκτημα σε σχέση με τους turbo κώδικες, καθώς προσφέρουν μια πιο ευέλικτη δομή, κάτι που με τη σειρά του προσφέρει περισσότερους βαθμούς ελευθερίας στην επιλογή των παραμέτρων για δεδομένο κριτήριο σχεδίασης [18].

3.3.1 Κωδικοποίηση LDPC

Όπως αναφέρθηκε στο Κεφάλαιο 2, οι γραμμικοί μπλοκ κώδικες αποτυπώνουν το μήνυμα πληροφορίας $\mathbf{u}$ στην κωδική λέξη $\mathbf{v}$ κάνοντας χρήση της εξίσωσης 2.4, για

δοσμένο γεννήτορα πίνακα **G**. Γενικά, όπως έχει επίσης αναφερθεί και στο Κεφάλαιο 2, ο πίνακας **G** ενός κώδικα δίνεται από το null-space του -αραιού για τους LDPC- πίνακα **H**. Συνεπώς είναι απίθανο να είναι και ο ίδιος αραιός, κάτι που θα οδηγούσε σε κέρδος ως προς το χρόνο κωδικοποίησης (ως προς το μήκος του κώδικα). Προκύπτει επομένως πως, η κωδικοποίηση ενός LDPC με βάση την εξίσωση 2.4 γίνεται σε τετραγωνικό χρόνο ως προς το μήκος κώδικα. Σημειώνεται πως έχουν γίνει διάφορες προσπάθειες προς την κατεύθυνση της μείωσης του χρόνου αυτού. Η πιο εξέχουσα από αυτές είναι η κωδικοποίηση RA [29].

3.3.2 Κωδικοποίηση RA

Ο πίνακας **H** των RA κωδίκων έχει την ακόλουθη ειδική μορφή:

$$\mathbf{H} = [H_1 \; H_2] \tag{3.5}$$

Έχοντας τον υποπίνακα H_2 στη μορφή 3.6 παρατηρούμε ότι το πρώτο parity-check bit υπολογίζεται από μια υποομάδα των bits πληροφορίας που ορίζει η πρώτη γραμμή του υποπίνακα H_1. Κατόπιν, το δεύτερο parity-check bit υπολογίζεται από το άρτι υπολογισθέν bit ελέγχου ισοτιμίας συν μια υποομάδα bits ληροφορίας που ορίζει η δεύτερη γραμμή του υποπίνακα H_1. Έτσι, η διαδικασία της κωδικοποίησης καθίσταται γραμμικού χρόνου. Αντιλαμβανόμαστε ότι ο υποπίνακας H_2 αντιστοιχεί στη λειτουργία του συσσωρευτή, ενώ ο H_1 στις λειτουργίες του επαναληπτικού κώδικα, του αναδιατάκτη και του συνδυαστή.

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 0 & 0 & & 0 & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & & 0 & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 & 0 \\ 0 & 0 & 0 & & 0 & 1 & 1 \end{bmatrix} \tag{3.6}$$

3.3.3 Αποκωδικοποίηση RA

Αναφέρθηκε ήδη η δυνατότητα των RA κωδίκων για turbo αποκωδικοποίηση. Ωστόσο, στο πλαίσιο της συγκεκριμένης εργασίας, μελετάται η δυνατότητα για αποκωδικοποίηση ως LDPC, χρησιμοποιώντας τον αλγόριθμο *Sum-Product* (Sum-Product algorithm, SPA).

Ο αλγόριθμος παρουσιάστηκε από τον Gallager μαζί με τους κώδικες LDPC. Χαρακτηρίζεται από near-optimal επίδοση αποκωδικοποίησης καθώς και από καθολικότητα

για τα διάφορα κανάλια χωρίς μνήμη (BEC, BSC, BI-AWGN, κ.λ.π.), συνεπώς η ανάπτυξή του είναι γενική. Το κριτήριο βέλτιστης ανάπτυξης του αλγορίθμου, είναι ο υπολογισμός της μέγιστης εκ των υστέρων (maximum a posteriori - MAP) πιθανότητας. Υπολογίζεται δηλαδή η (εκ των υστέρων) πιθανότητα για την τιμή κάθε συγκεκριμένου bit της κωδικής λέξης $\mathbf{v}$, με δεδομένο το ληφθέν διάνυσμα $\mathbf{r}$.

Το γράφημα Tanner

Το γράφημα Tanner ενός LDPC κώδικα [28], είναι ένα *διμερές* γράφημα (bipartite graph) το οποίο δίνει μια πλήρη αναπαράσταση του κώδικα και αποτυπώνει τη λειτουργία των αλγορίθμων αποκωδικοποίησής του. Αποτελείται από δύο διαφορετικού τύπου κόμβους, με τις ακμές του να συνδέουν αποκλειστικά κόμβους του αντίθετου τύπου.

Οι δύο τύποι κόμβων στο γράφημα Tanner είναι οι *κόμβοι μεταβλητών* (variable nodes, VNs) και οι *κόμβοι ελέγχου* (check nodes, CNs). Ο σχηματισμός του γραφήματος Tanner γίνεται συνδέοντας τον CN i με τον VN j, αν το στοιχείο h_{ij} του πίνακα $\mathbf{H}$ είναι 1. Παρατηρείται επομένως πως υπάρχουν ακριβώς m CNs, ένας για κάθε μια εξίσωση ελέγχου και n VNs, ένας για κάθε κωδικό bit. Αντίστοιχα οι m γραμμές του $\mathbf{H}$ αντιπροσωπεύουν τις συνδέσεις των CN και οι n στήλες τις συνδέσεις των VN. Τέλος, οι n-bit κωδικές λέξεις που αντιπροσωπεύονται από τους VNs, είναι ακριβώς όλες οι κωδικές λέξεις του κώδικα.

Για τους LDPC κώδικες, το γράφημα Tanner αποτελεί σκιαγράφημα της επαναληπτικής αποκωδικοποίησης, καθώς κάθε κόμβος δρα σαν τοπικός επεξεργαστής και κάθε ακμή σα δίαυλος μεταφοράς πληροφορίας, από ένα κόμβο στους γειτονικούς. Η ανταλλασσόμενη πληροφορία είναι πιθανοτική, π.χ. LLRs [26].

Ο Sum-Product αλγόριθμος για LDPC

Η αποκωδικοποίηση του κωδικού bit v_j απαιτεί τον υπολογισμό της APP πιθανότητας (για τιμή bit 1):

$$\Pr(v_j = 1|\mathbf{r})$$

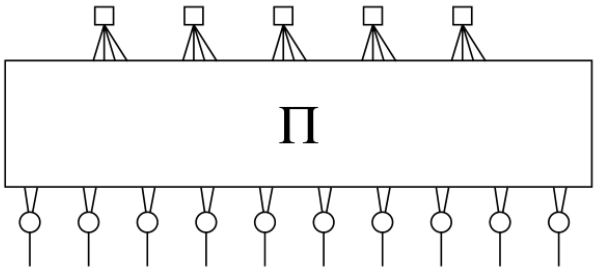
το λόγο APP

$$l(v_j|\mathbf{r}) \triangleq \frac{\Pr(v_j = 0|\mathbf{r})}{\Pr(v_j = 1|\mathbf{r})}$$

ή το (σταθερότερο αριθμητικά) λόγο log-APP, που καλείται επίσης log-likelihood ration (LLR):

$$L(v_j|\mathbf{r}) \triangleq \log\left(\frac{\Pr(v_j = 0|\mathbf{r})}{\Pr(v_j = 1|\mathbf{r})}\right) \tag{3.7}$$

Ο υπολογισμός της εξίσωσης 3.7 γίνεται εφαρμόζοντας την αρχή turbo στο γράφημα Tanner. Με αναφορά το Σχήμα 3.7, ο LDPC μπορεί να θεωρηθεί ως ένα σύνολο από



Σχήμα 3.7: Αναπαράσταση LDPC ως αλληλουχία από SPC (επάνω) και REP (κάτω) κώδικες. Με Π συμβολίζεται η αναδιάταξη

SPC κώδικες συνδεόμενους μέσω του αναδιατάκτη σε ένα σύνολο REP κωδίκων. Οι SPC κώδικες (CNs στο γράφημα Tanner) θεωρούνται εξωτερικοί κώδικες, συνεπώς δε συνδέονται στο κανάλι.

