ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αρχιτεκτονικές υλικού για αποκωδικοποιητές LDPC βέλτιστης πληροφορίας

Δ Ι Π Λ Ω Μ Α Τ Ι Κ Η Ε Ρ Γ Α Σ Ι Α

ΠΑΝΤΕΛΕΗΜΩΝ ΦΩΤΙΑΔΗΣ

ΕΠΙΒΛΕΠΩΝ: ΒΑΣΙΛΗΣ ΠΑΛΙΟΥΡΑΣ

ΠΑΤΡΑ – ΟΚΤΩΒΡΙΟΣ 2021

Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών.

Παντελεήμων Φωτιάδης

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με τίτλο

Αρχιτεκτονικές υλικού για αποκωδικοποιητές LDPC βέλτιστης πληροφορίας

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

ΠΑΝΤΕΛΕΗΜΟΝΑ ΦΩΤΙΑΔΗ ΤΟΥ **ΔΗΜΗΤΡΙΟΥ**

Αριθμός Μητρώου: 1020686

Παρουσιάστηκε δημόσια στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

25/10/2021

και εξετάστηκε από την ακόλουθη εξεταστική επιτροπή:

Βασίλης Παλιουράς, καθηγητής, ΗΜΤΥ (επιβλέπων)

Γεώργιος Θεοδωρίδης, επίκουρος καθηγητής, ΗΜΤΥ (μέλος επιτροπής)

Μιχάλης Μπίρμπας, επίκουρος καθηγητής, ΗΜΤΥ (μέλος επιτροπής)

Ο Επιβλέπων	Ο Διευθυντής του Τομέα
Παλιουράς Βασίλης Καθηγητής	Καλύβας Γρηγόριος Καθηγητής

ΠΡΟΛΟΓΟΣ

Θα ήθελα να ευχαριστήσω τον κύριο Βασίλη Παλιουρά για την άψογη συνεργασία και την πολύτιμη βοήθειά του στην εκπόνηση αυτής της εργασίας. Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την στήριξή τους και την δύναμη που μου έδωσαν για την ολοκλήρωση της εργασίας.

ΠΕΡΙΛΗΨΗ

Αρχιτεκτονικές υλικού για αποκωδικοποιητές LDPC βέλτιστης πληροφορίας

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ:
ΠΑΝΤΕΛΕΗΜΩΝ ΦΩΤΙΑΔΗΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΟΣ:
ΒΑΣΙΛΗΣ ΠΑΛΙΟΥΡΑΣ

Οι κώδικες LDPC ανήκουν στην κατηγορία των block κωδίκων. Πρόκειται για κώδικες ελέγχου σφαλμάτων μετάδοσης και πιο συγκεκριμένα για κώδικες διόρθωσης σφαλμάτων. Αν και αναπτύχθηκαν από τον Gallager στις αρχές της δεκαετίας του 60, μόλις τα τελευταία χρόνια έχουν καταφέρει να κεντρίσουν το ενδιαφέρον της επιστημονικής και ερευνητικής κοινότητας λόγω και της ραγδαίας ανάπτυξης στο κομμάτι της αρχιτεκτονικής υλικού που επιτρέπει την πιο εύκολη υλοποίησή τους παρά την μεγάλη υπολογιστή τους πολυπλοκότητα. Είναι κώδικες ελέγχου ισοτιμίας με κυριότερο χαρακτηριστικό τον χαμηλής πυκνότητας πίνακα ελέγχου ισοτιμίας (Low Density Parity Check) από τον οποίο πήραν και το όνομά τους. Δεδομένου ότι η κωδικοποίηση των συγκεκριμένων κωδίκων είναι σχετικά απλή, η αποκωδικοποίησή τους είναι εκείνη που καθορίζει σε μεγάλο βαθμό τα χαρακτηριστικά του κώδικα που μας ενδιαφέρουν, όπως η ικανότητα διόρθωσης σφαλμάτων μετάδοσης (επίδοση). Για αυτό το λόγο έχουν αναπτυχθεί διάφοροι αλγόριθμοι αποκωδικοποίησης, οι οποίοι είναι επαναληπτικοί.

Στην παρούσα διπλωματική εργασία μελετήθηκε η υλοποίηση του LDPC αποκωδικοποιητή της matlab με χρήση ενός κβαντιστή που είναι βασισμένος στην μέθοδο του Information Bottleneck και συγκεκριμένα στον τροποποιημένο Sequential Information Bottleneck που ανέπτυξαν οι Jan Lewandowsky και Gerhard Bauch. Μελετήθηκε η απόδοση του αποκωδικοποιητή για πέντε διαφορετικούς πίνακες ελέγχου ισοτιμίας για 5 και 10 επαναλήψεις του αποκωδικοποιητή και ελέγχθηκε η απόδοσή του σε σύγκριση με την υλοποίηση με έναν γραμμικό κβαντιστή με ίδια αλλά και περισσότερα bit κβάντισης, σύμφωνα με τα πειραματικά δεδομένα της υλοποίησης. Τέλος, εξετάστηκε η υλοποίηση του κβαντιστή σε αρχιτεκτονικές υλικού.

EXTENSIVE ENGLISH SUMMARY

Hardware architectures for LDPC decoders with optimal information

STUDENT NAME, SURNAME: **SUPERVISOR NAME, SURNAME:**

PANTELEHMON FOTIADIS **VASSILIS PALIOURAS**

LDPC codes belong to the category of block codes. These are transmission error control codes and more specifically error correction codes. Although developed by Gallager in the early 60's, only in recent years have they been able to pique the interest of the scientific and research community due to the rapid development in the field of hardware architecture that allows them to be more easily implemented despite their large computer complexity. They are exchange rate codes with the main feature being the Low-Density Parity Check matrix from which they got their name. Since the encoding of these codes is relatively simple, their decoding is what largely determines the features of the code we are interested in, such as the ability to correct transmission errors (performance). For this reason, various decoding algorithms have been developed, which are iterative.

In the present dissertation the implementation of the matlab LDPC decoder was studied using a quantizer based on the Information Bottleneck method and in particular the modified Sequential Information Bottleneck developed by Jan Lewandowsky and Gerhard Bauch. The performance of the decoder was studied for five different parity check matrices for 5 and 10 repetitions of the decoder and its performance was tested in comparison with the implementation with a linear quantizer with the same and also more quantization bits, according to the experimental data of the implementation. Finally, the implementation of the quantizer in material architectures was examined.

Περιεχόμενα

1 Εισαγωγή 11

1.1 Σύστημα ψηφιακής επικοινωνίας 11

1.2 Λόγος σήματος προς θόρυβο 14

1.3 Κωδικοποίηση καναλιού 16

1.4 Κώδικες ανίχνευσης και διόρθωσης σφαλμάτων 17

1.5 Γραμμικοί block κώδικες 17

1.6 Απόσταση και βάρος Hamming 19

2 Κώδικες LDPC 20

2.1 Ιστορική Αναδρομή 20

2.2 Βασικά χαρακτηριστικά των LDPC κωδίκων 20

2.3 Διαγράμματα Tanner 22

2.4 Πλεονεκτήματα - Μειονεκτήματα 24

2.5 Αποκωδικοποίηση 25

2.6 Ο αλγόριθμος Belief Propagation(BP) 27

2.7 Ο αλγόριθμος Sum-Product 28

3 Information Bottleneck 33

3.1 Γενική περιγραφή της μεθόδου Information Bottleneck 33

3.2 Περιγραφή του sequential Information Bottleneck αλγορίθμου 35

3.3 Σχεδιασμός Quantizer 37

4 Υλοποίηση LDPC Decoder 40

4.1 Περιγραφή της υλοποίησης 40

4.2 Επιλογή Πινάκων Ισοτιμίας (Parity Check Matrices) 41

4.3 Περιγραφή κώδικα matlab 41

5 Αποτελέσματα υλοποίησης 44

5.1 Τρόπος υπολογισμού των αποτελεσμάτων 44
5.2 Παρουσίαση αποτελεσμάτων 44
5.2.1 Πίνακας ισοτιμίας H96.33.964 για 5 και 10 επαναλήψεις 45
5.2.2 Πίνακας ισοτιμίας H204.55.187 για 5 και 10 επαναλήψεις 47
5.2.3 Πίνακας ισοτιμίας H252.252.3.252 για 5 και 10 επαναλήψεις 50
5.2.4 Πίνακας ισοτιμίας H408.33.844 για 5 και 10 επαναλήψεις 53
5.2.5 Πίνακας ισοτιμίας H10000.10000.3.631 για 5 και 10 επαναλήψεις 56
5.3 Χρήση γραμμικού κβαντιστή με παραπάνω bit κβάντισης 58
5.4 Υλοποίηση του Κβαντιστή σε υλικό(hardware) 59
5.5 Συμπεράσματα – Παρατηρήσεις 60
6 Βιβλιογραφία 61