Τα Σχήματα 3.8, 3.9 αναπαριστούν στιγμιότυπα των REP και SPC αποκωδικοποιητών (κόμβοι VN και CN αντίστοιχα).

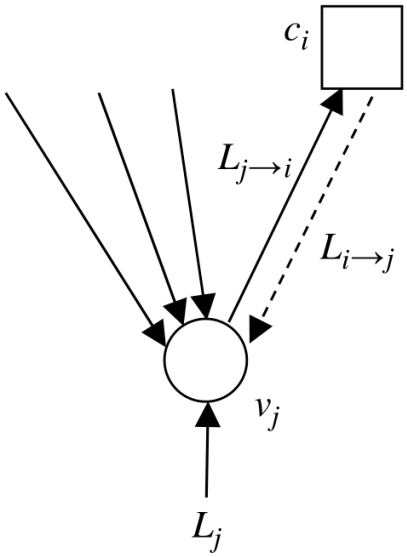
Οι αποκωδικοποιητές (κόμβοι) VN και CN λειτουργούν συνεργατικά για τον υπολογισμό της εκτίμησης $L(v_j|\mathbf{r}),\ \ j=0,1,\cdots,n-1$, επαναληπτικά. Κατά τη λειτουργία τους, υιοθετείται το *flooding schedule*, σύμφωνα με το οποίο ξεκινώντας από τους VN, οι VN (CN) επεξεργάζονται την είσοδο και υπολογίζουν *εξωγενή πληροφορία* την οποία στέλνουν στους γειτονικούς CN (VN). Η διαδικασία ολοκληρώνεται μετά από ένα προκαθορισμένο αριθμό επαναλήψεων ή αφού εκπληρωθεί ένα κριτήριο τερματισμού και υπολογίζεται η πιθανότητα $L(v_j|\mathbf{r})$, μέσω της οποίας λαμβάνεται απόφαση για την τιμή του bit v_j. Η υλοποίηση του SPA βασίζεται επίσης στη *υπόθεση ανεξαρτησίας*: οι ληφθείσες ποσότητες LLR σε κάθε κόμβο από τους γειτονικούς, είναι μεταξύ τους ανεξάρτητες. Η υπόθεση αυτή, όμως, παύει να ισχύει από έναν αριθμό επαναλήψεων και μετά, γι'αυτό και ο SPA δεν εξάγει πάντα την ακριβή τιμή του $L(v_j|\mathbf{r})$ αλλά μια καλή προσέγγισή της.

Θα αναφερθούν επίσης οι εσωτερικές διαδικασίες και υπολογισμοί, που λαμβάνουν χώρα στους κόμβους VN και CN, για τον υπολογισμό της μετρικής LLR που ανταλλάσσεται μεταξύ γειτονικών κόμβων. Οι VN και CN λειτουργούν σαν APP επεξεργαστές για REP και SPC κώδικες αντίστοιχα, σε κάθε επανάληψη του αλγορίθμου.

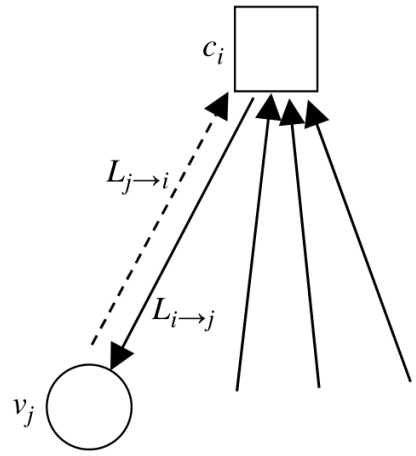
Η εξωγενής πληροφορία που στέλνεται από τον VN j στον CN i δίνεται από την εξίσωση:

$$L_{j\to i} = L_j + \sum_{i'\in N(j)-\{i\}} L_{i'\to j} \tag{3.8}$$

, όπου η ποσότητα L_j δίνεται από την εξίσωση 3.7, αφού η ακολουθία $\mathbf{r}$ αντικατασταθεί



Σχήμα 3.8: Ο κόμβος VN j (αποκωδικοποιητής REP) λαμβάνει πληροφορία από τους γειτονικούς CN (εκτός από τον i ($L_{i\to j}$)) και στέλνει στον CN i την ποσότητα $L_{j\to i}$



Σχήμα 3.9: Ο κόμβος CN i (αποκωδικοποιητής SPC) λαμβάνει πληροφορία από τους γειτονικούς VN (εκτός από τον j ($L_{j\to i}$)) και στέλνει στον VN j την ποσότητα $L_{i\to j}$

από το κωδικό bit r_j. Η εξαίρεση του i-οστού CN κόμβου από το άθροισμα της Σχέσης 3.8 οφείλεται στην αρχή της εξωγενούς πληροφορίας: ένας κόμβος δεν παίρνει ξανά την πληροφορία που ο ίδιος είχε στείλει κατά το άλλο μισό της τρέχουσας επανάληψης. Αντίστοιχα, η εξωγενής πληροφορία που στέλνεται από τον CN i στον VN j, δίνεται από την παρακάτω εξίσωση:

$$L_{i\to j} = 2\tanh^{-1}\left(\prod_{j'\in N(i)-\{j\}} \tanh\left(\frac{1}{2}L_{j'\to i}\right)\right) \tag{3.9}$$

, όπου και πάλι η εξαίρεση του j-οστού VN κόμβου οφείλεται στην αρχή της εξωγενούς πληροφορίας. Στο τέλος των επαναλήψεων, ο VN j παράγει μια πρόβλεψη με βάση την ποσότητα

$$L_j^{total} = L_j + \sum_{i\in N(j)} L_{i\to j} \tag{3.10}$$

Η πληροφορία $L_{j\to i}$ που ανταλάσσεται μεταξύ των VN j και CN i, αποτελεί εκτίμηση της τιμής του v_j (πρόσημο του $L_{j\to i}$) και το επίπεδο εμπιστοσύνης της τιμής (πλάτος του $L_{j\to i}$), βασισμένη στο REP constraint για τον αντίστοιχο κόμβο. Αντίστοιχα, η εκτίμηση του CN i βασίζεται στο SPC constraint. Τέλος αναφέρεται πως για την

αρχικοποίηση, χρησιμοποιείται η εξίσωση 3.7 με την τροποποίηση που αναφέραμε παραπάνω, η οποία διαμορφώνεται διαφορετικά για κάθε τύπο καναλιού. Ενδεικτικά, για το δυαδικό AWGN κανάλι, προκύπτει η σχέση:

$$L(v_j|r_j) = \frac{2r_j}{\sigma^2} \tag{3.11}$$

Τέλος, ο αλγόριθμος απαιτεί ένα κριτήριο τερματισμού. Χρησιμοποιείται η εξίσωση

$$\mathbf{vH}^T = \mathbf{0}$$

, όπου $\mathbf{v}$ είναι μια εκτίμηση της κωδικής λέξης στο πέρας κάθε επανάληψης [26], [18], [29].

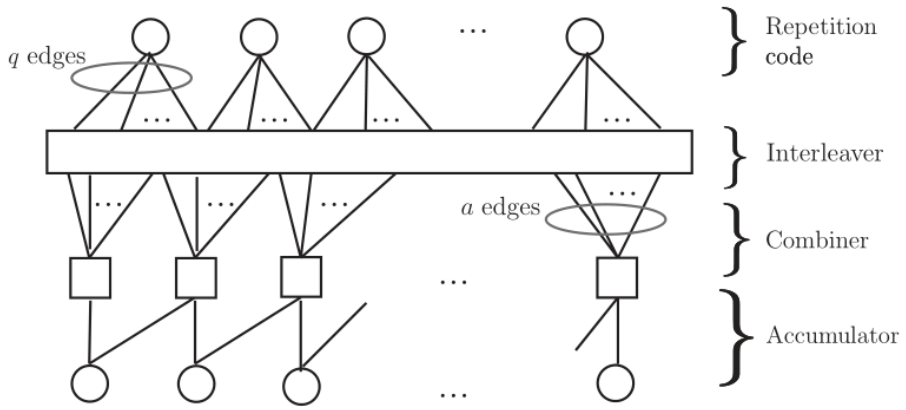
Ο Sum-Product αλγόριθμος για RA

Αναφέρθηκε ήδη πως οι εξισώσεις 3.3 και 3.4, είναι οι εξισώσεις ελέγχου ισοτιμίας για ένα RA κώδικα. Η κωδικοποίηση είναι συστηματική, συνεπώς οι πρώτες k στήλες του πίνακα $\mathbf{H}$ αντιστοιχούν στα bits πληροφορίας, ενώ οι επόμενες $m = n - k$ στήλες στα parity bits. Αναφέρθηκε επίσης πως ο πίνακας $m \times n\mathbf{H}$ είναι της μορφής

$$H = [H_1 \; H_2] \tag{3.12}$$

όπου ο υποπίνακας H_1 είναι πίνακας $n - k \times k$ με βάρη γραμμών και στηλών, (a, q) αντίστοιχα και ο H_2 οφείλεται στον συσσωρευτή.

Παρόμοια με τους LDPC κώδικες, το γράφημα Tanner των RA κωδίκων ορίζεται από τον πίνακα $\mathbf{H}$, με διαφορά ότι είναι εμφανής η διάκριση μεταξύ info bits και parity bits. Στο Σχήμα 3.10 φαίνεται το γράφημα Tanner ενός RA, στο οποίο γίνεται διάκριση μεταξύ των k συστηματικών bits βαθμού q στην κορυφή και των m parity bits κόμβων βαθμού 2 (πλην του τελευταίου που διακρίνεται από βαθμό 1). Οι κόμβοι ελέγχου του γραφήματος διακρίνονται από βαθμό $a + 2$, εκτός του τελευταίου που έχει βαθμό $a + 1$ [18].



Σχήμα 3.10: Γράφημα Tanner RA κώδικα

Στην περίπτωση που οι κόμβοι parity bits και συστηματικών bits αντιμετωπίζονται ως ενιαίοι κόμβοι bits ο αλγόριθμος SPA για RA κώδικα, ταυτίζεται με αυτόν για ένα LDPC που έχει τον ίδιο πίνακα **H**. Σε αντίθετη περίπτωση απαιτούνται τροποποιήσεις για τη συμπλήρωση μιας επανάληψης του αλγορίθμου.

Τέλος, ως σύγκριση μεταξύ του Sum-Product αποκωδικοποιητή και της turbo αποκωδικοποίησης, αναφέρεται πως, ενώ στον SPA οι parity bit κόμβοι αποκωδικοποιούνται όμοια με τους systematic bit κόμβους, ο turbo αποκωδικοποιητής χρησιμοποιεί BCJR αποκωδικοποιητή σε διάγραμμα trellis. Μέσω αυτού οι επαναλήψεις της turbo αποκωδικοποίησης είναι πιο πολύπλοκες, ωστόσο απαιτούνται λιγότερες επαναλήψεις συνολικά [26], [18].

Κεφάλαιο 4

Προσομοίωση DVB-S2 RA κώδικα σε κανάλι AWGN

Στο κεφάλαιο αυτό, θα παρουσιαστεί ο δέκτης που προσομοιώθηκε και τα αποτελέσματα της προσομοίωσης σε καμπύλες BER/FER για τους διάφορους ρυθμούς που έχουν προτυποποιηθεί για την ψηφιακή τηλεόραση δεύτερης γενιάς (DVB-S/T2).

Αρχικά παρουσιάζεται συνοπτικά η διαμόρφωση QPSK και κατόπιν η διαδικασία αποδιαμόρφωσης (demapping), μέσω έτοιμων συναρτήσεων που παρέχονται από το MATLAB. Στη συνέχεια παρουσιάζεται το σύνολο της προσομοίωσης και στο τέλος του κεφαλαίου, τα αποτελέσματά της.

4.1 QPSK Demapper

4.1.1 Διαμόρφωση QPSK

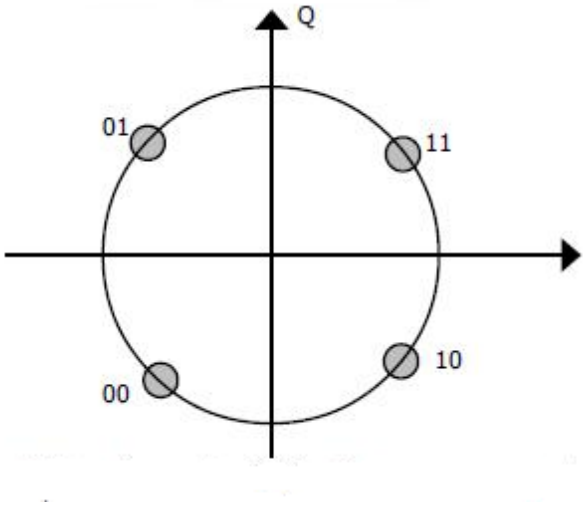
Η κωδική λέξη, μετά την έξοδό της από τον κωδικοποιητή καναλιού, απεικονίζεται στον QPSK αστερισμό και προκύπτει η ακολουθία μιγαδικών σημάτων, μήκους $n/2$. Σε μορφή ημιτονοειδών κυμάτων μετάδοσης, ο QPSK αστερισμός γράφεται ως εξής:

$$s_n(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f_c t + (2m-1)\frac{\pi}{4}\right), \quad m = 1, 2, 3, 4. \tag{4.1}$$

όπου E_s η ενέργεια συμβόλου, T_s η περίοδος σηματοδοσίας, f_c η συχνότητα του φέροντος κύματος και m ο δείκτης των συμβόλων. Το αποτέλεσμα είναι ένας δι-διάστατος χώρος σημάτων με μοναδιαίες συναρτήσεις βάσης:

$$\begin{aligned} \phi_1 &= \sqrt{\frac{2}{T_s}} \cos(2\pi f_c t) \\ \phi_2 &= \sqrt{\frac{2}{T_s}} \sin(2\pi f_c t) \end{aligned} \tag{4.2}$$

όπου ϕ_1 είναι η συμφασική και ϕ_2 η ορθογώνια συνιστώσα. Το σήμα αποτυπώνεται στα 4 σημεία του αστερισμού QPSK $\left(\pm\sqrt{E_s/2}, \pm\sqrt{E_s/2}\right)$ και οι φάσεις των συμβόλων προκύπτουν $\pi/4$, $3\pi/4$, $5\pi/4$ και $7\pi/4$. Ο παραπάνω αστερισμός φαίνεται στο Σχήμα 4.1:



Σχήμα 4.1: Ο αστερισμός QPSK

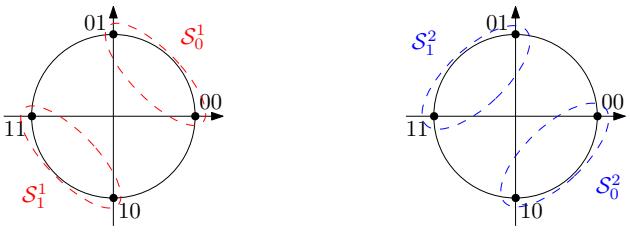
Στη συνέχεια το σήμα περνάει από το κανάλι μετάδοσης, όπου προστίθεται το AWGN θόρυβος. Στο δέκτη ακολουθείται η αντίστροφη διαδικασία. Το Matlab παρέχει έτοιμη μέθοδο αποδιαμόρφωσης και ανίχνευσης μέσω του αντικειμένου comm.QPSKDemodulator και της μεθόδου step.

Το αντικείμενο αυτό δουλεύει ως εξής: μετά την έξοδο από το κανάλι η ακολουθία μιγαδικών αριθμών $\mathbf{y}$ μήκους $n/2$ εισάγεται στον demapper, ο οποίος παράγει μετρικές $L(x_{ij})$ στο διάστημα $(-\infty, \infty)$ για καθένα από τα κωδικά bits x_{ij}, σύμφωνα με τη σχέση:

$$L(x_{ij}) = \ln \frac{p(x_{ij} = 0 \mid \mathbf{y}_i)}{p(x_{ij} = 1 \mid \mathbf{y}_i)}, \quad i \in \left[1, \frac{n}{2}\right], j = 1, 2. \tag{4.3}$$

οι οποίες εισάγονται στον επαναληπτικό αποκωδικοποιητή για την αρχικοποίησή του.