Πίνακας περιεχομένων εικόνων

Εικόνα 1: Τυπικό τηλεπικοινωνιακό σύστημα 12
Εικόνα 2: Απλοποιημένο ψηφιακό τηλεπικοινωνιακό σύστημα 13
Εικόνα 3: κώδικας matlab 42

Πίνακας περιεχομένων σχημάτων

Σχήμα 1: Γραφική αναπαράσταση τυχαίου ψηφιακού σήματος 14
Σχήμα 2: Γραφική αναπαράσταση επίδρασης θορύβου σε ψηφιακό σήμα. 15
Σχήμα 3: Παράδειγμα γραφικής απεικόνισης BER σε συνάρτηση με το SNR 15
Σχήμα 4: Παράδειγμα πίνακα ελέγχου ισοτιμίας με n=20, $d_u = 3$, $d_c = 4$ 22
Σχήμα 5: Παράδειγμα διαγράμματος Tanner για συγκεκριμένο πίνακα ελέγχου ισοτιμίας H 24
Σχήμα 6: Αποκωδικοποίηση με χρήση του Message-Passing αλγορίθμου δύο φάσεων. 26
Σχήμα 7: Αναπαράσταση της γενικής ιδέας της μεθόδου Information Bottleneck 33
Σχήμα 8: Είσοδοι και έξοδοι ενός Information Bottleneck αλγορίθμου 35

Σχήμα 9: Η διαδικασία της sequential Information Bottleneck μεθόδου 36

Σχήμα 10: Το quantization των εξόδων του καναλιού. Σε αυτό το παράδειγμα, οι έξοδοι είναι διακριτές σε |y|= 20 τιμές και έχουν ομαδοποιηθεί σε |T| = 4 ομάδες. 38

Σχήμα 11: Διαδικασία της τροποποιημένης sequential Information Bottleneck μεθόδου 39

Σχήμα 12: Όρια κβαντοποίησης της εξόδου του καναλιού με τον quantizer που χρησιμοποιεί την μέθοδο Information Bottleneck 39

Σχήμα 13: H96.33.964 επαναλήψεις: 5 45

Σχήμα 14: H96.33.964 επαναλήψεις: 10 46

Σχήμα 15: H204.55.187 επαναλήψεις: 5 47

Σχήμα 16: H204.55.187 επαναλήψεις: 10 49

Σχήμα 17: H252.252.3.252 επαναλήψεις: 5 50

Σχήμα 18: H252.252.3.252 επαναλήψεις: 10 51

Σχήμα 19: H408.33.844 επαναλήψεις: 5 53

Σχήμα 20: H408.33.844 επαναλήψεις: 10 54

Σχήμα 21: H10000.10000.3.631 επαναλήψεις: 5 56

Σχήμα 22: H10000.10000.3.631 επαναλήψεις: 10 57

Σχήμα 23: H96.33.964 10 επαναλήψεις, κβαντιστής με IB 4 bit σε σχέση με γραμμικό κβαντιστή 12 bit 59

Σχήμα 24: Αναπαράσταση του κβαντιστή σε υλικό 60

1 Εισαγωγή

1.1 Σύστημα ψηφιακής επικοινωνίας

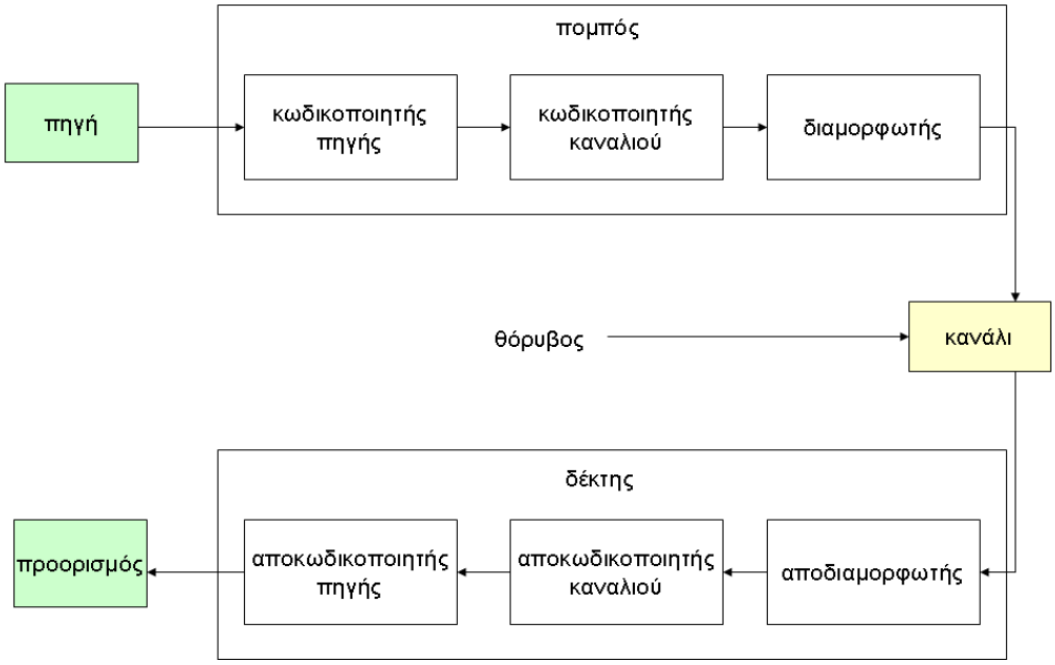
Τα τελευταία χρόνια, η δημιουργία όλο και μεγαλύτερων και ταχύτερων δικτύων επικοινωνίας για την ανταλλαγή, επεξεργασία και αποθήκευση ψηφιακών δεδομένων έχει δημιουργήσει την ανάγκη για πιο αποδοτικά και αξιόπιστα συστήματα επικοινωνιών και αποθήκευσης δεδομένων. Ειδικότερα η χρήση του διαδικτύου και των εφαρμογών του, που εξυπηρετούν εκατομμύρια χρήστες, έχει δημιουργήσει την ανάγκη για υψηλού επιπέδου μετάδοση. Ένα μεγάλο ζήτημα που αφορά στη σχεδίαση συστημάτων μετάδοσης είναι ο έλεγχος των λαθών, ώστε να εξασφαλίζεται κατά το δυνατόν αξιόπιστη επικοινωνία.

Τα συστήματα ηλεκτρικών επικοινωνιών έχουν σχεδιαστεί για να στέλνουν μηνύματα ή πληροφορία από μια πηγή που γεννά τα μηνύματα σε έναν ή περισσότερους προορισμούς . Η εικόνα (1) περιγράφει τις θεμελιώδεις βασικές δομές, που συγκροτούν ένα ψηφιακό τηλεπικοινωνιακό σύστημα. Η πληροφορία που γεννάται από την πηγή μπορεί να έχει μορφή φωνής (πηγή ομιλίας), εικόνας (πηγή εικόνας) η απλώς κειμένου σε κάποια συγκεκριμένη γλώσσα. Ένα χαρακτηριστικό οποιασδήποτε πηγής παραγωγής πληροφορίας είναι ότι η έξοδος της πηγής περιγράφεται με τη βοήθεια πιθανοτήτων, δηλαδή η έξοδος της πηγής δεν είναι νομοτελειακή, διαφορετικά δε θα χρειαζόταν η μετάδοση του μηνύματος.

Ένας ‘μετατροπέας’ είναι συνήθως αναγκαίος για να μετατρέπει την έξοδο της πηγής σε ηλεκτρικό σήμα κατάλληλο για μετάδοση. Για παράδειγμα, για πηγή ακουστικού σήματος χρησιμοποιείται ένα μικρόφωνο για τη μετατροπή σε ηλεκτρικό σήμα, ενώ για πηγή εικόνας χρησιμοποιείται μια βίντεο κάμερα. Στον προορισμό χρειάζεται μία αντίστοιχη αντίστροφη μετατροπή των ηλεκτρικών σημάτων σε κατάλληλη για χρήση μορφή, για παράδειγμα ήχο, εικόνα.

Το βασικό κομμάτι ενός συστήματος επικοινωνίας αποτελείται από τρία βασικά μέρη, τον πομπό, το κανάλι και τον δέκτη. Παρακάτω περιγράφονται οι λειτουργίες που επιτελούν τα σήματα αυτά.

Ένα τυπικό μοντέλο τηλεπικοινωνιακού συστήματος φαίνεται στο παρακάτω σχήμα:



Εικόνα 1: Τυπικό τηλεπικοινωνιακό σύστημα

Ο πομπός αναλαμβάνει να μετατρέψει την έξοδο της πηγής σε μία μορφή κατάλληλη για μετάδοση μέσα από το φυσικό κανάλι ή το οποιοδήποτε μέσο διάδοσης. Περιλαμβάνει τον κωδικοποιητή πηγής, τον κωδικοποιητή καναλιού και το διαμορφωτή ο ρόλος των οποίων είναι ο εξής:

- κωδικοποιητής πηγής: μετατροπή της αναλογικής πληροφορίας σε μία ακολουθία δυαδικών ψηφίων, ώστε να μπορέσει να μεταδοθεί μέσω του ψηφιακού συστήματος.

- κωδικοποιητής καναλιού: προσθήκη επιπλέον δυαδικών ψηφίων στην προς μετάδοση πληροφορία με σκοπό να καταστεί δυνατή η ανίχνευση και διόρθωση σφαλμάτων που πιθανώς θα προκύψουν κατά τη μετάδοση.

- διαμορφωτής: μετατροπή της ακολουθίας ψηφιακών δεδομένων που εξέρχεται από τον κωδικοποιητή καναλιού σε μία συνεχή κυματομορφή, με βάση τα χαρακτηριστικά του καναλιού, έτσι ώστε να μπορεί να μεταδοθεί μέσω του καναλιού.

Το κανάλι επικοινωνίας είναι το φυσικό μέσο που χρησιμοποιείται για τι μετάδοση της πληροφορίας από τον πομπό στο δέκτη. Ανάλογα με τον τρόπο μετάδοσης, ασύρματη ή ενσύρματη, το κανάλι μπορεί να είναι η ατμόσφαιρα, ενσύρματες γραμμές, καλώδια οπτικών ινών, κλπ. Κάθε κανάλι ανάλογα με το θόρυβο που εισάγει μπορεί αν μοντελοποιηθεί. Από τα πιο διαδεδομένα μοντέλα είναι αυτό του καναλιού δίχως μνήμη και του λευκού γκαουσιανού προσθετικού θορύβου (Additive White Gaussian Noise-AWGN).

Τέλος, ο δέκτης αποτελεί το τρίτο βασικό τμήμα του συστήματος επικοινωνίας και αποτελείται και αυτός από τρία βασικά κομμάτια ανάλογα με αυτά του πομπού, τον

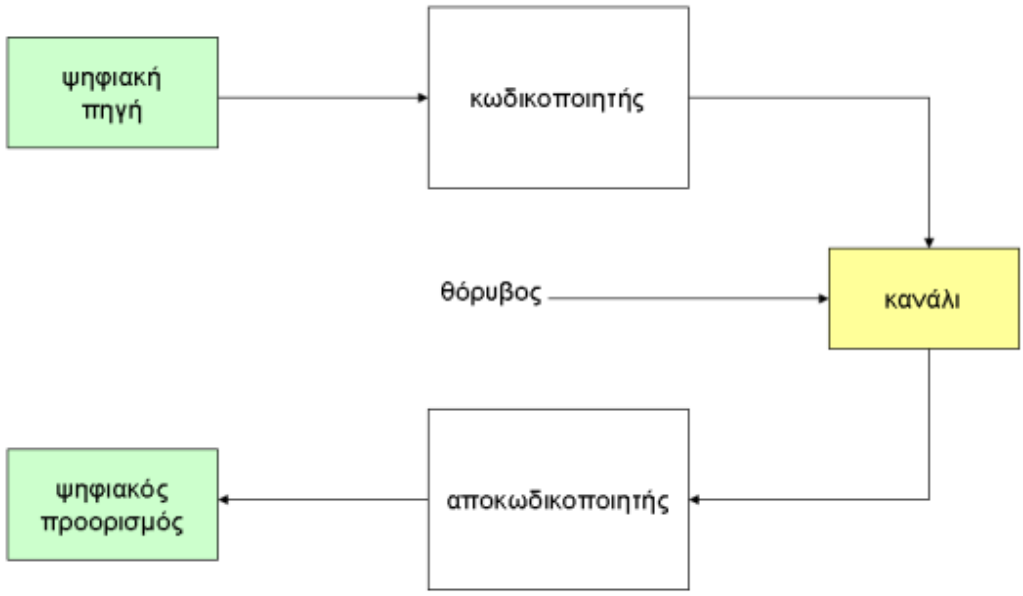
αποδιαμορφωτή, τον αποκωδικοποιητή καναλιού και τον αποκωδικοποιητή πηγής, οι ρόλοι των οποίων είναι συνοπτικά:

• αποδιαμορφωτής: μετατροπή της ληφθείσας κυματομορφής από το κανάλι σε μία ακολουθία εξόδου, που αποκαλείται ληφθείσα ακολουθία, και ενδέχεται να είναι διακριτή ή συνεχής.

• αποκωδικοποιητής καναλιού: αυτός αναλαμβάνει αφενός να αφαιρέσει την πλεονάζουσα πληροφορία που εισήγαγε ο κωδικοποιητής καναλιού και αφετέρου να μετατρέψει τη ληφθείσα πληροφορία σε δυαδική, αξιοποιώντας την πλεονάζουσα πληροφορία. Στην ιδανική περίπτωση η αποκωδικοποιημένη δυαδική ακολουθία θα είναι πανομοιότυπη με την αντίστοιχη που παρήγαγε ο κωδικοποιητής πηγής του πομπού.

• αποκωδικοποιητής πηγής: ανακατασκευή του αρχικού αναλογικού σήματος της πηγής, όσο το δυνατόν πιο πιστά, με βάση την έξοδο που λαμβάνει από τον αποκωδικοποιητή καναλιού. Τα τελικό αναλογικό σήμα πιθανότατα θα αποτελεί μία προσέγγιση του αρχικού σήματος αφού στην πληροφορία έχουν υπεισέλθει σφάλματα που δημιουργήθηκαν κατά τη μετάδοση.

Η δομή ενός απλοποιημένου ψηφιακού τηλεπικοινωνιακού συστήματος φαίνεται στο ακόλουθο σχήμα, όπου ο κωδικοποιητής πηγής έχει ενσωματωθεί στην πηγή σχηματίζοντας την ψηφιακή πηγή, ο αποκωδικοποιητής πηγής στον προορισμό σχηματίζοντας τον ψηφιακό προορισμό, ενώ ο διαμορφωτής και ο αποδιαμορφωτής έχουν ενσωματωθεί στο κανάλι.