Από τις εξισώσεις 4.1, 4.2 προκύπτει πως η διαμόρφωση QPSK αποτελεί απεικόνιση των δυάδων $\{0,1\}^2$, οι οποίες καλούνται *ετικέτες* (labels), στο σύνολο των QPSK σημάτων S, δηλαδή $\{0,1\}^2 \to S$. Αν συμβολιστεί ως S_0^j το υποσύνολο των στοιχείων του S που στην j-οστή θέση της ετικέτας τους έχουν 0 και ως S_1^j το υποσύνολο των στοιχείων του S που στην j-οστή θέση της ετικέτας τους έχουν 1 και θεωρηθεί απεικόνιση Gray, προκύπτουν σχηματικά τα υποσύνολα του Σχήματος 4.2.



Σχήμα 4.2: Τα υποσύνολα S_1^0, S_1^1 και S_0^0, S_0^1

Η σχέση 4.3 μπορεί να αναλυθεί περεταίρω για το κωδικό bit $L(x_{i1})$, αν εφαρμοστεί ο νόμος ολικής πιθανότητας, επιμερίζοντας σε όλα τα γεγονότα x_{i2}, το γεγονός $\{x_{i1} = 0\}$ στον αριθμητή και το $\{x_{i1} = 1\}$ στον παρονομαστή, ώστε να πάρει την μορφή της εξίσωσης 4.4:

$$\begin{aligned} L(x_{i1}) &= \ln \frac{\sum_{x_{i2}} p(x_{i1} = 0, x_{i2} \mid \mathbf{y}_i)}{\sum_{x_{i2}} p(x_{i1} = 1, x_{i2} \mid \mathbf{y}_i)} \\ &= \ln \frac{\sum_{\mathbf{s} \in S_0^1} p(\mathbf{s} \mid \mathbf{y}_i)}{\sum_{\mathbf{s} \in S_1^1} p(\mathbf{s} \mid \mathbf{y}_i)} \end{aligned} \tag{4.4}$$

Γενικεύοντας από την εξίσωση 4.4, για το κωδικό bit $L(x_{ij})$, προκύπτει:

$$\begin{aligned} L(x_{ij}) &= \ln \frac{\sum_{\mathbf{s} \in S_0^j} p(\mathbf{s} \mid \mathbf{y}_i)}{\sum_{\mathbf{s} \in S_1^j} p(\mathbf{s} \mid \mathbf{y}_i)} \\ &= \ln \frac{\sum_{\mathbf{s} \in S_0^j} p(\mathbf{y}_i \mid \mathbf{s}) p(\mathbf{s})}{\sum_{\mathbf{s} \in S_1^j} p(\mathbf{y}_i \mid \mathbf{s}) p(\mathbf{s})} \\ &= \ln \frac{\sum_{\mathbf{s} \in S_0^j} p(\mathbf{y}_i \mid \mathbf{s})}{\sum_{\mathbf{s} \in S_1^j} p(\mathbf{y}_i \mid \mathbf{s})} \end{aligned} \tag{4.5}$$

Οι ισότητες λογαρίθμων στην εξίσωση 4.5 προκύπτουν χρησιμοποιώντας τον κανόνα του Bayes και υποθέτοντας ότι τα σήματα $\mathbf{s} \in S$ είναι ισοπίθανα.

Ακόμη για το διακριτό κανάλι AWGN, η κατανομή πυκνότητας πιθανότητας $p(\mathbf{y}_i \mid \mathbf{s})$ ορίζεται ως εξής:

$$p(\mathbf{y}_i \mid \mathbf{s}) = \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{\|\mathbf{y}_i - \mathbf{s}\|^2}{2\sigma^2} \right\} \tag{4.6}$$

Συνοψίζοντας, η εξίσωση 4.4, λόγω των 4.5, 4.6 γίνεται:

$$
\begin{aligned}
L(x_{ij}) &= \ln \frac{\sum_{\mathbf{s}\in S_0^j} \exp\left\{-\frac{\|\mathbf{y}_i\|^2 - 2\Re(\mathbf{y}_i\cdot\mathbf{s}) + \|\mathbf{s}\|^2}{2\sigma^2}\right\}}{\sum_{\mathbf{s}\in S_1^j} \exp\left\{-\frac{\|\mathbf{y}_i\|^2 - 2\Re(\mathbf{y}_i\cdot\mathbf{s}) + \|\mathbf{s}\|^2}{2\sigma^2}\right\}} \\
&= \ln \frac{\sum_{\mathbf{s}\in S_0^j} \exp\left\{\frac{\Re(\mathbf{y}_i\cdot\mathbf{s})}{\sigma^2}\right\}}{\sum_{\mathbf{s}\in S_1^j} \exp\left\{\frac{\Re(\mathbf{y}_i\cdot\mathbf{s})}{\sigma^2}\right\}}
\end{aligned}
\tag{4.7}
$$

Στην εξίσωση 4.7, θεωρώντας πως η ενέργεια σήματος $\|\mathbf{s}\|^2 = E_s,\ \forall\, \mathbf{s} \in S$ είναι η ίδια, οι παράγοντες $\exp\left\{-\frac{\|\mathbf{y}_i\|^2}{2\sigma^2}\right\}$, $\exp\left\{-\frac{\|\mathbf{s}\|^2}{2\sigma^2}\right\}$ μπορούν να απαλειφθούν με παραγοντοποίηση σε αριθμητή και παρονομαστή και τελικά η εξίσωση 4.7 να λάβει τη μορφή:

$$
L(x_{ij}) = \ln \frac{\sum_{\mathbf{s}\in S_0^j} \exp\left\{\frac{y_{iI}s_I + y_{iQ}s_Q}{\sigma^2}\right\}}{\sum_{\mathbf{s}\in S_1^j} \exp\left\{\frac{y_{iI}s_I + y_{iQ}s_Q}{\sigma^2}\right\}} \tag{4.8}
$$

όπου οι δείκτες I, Q υποδεικνύουν προβολή του αντίστοιχου διανύσματος στη συμφασική και στην ορθογώνια συνιστώσα αντίστοιχα.

4.2 Προσομοίωση στο Matlab

Rate R	$(E_b/N_0)_{soft}$ (dB)
1/4	-0.793
1/3	-0.497
2/5	-0.236
1/2	0.187
3/5	0.682
2/3	1.059
3/4	1.626
4/5	2.039
9/10	3.199

Πίνακας 4.1: Χωρητικότητα ως E_b/N_0 για τους ρυθμούς κώδικα της προσομοίωσης

Στον πίνακα 4.1, φαίνεται το (θεωρητικό) κάτω όριο E_b/N_0 για τον κάθε ρυθμό κώδικα που προσομοιώνεται, το οποίο καλείται *όριο κωδικοποίησης* και αντιστοιχεί σε διαμόρφωση BPSK ή QPSK. Η χωρητικότητα, όπως εκφράζεται από τα παραπάνω όρια για δεδομένο ρυθμό R, δίνει ένα κατώφλι SNR μετά από το οποίο και με τη χρήση κωδικοποίησης καναλιού, μπορεί να επιτευχθεί -θεωρητικά- αξιόπιστη επικοινωνία, σε διαφορετική περίπτωση η αξιοπιστία της οποίας είναι μη ελέγξιμη.