Εικόνα 2: Απλοποιημένο ψηφιακό τηλεπικοινωνιακό σύστημα

1.2 Λόγος σήματος προς θόρυβο

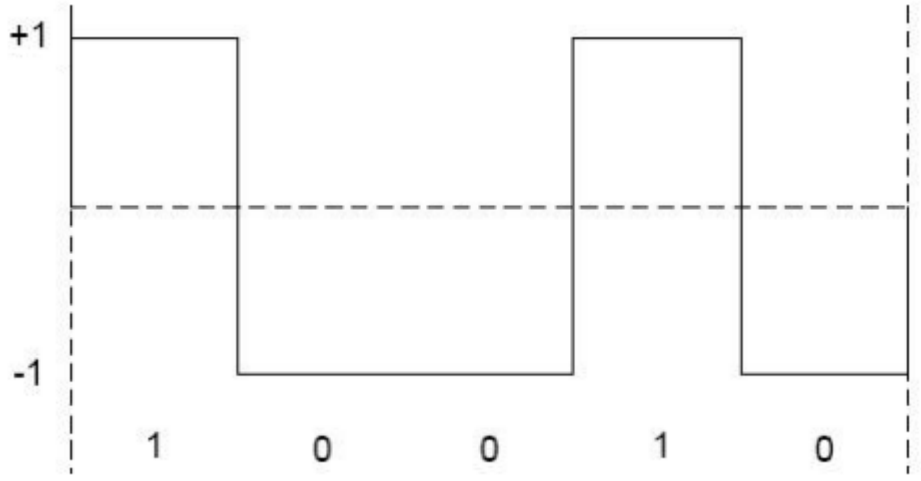
Δυστυχώς, όπως είναι γνωστό η επίδραση του θορύβου του καναλιού στο εκπεμπόμενο σήμα είναι αναπόφευκτη. Ο λόγος σήματος προς θόρυβο είναι μια χαρακτηριστική έννοια κάθε τηλεπικοινωνιακού συστήματος αλλά και γενικά της ηλεκτρικής εφαρμοσμένης μηχανικής. Ορίζεται ως η αναλογία της ισχύς του σήματος προς την ισχύ θορύβου που αλλοιώνει το σήμα (1). Με λιγότερο τεχνικούς όρους, η αναλογία σήματος προς θόρυβο συγκρίνει το επίπεδο ενός επιθυμητού σήματος (όπως η μουσική, μια τηλεφωνική συνδιάλεξη κ.α.) με το επίπεδο παρασιτικού θορύβου. Όσο υψηλότερη η αναλογία, λιγότερο ο αδιάκριτος ο παρασιτικός θόρυβος είναι.

$$\mathrm{SNR} = \frac{\mathrm{P_{signal}}}{\mathrm{P_{noise}}} = \left(\frac{\mathrm{A_{signal}}}{\mathrm{A_{noise}}}\right)^2 \qquad (1)$$

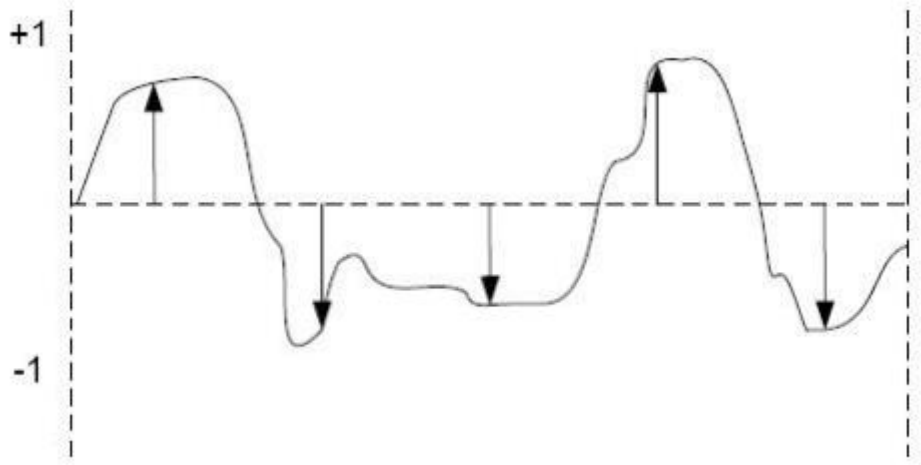
Επειδή πολλά σήματα έχουν μια πολύ ευρεία δυναμική περιοχή, το SNR εκφράζεται συνήθως με τη μορφή της λογαριθμικής decibel κλίμακας. Decibels, το SNR είναι, εξ ορισμού, 10 φορές ο λογάριθμος της αναλογίας ισχύος σήματος ανά ισχύ θορύβου. Εάν το σήμα και ο θόρυβος μετριούνται υπό την ίδια σύνθετη αντίσταση, το SNR μπορεί να ληφθεί με να υπολογίσει 20 φορές το λογάριθμο με βάση-10 της αναλογίας εύρους:

$$\mathrm{SNR}(dB) = 10\log_{10}\left(\frac{\mathrm{P_{signal}}}{\mathrm{P_{noise}}}\right) = 20\log_{10}\left(\frac{\mathrm{A_{signal}}}{\mathrm{A_{noise}}}\right)^2 \qquad (2)$$

Η ύπαρξη του παρασιτικού θορύβου του καναλιού, καθώς και η απόκριση του καναλιού μετάδοσης οδηγούν σε εσφαλμένα τιμές στα σήματα κατά τη λήψη τους στους δέκτες.

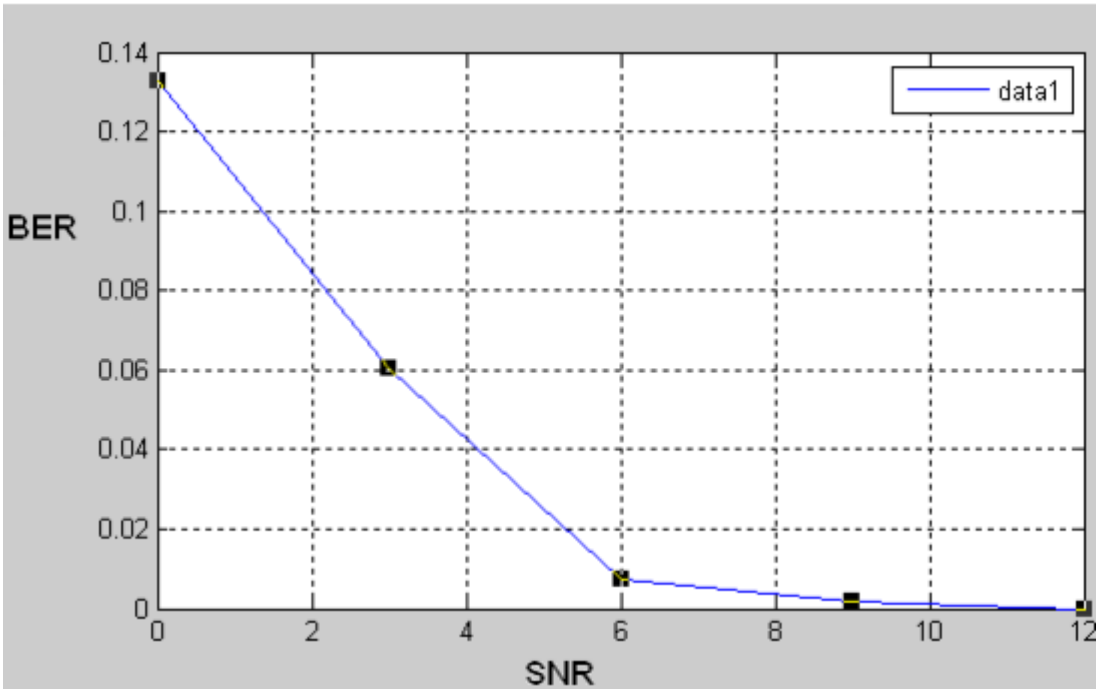


Σχήμα 1: Γραφική αναπαράσταση τυχαίου ψηφιακού σήματος



Σχήμα 2: Γραφική αναπαράσταση επίδρασης θορύβου σε ψηφιακό σήμα.