Για να υλοποιηθεί προσομοίωση στο Matlab μεταβάλλεται η μεταβλητή R η οποία αποθηκεύει το ρυθμό του κώδικα και ορίζεται ο LDPC κώδικας με βάση τον πίνακα **H**. Κατόπιν αρχικοποιούνται τα αντικείμενα που ορίζουν τις παραμέτρους της διαμόρφωσης-αποδιαμόρφωσης και κωδικοποίησης-αποκωδικοποίησης.

```
R = 1/2;
H = dvbs2ldpc(R);
```

```
enc = comm.LDPCEncoder(H);
dec = comm.LDPCDecoder('ParityCheckMatrix', H, ...
                       'DecisionMethod', 'Hard decision', ...
                       'IterationTerminationCondition', 'Parity ...
                          check satisfied');
```

```
mod = comm.QPSKModulator('BitInput', true);
deMod = comm.QPSKDemodulator('BitOutput', true, ...
                             'DecisionMethod', 'Log-likelihood ...
                                ratio', ...
                             'Variance', 1);
```

Η κωδικοποίηση γίνεται σύμφωνα με όσα αναφέρθηκαν στο Κεφάλαιο 3 και ακολουθεί την συστηματική LDPC, σύμφωνα με το πρότυπο Digital Video Broadcasting & Satellite - Second Generation (DVB-S2) [10]. Όπως φαίνεται από το παραπάνω κομμάτι κώδικα, το αντικείμενο *enc* αρχικοποιείται με βάσει τον πίνακα ελέγου ισοτιμίας για δεδομένο ρυθμό κώδικα. Διακρίνεται από τα εξής βήματα:

- Αρχικοποίηση των parity bits: $p_0 = p_1 = \cdots = p_{n-k-1} = 0$
- Συσσώρευση του πρώτου info bit στις διευθύνσεις που δίνονται από την πρώτη γραμμή των πινάκων B.1 έως B.11 του παραρτήματος B στο πρότυπο ETSI για το DVB-S2 [10], με mod-2 πρόσθεση.
- Για τα επόμενα 359 info bits οι διευθύνσεις των συσσωρευτών δίνονται από τον τύπο $\{x + (m \bmod 360) \times q\} mod\,(n-k)$, όπου $m = 1, 2, \cdots, 359$, το x αντιστοιχεί στη διεύθυνση συσσωρευτή του πρώτου info bit και το q δίνεται από τον Πίνακα 4.2

- Αντίστοιχα, για κάθε επόμενο group από 360 info bits χρησιμοποιείται μια καινούργια γραμμή διευθύνσεων στους πίνακες B.1 έως B.11 του παραρτήματος B του [10]

- Με την εξάντληση των info bits, τα τελικά parity bits p_i δίνονται ως εξής:

$$p_0 = p_0$$

$$p_i = p_i \oplus p_{i-1}, \quad i = 1, 2, \cdots, n-k$$

Code Rate	q
1/4	135
1/3	120
2/5	108
1/2	90
3/5	72
2/3	60
3/4	45
4/5	36
9/10	18

Πίνακας 4.2: Τιμές q για τους διάφορους ρυθμούς κώδικα της προσομοίωσης

Αντίστοιχα η αποκωδικοποίηση στηρίζεται στην εφαρμογή του SPA αλγορίθμου για LDPC και ακολουθεί την παρακάτω πορεία:

- Η αρχικοποίηση των L_j γινεται μέσω της σχέσης 4.8

- Τα μηνύματα που εξέρχονται απο τους CN υπολογίζονται από την εξίσωση

$$L_{i\to j} = 2\tanh^{-1}\left(\prod_{j'\in N(i)-\{j\}} \tanh\left(\frac{1}{2}L_{j'\to i}\right)\right) \tag{4.9}$$

  και κατόπιν μεταδίδονται στους VN

- Τα μηνύματα που εξέρχονται απο τους VN υπολογίζονται από την εξίσωση

$$L_{j\to i} = L_j + \sum_{i'\in N(j)-\{i\}} L_{i'\to j}$$

  και κατόπιν μεταδίδονται στους CN

- Τα συνολικα LLR υπολογίζονται από τη σχέση

$$L_j^{total} = L_j + \sum_{i \in N(j)} L_{i \to j}$$

- Για $j = 0, 1, \cdots, n-1$ υπολογίζονται οι τιμές των bit

$$\hat{v_j} = \begin{cases} 1 & L_j^{total} < 0 \\ 0 & else \end{cases}$$

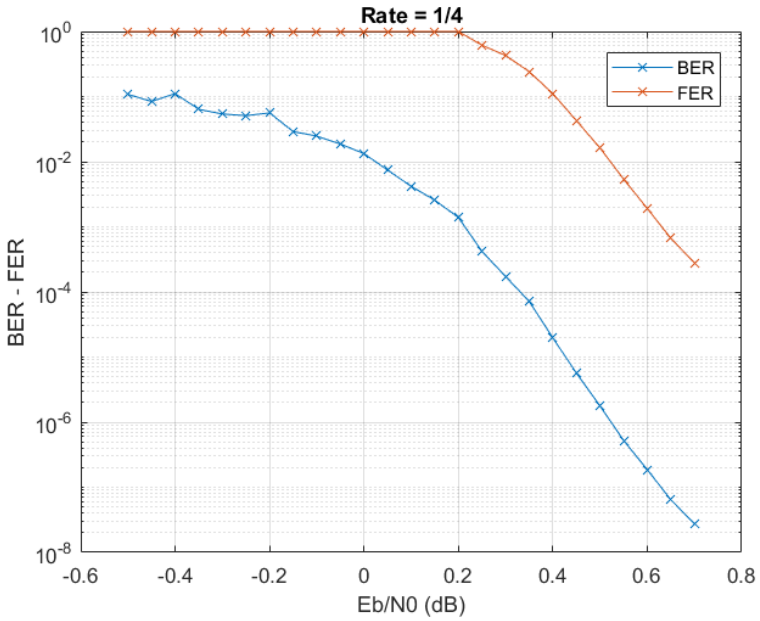
  που διαμορφώνουν την $\hat{\mathbf{v}}$. Αν $\hat{\mathbf{v}}\mathbf{H} = \mathbf{0}$, ο αλγόριθμος τερματίζει, αλλιώς συνεχίζει με την επόμενη επανάληψη.

- Στο τέλος της προσομοίωσης υπολογίζονται οι μετρικές BER και FER για την προσομοίωση του εκάστοτε ρυθμού στο αντίστοιχο SNR

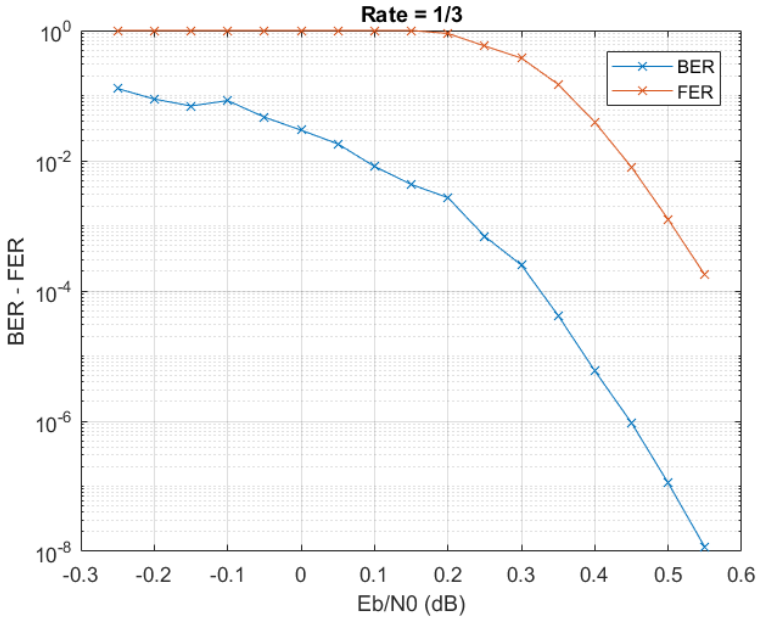
Η προσομοίωση έχει ως όριο τα 10^3 σφάλματα ή τις 10^6 προσπάθειες (trials) να εντοπιστούν. Τα αποτελέσματα που θα παρουσιαστούν, λαμβάνονται αφού αποθηκευτούν για κάθε ρυθμό το Bit Error Rate - BER και το Frame Error Rate - FER.