Μία μέθοδος αναπαράστασης του πλήθους των σφαλμάτων στο σύνολο των δεδομένων είναι ο υπολογισμός του ποσοστού Bit Error Rate (BER) σε συνάρτηση με το SNR.(Γραφική 1.1).



Σχήμα 3: Παράδειγμα γραφικής απεικόνισης BER σε συνάρτηση με το SNR

Παρατηρούμε ότι όσο αυξάνουμε την ένταση του σήματος σε σχέση με το θόρυβο τόσο μειώνεται η πιθανότητα λάθους. Σε εφαρμογές που η αξιοπιστία και η ποιότητα δεν είναι πρωτεύοντα ζητούμενα, όπως στην περίπτωση μιας τηλεφωνικής συνδιάλεξης δεν επιδιώκεται μεγιστοποίηση στο βαθμό που απαιτείται για άλλες εφαρμογές, όπως στρατιωτικές, ή μεταδόσεις δεδομένων. Όσο χαμηλότερα ή πιο αριστερά βρίσκεται μια καμπύλη στην παραπάνω γραφική τόσο πιο αποδοτική είναι η διόρθωση των λαθών.

1.3 Κωδικοποίηση καναλιού

Ένα από τα πιο βασικά κομμάτια ενός συστήματος ψηφιακής μετάδοσης είναι η κωδικοποίηση καναλιού. Αυτή υλοποιείται με τη χρήση κωδίκων, που ονομάζονται κώδικες ελέγχου λαθών (ή κώδικες καναλιού). Οι κώδικες αυτοί εισάγουν κάποια πλεονάζουσα πληροφορία στην προς μετάδοση πληροφορία, πριν αυτή μεταδοθεί. Το επιπλέον αυτό τμήμα μπορεί να το χρησιμοποιήσει στη συνέχεια ο δέκτης ώστε να ανιχνεύσει και ενδεχομένως και να διορθώσει τα λάθη που θα δημιουργηθούν κατά τη μετάδοση.

Η δημιουργία τέτοιων κωδίκων είναι εφικτή σύμφωνα με το θεώρημα κωδικοποίησης διακριτού καναλιού με θόρυβο του Shannon. Ο Shannon απέδειξε πως η πιθανότητα λανθασμένης μετάδοσης δεδομένων μέσω ενός τηλεπικοινωνιακού καναλιού με θόρυβο μπορεί να περιοριστεί οσοδήποτε επιθυμούμε, αρκεί ο ρυθμός μετάδοσης των δεδομένων πληροφορίας (R) να μην ξεπερνάει ένα συγκεκριμένο όριο. Το όριο καλείται όριο Shannon ή χωρητικότητα του καναλιού και αποτελεί χαρακτηριστικό του καναλιού μετάδοσης. Έτσι χρησιμοποιώντας την κατάλληλη κωδικοποίηση πληροφορίας, ο ρυθμός εμφάνισης λαθών μπορεί να γίνει αυθαίρετα μικρός χωρίς όμως να μειωθεί ο ρυθμός μετάδοσης δεδομένων. Όσο μικρότερη όμως είναι η επιθυμητή πιθανότητα σφάλματος, τόσο πολυπλοκότερη θα πρέπει να είναι και η κωδικοποίηση.

Ο λόγος λοιπόν για τον οποίο είναι σημαντική η αναφορά στην κωδικοποίηση του καναλιού και στο μαθηματικό μοντέλο είναι γιατί με τον τρόπο αυτό είναι δυνατή η ελαχιστοποίηση της λανθασμένης πληροφορίας που υπήρξε λόγω της μετάδοσης. Με βάση το θεώρημα του Shannon είναι εφικτή η μείωση της πιθανότητας του λάθους κατά τη μετάδοση αρκεί ο ρυθμός της μεταδιδόμενης πληροφορίας να μην υπερβαίνει τη χωρητικότητα C του καναλιού. Η χωρητικότητα C δίνεται από τη σχέση:

$$\mathrm{C} = \mathrm{B} \cdot \log_2(1 + \mathrm{SNR}) \qquad (3)$$

όπου το B συμβολίζει το εύρος ζώνης του καναλιού και το SNR το λόγο της ισχύος του σήματος προς μετάδοση προς την ισχύ του σήματος θορύβου. Για να υπάρχει η δυνατότητα ελαχιστοποίησης των λαθών στο μεταδιδόμενο σήμα με βάση το θεώρημα Shannon θα πρέπει να ισχύει:

$$R < C$$

Με βάση την (3) η χωρητικότητα του καναλιού εξαρτάται από το εύρος ζώνης και την ισχύ του θορύβου. Έτσι για να περιορίσουμε την πιθανότητα σφάλματος θα πρέπει η μετάδοση να γίνεται με ρυθμό που να προσεγγίζει όσο το δυνατό περισσότερο το όριο του Shannon. Για να είναι αυτό εφικτό, χωρίς όμως να αλλάξει ο ρυθμός μετάδοσης ή ο θόρυβος στο κανάλι, θα πρέπει να αυξηθεί η υπολογιστική πολυπλοκότητα της κωδικοποίησης του καναλιού, κάτι που θα επιφέρει αύξηση του κόστους μετάδοσης. Η υπολογιστική πολυπλοκότητα και η ισχύς του μεταδιδόμενου σήματος είναι παράγοντες που όσο αυξάνονται βελτιώνουν την αξιοπιστία του συστήματος αλλά ταυτόχρονα αυξάνουν και το κόστος μετάδοσης. Το ζητούμενο σε κάθε περίπτωση είναι να επιτευχθεί αποδοτικότερη κωδικοποίηση με το μικρότερο υπολογιστικό κόστος.

1.4 Κώδικες ανίχνευσης και διόρθωσης σφαλμάτων

Ο κώδικας που χρησιμοποιείται και οι διαδικασίες που ακολουθούνται κατά τις φάσεις της κωδικοποίησης και της αποκωδικοποίησης καθορίζουν σε μεγάλο βαθμό την απόδοση του συστήματος. Οι κώδικες διόρθωσης λαθών χωρίζονται σε δύο μεγάλες κατηγορίες, στους μπλοκ κώδικες (Block Codes) και στους συνελικτικούς κώδικες (Convolutional Codes).

Κύριο χαρακτηριστικό των μπλοκ κωδικών είναι ο τεμαχισμός της προς μετάδοση πληροφορίας σε μπλοκ των k συμβόλων και η αντιστοιχία καθενός από αυτά σε ένα μπλοκ n συμβόλων, το οποίο αποκαλείται κωδική λέξη ή codeword (n≥k). Στους συνελικτικούς κώδικες, κάθε m-bit σύμβολο πληροφορίας, μετατρέπεται σε ένα n-bit σύμβολο (n≥m) , με κάθε σύμβολο να εξαρτάται όχι μόνο από το ίδιο, αλλά και από τα k προηγούμενα από αυτό σύμβολα πληροφορίας. Επομένως, η κύρια διαφορά μεταξύ των δύο κατηγοριών κωδικών είναι η ύπαρξη μνήμης στους συνελικτικούς κώδικες.

Μια ακόμη σημαντική διάκριση που μπορεί να γίνει είναι βάσει του τρόπου διαχείρισης των σφαλμάτων. Υπάρχουν κώδικες οι οποίοι απλά ανιχνεύουν την ύπαρξη σφαλμάτων και κώδικες που πέρα από την ανίχνευση έχουν και την ικανότητα διόρθωσης των σφαλμάτων. Το επιπλέον κόστος, βέβαια, των τελευταίων είναι στην υπολογιστική πολυπλοκότητα, η οποία είναι σαφώς μεγαλύτερη.