4.3 Αποτελέσματα - Σχολιασμός

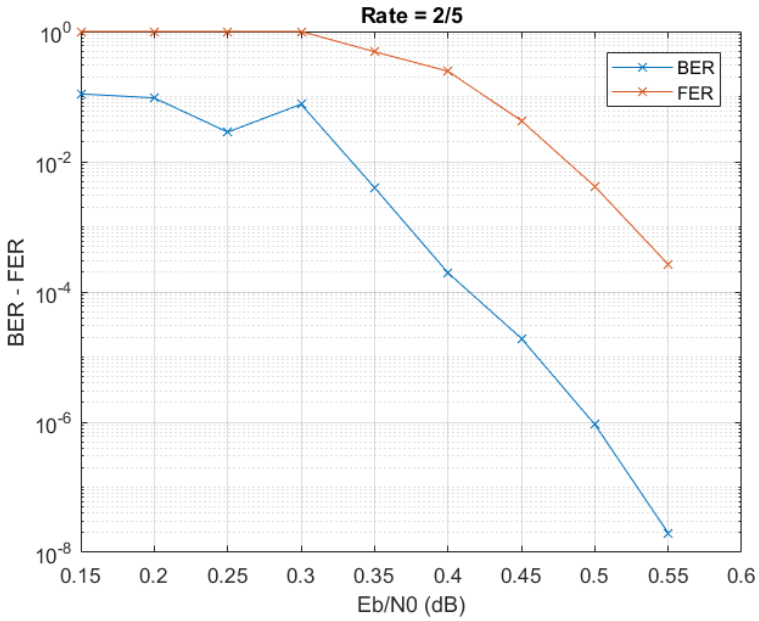
Σε αυτό το σημείο παρουσιάζονται τα αποτελέσματα της προσομοίωσης. Τα διαγράμματα που ακολουθούν απεικονίζουν για κάθε ρυθμό του Πίνακα 4.1 τις τιμές του BER, ξεκινώντας από την τιμή E_b/N_0 του Πίνακα 4.1 (θεωρητικό όριο χωρητικότητας για κάθε ρυθμό) και καταλήγοντας στη τιμή E_b/N_0, για την οποία το BER φτάνει στο 10^{-8}. Στα ίδια διαγράμματα απεικονίζονται και οι τιμές FER για τις αντίστοιχες τιμές E_b/N_0. Στο τέλος, δίνεται επίσης ένα συγκριτικό διάγραμμα όλων των καμπυλών BER και FER, για καλύτερη εποπτεία των αποτελεσμάτων των διαφόρων ρυθμών.



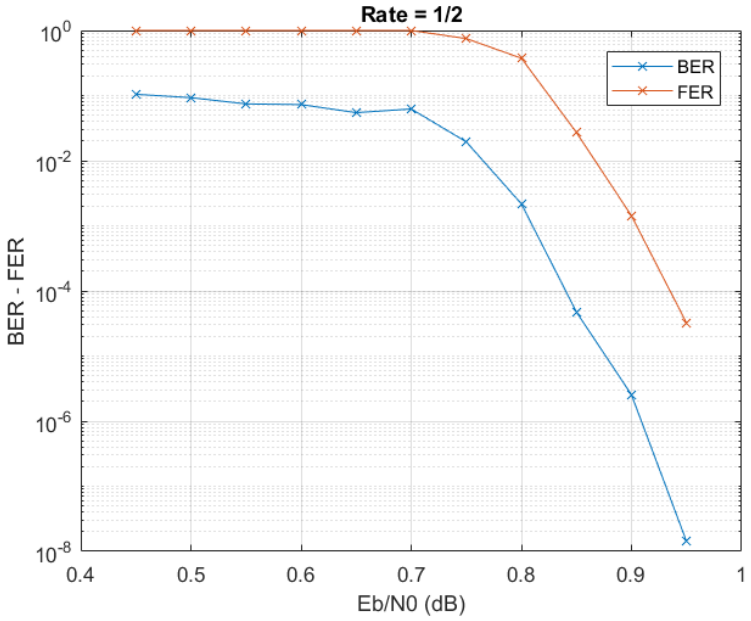
Σχήμα 4.3: BER,FER vs E_b/N_0 για ρυθμό 1/4



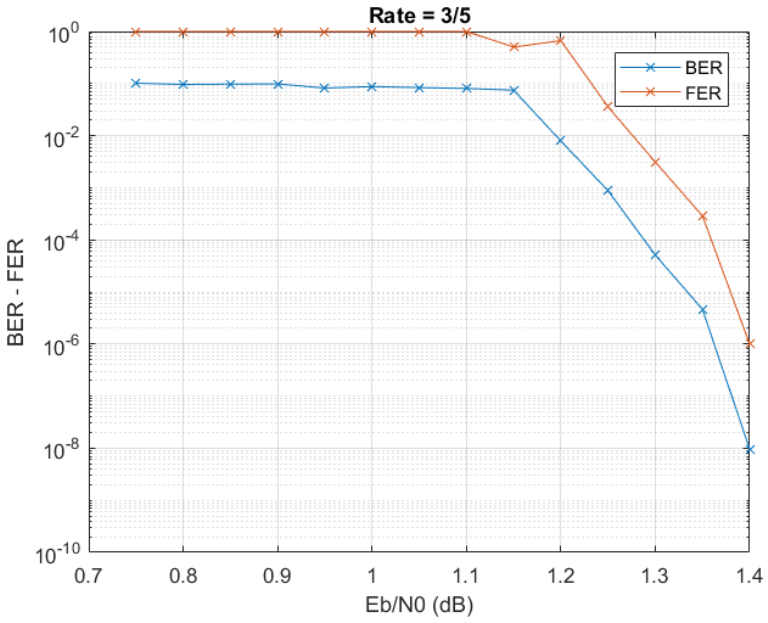
Σχήμα 4.4: BER,FER vs E_b/N_0 για ρυθμό 1/3



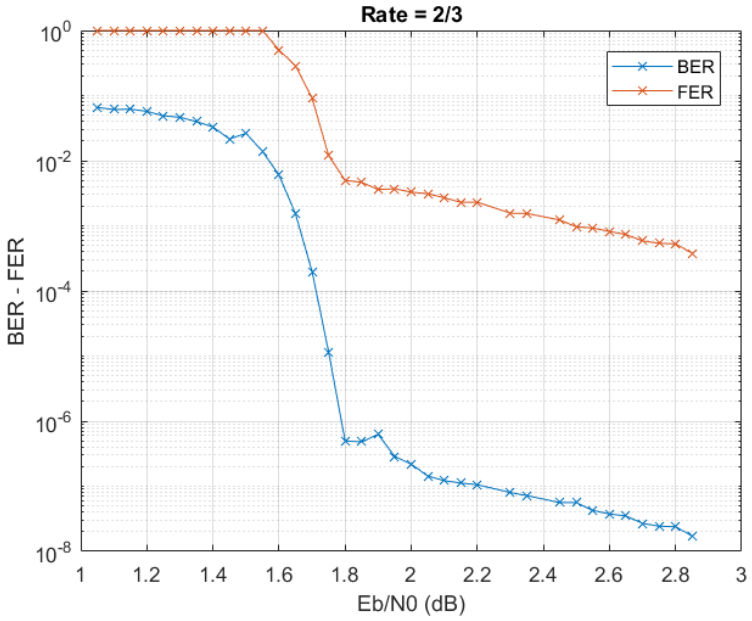
Σχήμα 4.5: BER,FER vs E_b/N_0 για ρυθμό 2/5



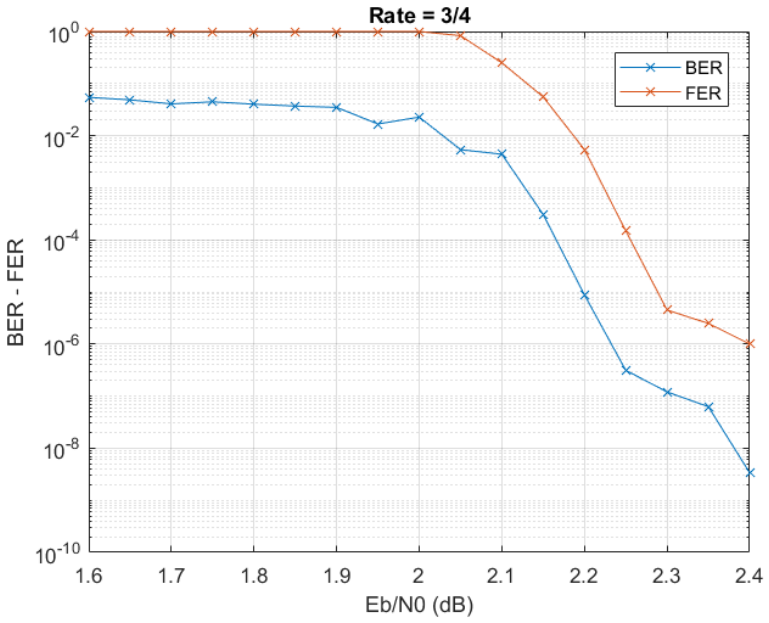
Σχήμα 4.6: BER,FER vs E_b/N_0 για ρυθμό 1/2



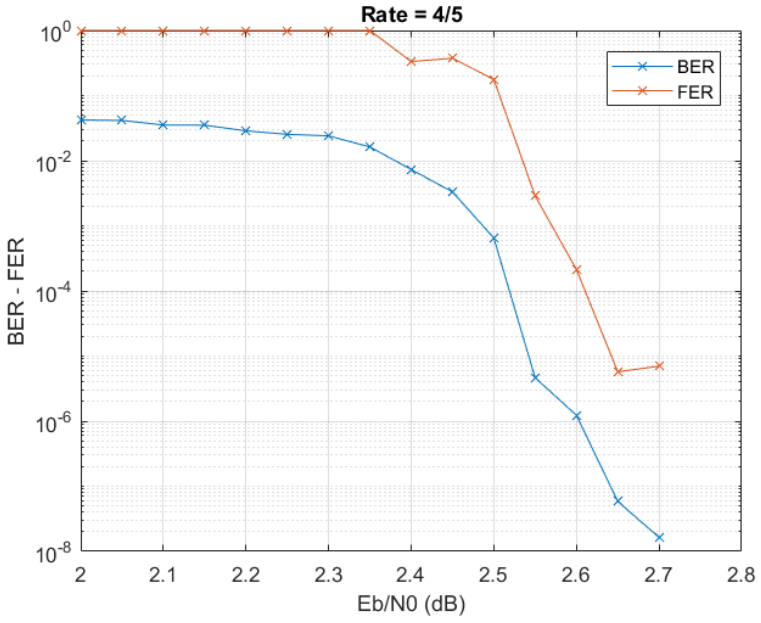
Σχήμα 4.7: BER,FER vs E_b/N_0 για ρυθμό 3/5



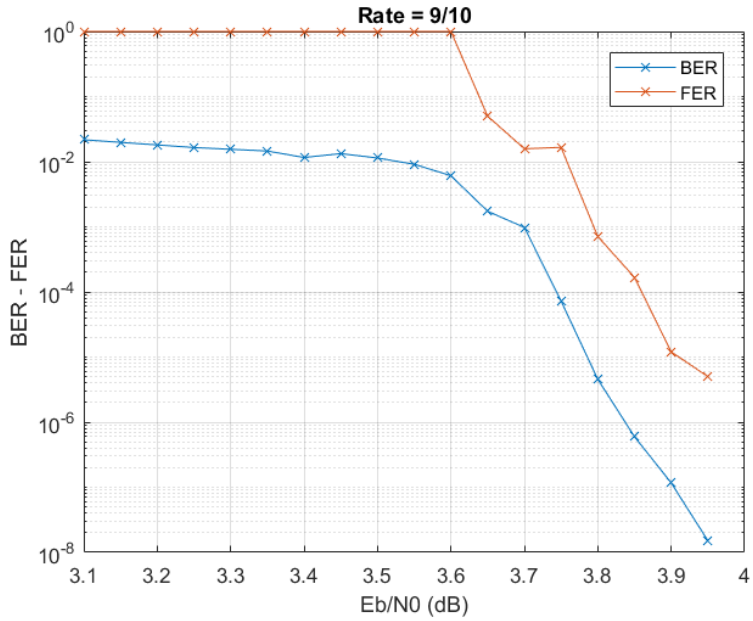
Σχήμα 4.8: BER,FER vs E_b/N_0 για ρυθμό 2/3



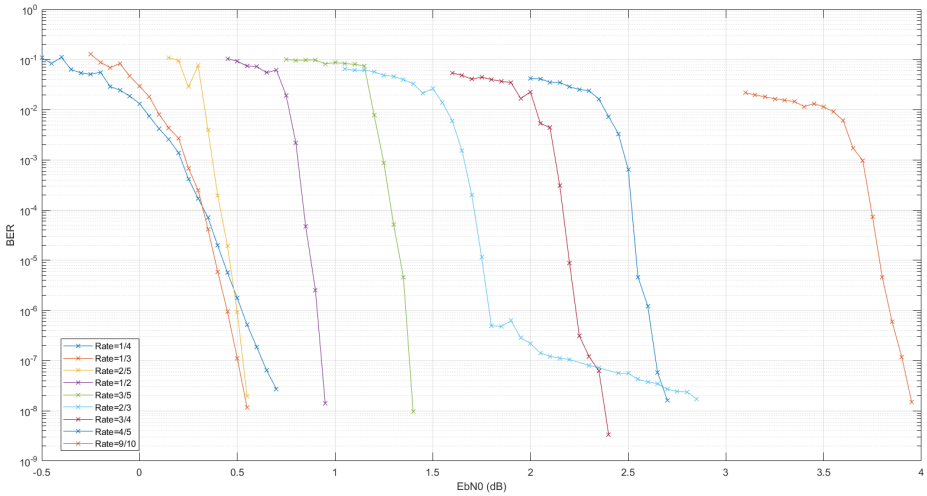
Σχήμα 4.9: BER,FER vs E_b/N_0 για ρυθμό 3/4



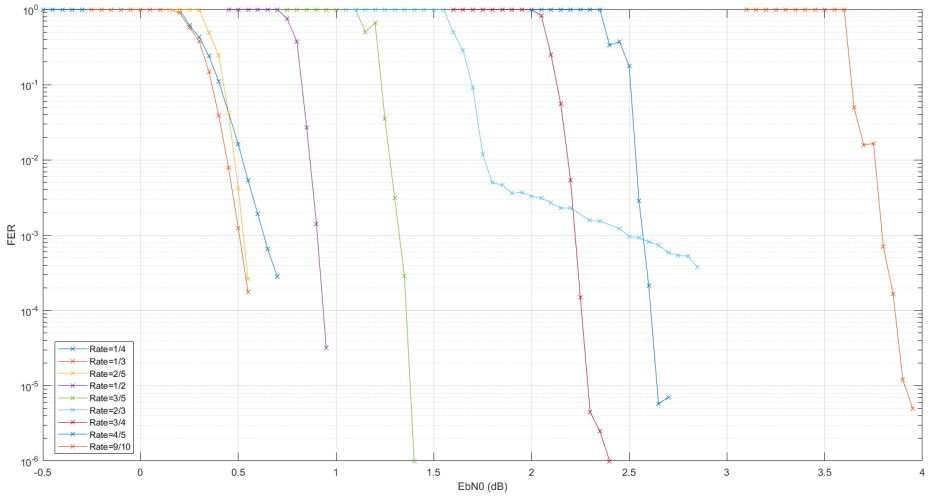
Σχήμα 4.10: BER,FER vs E_b/N_0 για ρυθμό 4/5



Σχήμα 4.11: BER,FER vs E_b/N_0 για ρυθμό 9/10



Σχήμα 4.12: Καμπύλες BER vs E_b/N_0 για διαφορετικούς ρυθμούς



Σχήμα 4.13: Καμπύλες FER vs E_b/N_0 για διαφορετικούς ρυθμούς

4.3.1 Σχολιασμός

Αρχικά παρατηρείται πως οι καμπύλες δεν είναι όσο λείες θα επιθυμούσαμε. Αυτό οφείλεται στον αριθμό των λαθών στα οποία σταματούσαμε την προσομοίωση, και ο οποίος σε κάποιες περιπτώσεις θα έπρεπε να είναι μεγαλύτερος. Αυτό θα βελτίωνε την αξιοπιστία της αντίστοιχης μέτρησης.

Ακόμη εντυπωσιάζει σε κάθε περίπτωση το ότι με πολύ μικρή αύξηση του SNR βελτιώνεται ραγδαία ο ρυθμός σφαλμάτων. Επίσης, για όλους τους ρυθμούς επιβεβαιώνεται το γεγονός πως οι RA κώδικες του προτύπου DVB-S2 είναι capacity-approaching κώδικες, μιας και μιας και απέχουν από τις χωρητικότητές τους απόσταση μικρότερη του 1 dB. Η μόνη περίπτωση στην οποία αυτό δεν ισχύει, είναι στον κώδικα με ρυθμό 1/4.