Στην κατηγορία των κωδικών μπλοκ ανήκουν οι κώδικες Hamming, οι κυκλικοί κώδικες, οι BCH, οι Reed-Solomon και αρκετοί άλλοι. Συνδυασμός απλών κωδικών έχει οδηγήσει στην δημιουργία εξαιρετικά επιτυχημένων σύνθετων σχημάτων κωδικοποίησης καναλιού. Οι κώδικες γινομένου, οι αλυσιδωτοί κώδικες και οι τούρμπο κώδικες (turbo codes) είναι ορισμένες χαρακτηριστικές περιπτώσεις. Οι τελευταίοι αποτέλεσαν τα τελευταία χρόνια τους πλέον αποδοτικούς κώδικες για κωδικοποίηση καναλιού, επιτυγχάνοντας σε αρκετές περιπτώσεις ρυθμούς μετάδοσης πολύ κοντά στο όριο του Shannon. Ωστόσο, στην παρούσα διπλωματική εργασία θα ασχοληθούμε με μια άλλη κατηγορία κωδίκων, τους LDPC κώδικες (Low-Density-Parity-Check codes). Πρόκειται για γραμμικούς μπλοκ κώδικες και θα αναφερθούμε πιο διεξοδικά σε αυτούς στο επόμενο κεφάλαιο.

1.5 Γραμμικοί block κώδικες

Η κωδικοποίηση με block είναι ένας τύπος κωδικοποίησης διόρθωσης σφαλμάτων στην οποία τα δεδομένα που επρόκειτο να μεταδοθούν χωρίζονται σε μηνύματα σταθερού μεγέθους. Πριν από τη μετάδοση, κάθε μήνυμα κωδικοποιείται σε μια κωδική λέξη (codeword)(αναφέρεται επίσης ως "block"), από έναν κωδικοποιητή. Τα δεδομένα ισοτιμίας(parity data), εισάγονται κατά τη διαδικασία κωδικοποίησης έτσι ώστε τα codewords να γίνουν πολύ μεγαλύτερα από τα μηνύματα. Κάθε codeword περιέχει και δυαδικά ψηφία του μηνύματος και δυαδικά ψηφία ισοτιμίας (parity bits). Υποθέτοντας ότι κάθε codeword αποτελείται από ν δυαδικά ψηφία, μόνο συγκεκριμένα μοτίβα από ν δυαδικά ψηφία είναι έγκυρα codewords ενώ τα υπόλοιπα είναι άκυρα. Στην συνέχεια τα codewords μεταδίδονται, το οποίο μπορεί να προκαλέσει αλλοίωσή

τους. Κατά την λήψη τους, ένας αποκωδικοποιητής επιχειρεί να εξάγει τα πρωτότυπα μηνύματα από τα ληφθέντα και ενδεχομένως αλλοιωμένα codewords.

Ένας δυαδικός κώδικας block δημιουργεί ένα block n κωδικοποιημένων δυαδικών ψηφίων από k δυαδικά ψηφία πληροφορίας. Αυτό το ονομάζουμε δυαδικό block κωδικό. Τα κωδικοποιημένα δυαδικά ψηφία ονομάζονται επίσης σύμβολα codeword. Ο ρυθμός του κώδικα είναι $R_c=k/n$ δυαδικά ψηφία πληροφορίας ανά σύμβολο codeword. Αν υποθέσουμε ότι τα σύμβολα codeword μεταδίδονται σε όλο το κανάλι με ρυθμό R_s σύμβολα/δευτερόλεπτο, τότε ο ρυθμός της πληροφορίας που σχετίζεται με ένα (n,k) block κωδικό είναι $R_b=R_cR_s=(k/n)R_s$ (δυαδικά ψηφία)/δευτερόλεπτο. Έτσι βλέπουμε ότι η κωδικοποίηση block μειώνει το ρυθμό δεδομένων σε σύγκριση με αυτό που λαμβάνουμε με μη κωδικοποιημένη διαμόρφωση κατά το ρυθμό κώδικα R_c. Ένας block κώδικας είναι γραμμικός αν το άθροισμα οποιονδήποτε έγκυρων κωδικών λέξεων είναι επίσης κωδική λέξη. Στην περίπτωση δυαδικού κώδικα αυτό σημαίνει πως το αποτέλεσμα της συνιστώσας-προς-συνιστώσα modulo-2 λογικής πράξης (ή ισοδύναμα XOR λογική) μεταξύ δύο κωδικών λέξεων, είναι επίσης κωδική λέξη. Στους γραμμικούς κώδικες ανήκει η μηδενική λέξη, καθώς το άθροισμα μιας οποιασδήποτε έγκυρης κωδικής λέξης με τον εαυτό της, μας δίνει τη μηδενική λέξη και σύμφωνα με τον ορισμό των γραμμικών κωδικών, θα πρέπει και αυτή να είναι έγκυρη κωδική λέξη.

Η μετατροπή της ακολουθίας των k bits (λέξη πληροφορίας) σε ακολουθία των n bits πραγματοποιείται με την βοήθεια ενός k × n δυαδικού πίνακα G, ο οποίος ονομάζεται γεννήτορας πίνακας του κώδικα. Η κωδική λέξη παράγεται με τον πολλαπλασιασμό της λέξης πληροφορίας u_i (ακολουθία k bits) με τον γεννήτορα πίνακα

$$c_i = u_i \cdot G \qquad (4)$$

Προκύπτει έτσι η κωδική λέξη c_i. Τα n δυαδικά ψηφία που συνιστούν την κωδική λέξη ορίζουν έναν χώρο n διαστάσεων. Κάθε δυαδικό σύμβολο της κωδικής λέξης είναι και μία συνιστώσα του χώρου αυτού. Ο n-διάστατος χώρος περιλαμβάνει 2^n στοιχεία. Από αυτά μόνο τα 2^k αποτελούν έγκυρες κωδικές λέξεις του κώδικα. Οι 2^k έγκυρες κωδικές λέξεις συνιστούν έναν k-διάστατο υποχώρο του n-διάστατου χώρου. Ονομάζουμε C αυτό τον υποχώρο. Οι γραμμές του γεννήτορα πίνακα G δεν είναι τίποτα περισσότερο από τα k διανύσματα που αποτελούν τη βάση του υποχώρου C.

Έστω όλες οι δυαδικές ακολουθίες μήκους n οι οποίες είναι ορθογώνιες προ όλα τα διανύσματα του k-διάστατου υποχώρου C. Κάθε μία από αυτές τις ακολουθίες έχει την ιδιότητα :

$$H \oplus c_i^T = 0 \qquad (5)$$

Όπου $i=1,2,\ldots,2^k$, h είναι μία από τις ορθογώνιες ακολουθίες μήκους n, c_i^T είναι η ανάστροφη εκδοχή της έγκυρης κωδική λέξης c_i, ενώ ο τελεστής $\oplus$ δηλώνει την συνιστώσα προς συνιστώσα XOR λογική πράξη ή ισοδύναμα modulo-2 άθροιση. Αποδεικνύεται ότι το πλήθος των μήκους n ακολουθιών που έχουν την παραπάνω ιδιότητα είναι 2^{n-k}. Αυτές ορίζουν επομένως έναν

(n-k)-διάστατο υποχώρο του n-διάστατου χώρου. Ο υποχώρος αυτός είναι γραμμικός και καλείται ορθογώνιο συμπλήρωμα του υποχώρου.

Οι 2^{n-k} ορθογώνιες ακολουθίες h_i μπορεί να θεωρηθεί ότι είναι οι έγκυρες κωδικές λέξεις ενός (n,n-k) γραμμικού block κώδικα, ο οποίος συμβολίζεται με C^T και είναι σύμφωνα με τη σχέση (5) ορθογώνιες ως προς τις κωδικές λέξεις c_i του κώδικα C. Έστω H ο γεννήτορας πίνακας του κώδικα C^T. Ο πίνακας H αποτελείται από (n-k) γραμμές n στοιχείων οι οποίες είναι τα διανύσματα βάση του (n-k)-διάστατου χώρου. Κάθε μία εξ' αυτών h_i i=1,2,…,n-k είναι μία έγκυρη κωδική λέξη του κώδικα C^T. Επομένως κάθε γραμμή του πίνακα H είναι ορθογώνια ως προς εκάστη έγκυρη κωδική λέξη του κώδικα C, επαληθεύοντας την σχέση (5). Κατά συνέπεια ο πίνακας H μας παρέχει n-k σχέσεις, τις οποίες θα πρέπει να επαληθεύει μία κωδική λέξη για να είναι έγκυρη και οι οποίες συνοψίζονται στην παρακάτω σχέση,

$$c_i \oplus H^T = 0 \qquad (6)$$

Πρακτικά, στην περίπτωση του δυαδικού κώδικα, λόγω της modulo-2 άθροισης κάθε μία γραμμή ελέγχει αν μεταξύ συγκεκριμένων ψηφίων της κωδικής λέξης υπάρχει άρτιο πλήθος άσσων. Επομένως ο πίνακας H ελέγχει την ονομαζόμενη ισοτιμία των άσσων της κωδικής λέξης. Για τον λόγο αυτό ονομάζεται Πίνακας Ελέγχου Ισοτιμίας (Parity Check Matrix) του αρχικού κώδικα C και είναι, όπως και ο γεννήτορας πίνακας G, χαρακτηριστικός του κώδικα.

1.6 Απόσταση και βάρος Hamming

Για οποιαδήποτε κωδική λέξη ορίζεται ένας συγκεκριμένος αριθμός ο οποίος καλείται βάρος Hamming (Hamming weight) και αντιστοιχεί στον αριθμό των μη μηδενικών στοιχείων της κωδικής λέξης. Η απόσταση Hamming (Hamming distance) d μεταξύ δύο κωδικών λέξεων ορίζεται ως ο αριθμός των θέσεων στις οποίες διαφέρουν οι δύο λέξεις. Ως ελάχιστη απόσταση (minimum distance) d_{min} ενός μπλοκ κώδικα 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 block κώδικα με ελάχιστη απόσταση d_{min} αποδεικνύεται ότι σε κάθε κωδική λέξη υπάρχει δυνατότητα να ανιχνευθούν $s \leq (d_{min}-1)$ σφάλματα και να διορθωθούν αντίστοιχα $t \leq (d_{min}-1)/2$. Σημαντική είναι επίσης η σχέση που υπάρχει μεταξύ του πίνακα ελέγχου ισοτιμίας και της ελάχιστης απόστασης, η οποία μπορεί να οριστεί ως ο ελάχιστος αριθμός στηλών του πίνακα H που έχουν άθροισμα ίσο με το 0.

2 Κώδικες LDPC

2.1 Ιστορική Αναδρομή

Οι κώδικες LDPC εφευρέθηκαν από τον R.G.Gallager στις αρχές της δεκαετίας του 60 και ήταν οι πρώτοι κώδικες διόρθωσης λαθών, οι οποίοι μπορούσαν να προσεγγίσουν ρυθμούς μετάδοσης δεδομένων πολύ κοντά στο θεωρητικό μέγιστο, το όριο Shannon. Λόγω, όμως, του γεγονότος ότι τότε δεν ήταν εφικτό να υλοποιηθούν σε υλικό, εξαιτίας των μεγάλων απαιτήσεων τους σε υπολογιστική πολυπλοκότητα, έμειναν στο περιθώριο για αρκετά χρόνια. Με την αλματώδη εξέλιξη της τεχνολογίας, από τα μέσα της δεκαετίας του 90 το ενδιαφέρον στράφηκε ξανά στους κώδικες LDPC, αφού πλέον ήταν δυνατή η υλοποίηση τους. Επιπλέον, η μεγάλη ανάπτυξη της τεχνολογίας της πληροφορίας έστρεψε το ενδιαφέρον της αγοράς σε υψηλής απόδοσης κώδικες μετάδοσης δεδομένων, οι οποίοι παίζουν καθοριστικό ρόλο σε πλήθος παραγόντων που αφορούν τη μετάδοση. Οι παράγοντες αυτοί κυμαίνονται από την ποιότητα του σήματος έως και τη διάρκεια ζωής της μπαταρίας.

Το 1993 παρουσιάστηκαν οι turbo κώδικες (Berrou, Glavieux, and Thitimajshima, 1993) [4], και ο αντίστοιχος επαναληπτικός αλγόριθμος αποκωδικοποίησης. Οι αξιοσημείωτες επιδόσεις των κωδικών αυτών προκάλεσε ποικίλα ερωτήματα και μεγάλο ενδιαφέρον για παρόμοιες επαναληπτικές μεθόδους. Το 1995, οι D.MacKay και R. M. Neal [5] επανέφεραν στο προσκήνιο τους LDPC κώδικες, και συνέδεσαν τον επαναληπτικό τους αλγόριθμο με την έννοια του belief propagation (Pearl, 1988)[6], την οποία δανείστηκαν από την κοινότητα της τεχνητής νοημοσύνης (δίκτυα Bayes). Το 1996, οι M. Sipser and D. A. Spielman [7] χρησιμοποίησαν τον πρώτο αλγόριθμο αποκωδικοποίησης του R.G.Gallager (αλγόριθμος Α) στην αποκωδικοποίηση κωδικών διαστελλόμενων στοιχείων (expander codes). Το 1998, τα διμερή γραφήματα (διαγράμματα Tanner) [8] ξεκίνησαν να χρησιμοποιούνται για την αναπαράσταση κωδικών διόρθωσης σφαλμάτων (Kschischang and Frey, 1998) [9], με σκοπό να μπορούν να περιγραφούν πολλοί διαφορετικοί αλγόριθμοι μέσω κοινού φορμαλισμού. Μπορούμε να πούμε, πως οι κώδικες LDPC είχαν καθοριστική συμβολή σε δύο επαναστατικές εξελίξεις στον τομέα της κωδικοποίησης καναλιού: τη βασισμένη σε γράφημα περιγραφή του κώδικα και τις επαναληπτικές μεθόδους αποκωδικοποίησης.

2.2 Βασικά χαρακτηριστικά των LDPC κωδίκων

Πρόκειται για κώδικες ελέγχου ισοτιμίας. Περιγράφονται πλήρως, είτε από τον γεννήτορα πίνακα G, είτε από τον πίνακα ελέγχου ισοτιμίας (Parity Check Matrix) H, είτε από ένα διάγραμμα Tanner.

Όπως αναφέραμε στην προηγούμενη ενότητα, η κωδικοποίηση ενός μπλοκ κώδικα γίνεται με την βοήθεια του γεννήτορα πίνακα G. Τα δεδομένα προς μετάδοση, τεμαχίζονται σε μπλοκ μήκους k. Αν και οι κώδικες LDPC μπορούν να γενικευτούν και για μη δυαδικά αλφάβητα, η μελέτη τέτοιων περιπτώσεων ξεφεύγει από τους σκοπούς της παρούσας εργασίας. Από εδώ και στο εξής, κάθε φορά που θα αναφερόμαστε σε κώδικα LDPC θα εννοούμε δυαδικό κώδικα. Η λέξη πληροφορίας πολλαπλασιάζεται με τον k × n πίνακα G, σύμφωνα με την σχέση (4), και παράγει ένα διάνυσμα 1 × n. Αυτό το διάνυσμα είναι το κωδικοποιημένο μπλοκ και ονομάζεται κωδική λέξη. Δεδομένου ότι γνωρίζοντας τον γεννήτορα πίνακα μπορούμε να βρούμε θεωρητικά όλες τις κωδικές λέξεις του κώδικα, ένας πρώτος τρόπος περιγραφής ενός κώδικα LDPC είναι ο πίνακας αυτός.