Τέλος, παρατηρούμε το έντονο error-floor του κώδικα ρυθμού 2/3 για τιμές του BER απο 10^{-6} και μετά. Φαινόμενα σαν κι αυτό επιλύονται με τη σειριακή αλύσωση του RA κώδικα με έναν εξωτερικό, κάτι που πράγματι συμβαίνει στο πρότυπο DVB-S2, όπου αυτός ο εξωτερικός είναι ένας BCH κώδικας.

Κεφάλαιο 5

Μελλοντικές επεκτάσεις

Στο Κεφάλαιο 3, αναφέρθηκε πως οι VNs είναι ανεξάρτητοι κόμβοι που μπορούν ο καθένας να υπολογίζει τα εξαγόμενα LLR παράλληλα με τους υπόλοιπους. Το ίδιο ισχύει και για τους CNs. Άρα ο αλγόριθμος SPA είναι από τη φύση τους έντονα παραλληλοποιήσιμος. Καθώς οι ασύρματες συσκευές μεταδίδουν και δέχονται δεδομένα υψηλού ρυθμού σε πραγματικό χρόνο, αυξάνεται ραγδαία η ανάγκη για ταχύτητα και αξιοπιστία στην επικοινωνία.

Η χρήση των LDPC σε πολλά νέα πρότυπα (DVB-S2, WiMAX (802.16e), Wifi (802.11n), 10 Gbit Ethernet (802.3an), 5G) κ.λ.π. [12], καθώς και σε πολλαπλούς ρυθμούς δεδομένων για τα διαφορετικά πρότυπα, έχουν κάνει ορατή τη δυσκολία για υλοποίηση hardware, καθώς η επεξεργασία βασικής ζώνης στο επίπεδο του φυσικού εξαρτήματος είναι πολυδάπανη σε εύρος ζώνης και επεξεργαστική ισχύ.

Αντ'αυτού, ο σχεδιασμός συστημάτων ψηφιακής τηλεπικοινωνίας υιοθετεί όλο και περισσότερο υλοποιήσεις σε software, μέσω της χρήσης CPUs ή/και GPU [21], [1]. Σχετική έρευνα, έχει δείξει τη διαφορά στη χρήση GPU σε σχέση με κυκλώματα ASIC για LDPC αποκωδικοποίηση [11].

5.1 Προσομοίωση σε CUDA

Η χρήση προγραμματισμού σε GPU προσφέρει υψηλή υπολογιστική ισχύ, καθώς οι μονάδες επεξεργασίας γραφικών αποκτούν ολοένα και καλύτερες επιδόσεις. Προς την κατεύθυνση αυτή, αξιοσημείωτο ενδιαφέρον παρουσιάζει ο προγραμματισμός με βάση την αρχιτεκτονική Compute Unified Device Architecture (CUDA) της εταιρίας nVidia.

Σχετική έρευνα έχει υπάρξει αναφορικά με τη χρήση GPU για τη διαχείριση του SPA και την εξαγωγή πληροφορίας από τις μετρικές LLR [13]. Ακόμη, υπάρχει προτεινόμενος τρόπος για την υλοποίηση CUDA για χρήση LDPC αποκωδικοποίησης [12], καθώς και υβριδική συνδυαστική χρήση multicore CPU - GPU, για την ενσωμάτωση

πολλαπλών ρυθμών και τηλεπικοινωνιακών προτύπων [21].

Με βάση, λοιπόν, τα παραπάνω αποκτά ενδιαφέρον η μελέτη προς την κατεύθυνση της ταχύτερης υλοποίησης RA αποκωδικοποιητών κάνοντας χρήση της αρχιτεκτονικής CUDA, καθώς και η μέτρηση και παρουσίαση των επιδόσεών τους.

Βιβλιογραφία

[1] Kiran Kumar Abburi. A scalable ldpc decoder on gpu. Στο *VLSI design (VLSI design), 2011 24th international conference on*, σελίδες 183–188. IEEE, 2011.

[2] Silvio A Abrantes. From bcjr to turbo decoding: Map algorithms made easier. 2004.

[3] Elwyn Berlekamp, Robert McEliece και Henk Van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[4] Claude Berrou, Alain Glavieux και Punya Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. Στο *Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, τόμος 2, σελίδες 1064–1070. IEEE, 1993.

[5] Jehoshua Bruck και Moni Naor. The hardness of decoding linear codes with preprocessing. *IEEE Transactions on Information Theory*, 36(2):381–385, 1990.

[6] Sae Young Chung, G David Forney, Thomas J Richardson και Rüdiger Urbanke. On the design of low-density parity-check codes within 0.0045 db of the shannon limit. *IEEE Communications letters*, 5(2):58–60, 2001.

[7] Capacity Approaching Codes. Guest editorial capacity approaching codes. *IEEE Journal on Selected Areas in Communications*, 27(6):825, 2009.

[8] Thomas M Cover και Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[9] Dariush Divsalar, Hui Jin και Robert J McEliece. Coding theorems for" turbo-like" codes. Στο *Proceedings of the annual Allerton Conference on Communication control and Computing*, τόμος 36, σελίδες 201–210. UNIVERSITY OF ILLINOIS, 1998.

[10] EN ETSI. 302 307 v1. 2.1 (2009-08). *Digital video broadcasting (DVB)*, σελίδες 1–78, 2009.

[11] Gabriel Falcão, Vitor Silva και Leonel Sousa. How gpus can outperform asics for fast ldpc decoding. Στο *Proceedings of the 23rd international conference on Supercomputing*, σελίδες 390–399. ACM, 2009.

[12] Gabriel Falcao, Leonel Sousa και Vitor Silva. Massively ldpc decoding on multicore architectures. *IEEE Transactions on Parallel and Distributed Systems*, 22(2):309–322, 2011.

[13] Gabriel Falcão, Shinichi Yamagiwa, Vitor Silva και Leonel Sousa. Parallel ldpc decoding on gpus using a stream-based computing approach. *Journal of computer science and technology*, 24(5):913–924, 2009.

[14] Robert Gallager. Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28, 1962.

[15] Joachim Hagenauer και Norbert Gortz. The turbo principle in joint source-channel coding. Στο *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*, σελίδες 275–278. IEEE, 2003.

[16] Joachim Hagenauer, Elke Offer και Lutz Papke. Iterative decoding of binary block and convolutional codes. *IEEE Transactions on information theory*, 42(2):429–445, 1996.

[17] CC Heyde. Central limit theorem. *Encyclopedia of Actuarial Science*, 2006.

[18] Sarah J Johnson. *Iterative error correction: Turbo, low-density parity-check and repeat-accumulate codes*. Cambridge University Press, 2009.

[19] David JC MacKay και Radford M Neal. Near shannon limit performance of low density parity check codes. *Electronics letters*, 32(18):1645, 1996.

[20] Florence Jessie MacWilliams και Neil James Alexander Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[21] Joo Yul Park και Ki Seok Chung. Parallel ldpc decoding using cuda and openmp. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):172, 2011.

[22] William Wesley Peterson και Edward J Weldon. *Error-correcting codes*. MIT press, 1972.

[23] Henry D Pfister, Igal Sason και Rudiger Urbanke. Capacity-achieving ensembles for the binary erasure channel with bounded complexity. *IEEE Transactions on Information Theory*, 51(7):2352–2379, 2005.

[24] John G Proakis, Masoud Salehi, Ning Zhou και Xiaofeng Li. *Communication systems engineering*, τόμος 2. Prentice Hall New Jersey, 1994.

[25] Tom Richardson και Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008.

[26] William Ryan και Shu Lin. *Channel codes: classical and modern*. Cambridge University Press, 2009.

[27] Claude E Shannon. A mathematical theory of communication, part i, part ii. *Bell Syst. Tech. J.*, 27:623–656, 1948.

[28] R Tanner. A recursive approach to low complexity codes. *IEEE Transactions on information theory*, 27(5):533–547, 1981.

[29] Tuan Ta. A tutorial on low density parity-check codes. *the university of Texas at Austin*, 4, 2013.