Αντίστοιχα, μπορεί να περιγραφεί πλήρως ο κώδικας LDPC με τον πίνακα ελέγχου ισοτιμίας H. Αυτός έχει το χαρακτηριστικό ότι πολύ μικρό ποσοστό των στοιχείων του είναι μη μηδενικά (δηλαδή είναι άσοι). Πρόκειται επομένως για έναν «αραιό» (sparse) πίνακα, όσον αφορά την παρουσία άσων, χαρακτηριστικό από το οποίο πήραν και το όνομα τους οι συγκεκριμένοι κώδικες. Δεδομένου ότι κάθε έγκυρη λέξη του κώδικα θα πρέπει να ικανοποιεί την σχέση (6), γνωρίζοντας τον πίνακα H μπορούμε να διακρίνουμε από το σύνολο των 2n δυνατών συνδυασμών κωδικών λέξεων, τις 2k έγκυρες. Κάθε στήλη και κάθε γραμμή του πίνακα ισοτιμίας αποτελείται από έναν μικρό σταθερό αριθμό d_u και d_c , αντίστοιχα, άσσων. Έχει καθιερωθεί να συμβολίζεται ως (n, d_u, d_c) ένας κώδικας LDPC του οποίου ο πίνακας ελέγχου ισοτιμίας αποτελείται από n στήλες και m = n−k γραμμές, κάθε στήλη του οποίου περιέχει du άσσους και κάθε γραμμή του d_c άσους. Όπως γίνεται αντιληπτό, από την παραπάνω περιγραφή δεν συμβολίζεται ένας μόνος κώδικας αλλά μια ομάδα κωδίκων LDPC, αφού η αλλαγή θέσης σε ένα και μόνο άσο του πίνακα, μας δίνει ένα διαφορετικό κώδικα. Ένα παράδειγμα πίνακα ελέγχου ισοτιμίας δίνεται στο ακόλουθο σχήμα.

Κάθε γραμμή του πίνακα H αντιστοιχεί σε μία εξίσωση ελέγχου ισοτιμίας (parity check) και κάθε άσος στην θέση (i, j) του πίνακα H, σημαίνει πως το j-οστό σύμβολο δεδομένων συμμετέχει στην i-οστη εξίσωση ελέγχου ισοτιμίας. Γίνεται αντιληπτό πως αφού κάθε γραμμή του πίνακα ελέγχου ισοτιμίας ικανοποιεί την σχέση (5), θα πρέπει πρακτικά μεταξύ εκείνων των δυαδικών ψηφίων της κωδικής λέξης που αντιστοιχούν στους άσσους της γραμμής ελέγχου, να υπάρχει άρτιο πλήθος άσων. Επιπρόσθετα, δεδομένου ότι ο συνολικός αριθμός των άσων ενός πίνακα ελέγχου ισοτιμίας διαστάσεων (m, n) είναι σταθερός ισχύει:

$$m \times d_c = n \times d_u \qquad (7)$$

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}$$

Σχήμα 4: Παράδειγμα πίνακα ελέγχου ισοτιμίας με n=20, $d_u = 3$, $d_c = 4$

΄Ενα μέγεθος που έχει ενδιαφέρον είναι ο ρυθμός (rate) R του κώδικα. Ορίζεται ως ο λόγος των δεδομένων πληροφορίας, ή information bits (k), προς τα μεταδιδόμενα δεδομένα (n). Επομένως, σύμφωνα με τα όσα έχουμε αναφέρει έως τώρα, ο ρυθμός ενός (n, d_u, d_c) κώδικα LDPC, του οποίου η λέξη πληροφορίας αποτελείται από k δυαδικά ψηφία, δίνεται από την σχέση:

$$R = \frac{k}{n} = \frac{n-m}{n} = 1 - \frac{m}{n} \Rightarrow R = 1 - \frac{d_u}{d_c} \qquad (8)$$

Ένας κώδικας LDPC του οποίου οι παράμετροι d_u, d_c είναι σταθερές για όλες τις στήλες και όλες τις γραμμές του πίνακα ισοτιμίας, όπως ο κώδικας του σχήματος (4), ονομάζεται κανονικός (regular). Υπάρχουν, ωστόσο, και κώδικες για τους οποίους δεν ισχύει κάτι τέτοιο. Αυτοί ανήκουν στην κατηγορία των μη-κανονικών (irregular) κωδίκων LDPC. Σε αυτήν την περίπτωση, ο αριθμός των άσσων κάθε γραμμής ή κάθε στήλης, είναι συνάρτηση της γραμμής ή της στήλης, αντίστοιχα, και περιγράφεται συνήθως με συγκεκριμένα πολυώνυμα. Εμείς θα ασχοληθούμε με τους κανονικούς (regular).

2.3 Διαγράμματα Tanner

Το 1981 ο Tanner (8) προσδιόρισε ένα πολύ πρακτικό τρόπο περιγραφής ενός μπλοκ κώδικα, ο οποίος αποδείχτηκε εξαιρετικά χρήσιμος για τους κώδικες LDP C. Η περιγραφή του Tanner βασίζεται στον πίνακα ελέγχου ισοτιμίας και αποτελείται από έναν διμερή γράφο (bipartite graph) ή, όπως έχει επικρατήσει να λέγεται, «διάγραμμα Tanner». Ο γράφος

αποτελείται από δύο ειδών κορυφές - κόμβους και ακμές μεταξύ αυτών. Οι κορυφές της μιας κατηγορίας ονομάζονται κόμβοι μεταβλητής (variable nodes), και αντιστοιχούν στις στήλες του πίνακα ισοτιμίας, ενώ στην άλλη κατηγορία κορυφών ανήκουν οι κόμβοι ελέγχου (check nodes), οι οποίοι αντιστοιχούν στις γραμμές του. Μπορούμε να πούμε πως κάθε κόμβος μεταβλητής αναπαριστά κάποιο σύμβολο δεδομένων και κάθε κόμβος ελέγχου μια εξίσωση ελέγχου ισοτιμίας. Οι άσοι του πίνακα ισοτιμίας αντιστοιχούν στις ακμές του γράφου.

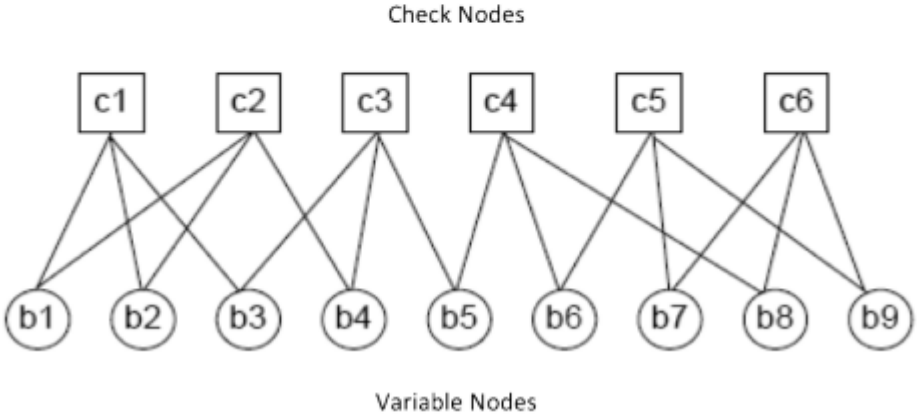
Έστω το στοιχείο (i, j) του πίνακα H είναι άσος. Σε αυτήν την περίπτωση, υπάρχει μία ακμή η οποία συνδέει τον κόμβο μεταβλητής ο οποίος αντιστοιχεί στην στήλη j με τον κόμβο ελέγχου, ο οποίος αντιστοιχεί στην γραμμή i. Αντίστοιχα, τοποθετούνται οι υπόλοιπες ακμές που συνδέουν τους κόμβους μεταβλητής με τους κόμβους ελέγχου σύμφωνα με την κατανομή των άσων στον πίνακα ισοτιμίας του κώδικα. Ο αριθμός των ακμών που συνδέονται σε κάθε κόμβο ονομάζεται βαθμός (degree) του συγκεκριμένου κόμβου. Ο βαθμός των κόμβων μεταβλητής είναι σταθερός για ένα κανονικό κώδικα LDPC όπως και εκείνος των κόμβων ελέγχου. Οι δύο αυτές παράμετροι αντιστοιχούν στις παραμέτρους d_u και d_c που αναφέραμε παραπάνω, και για το λόγο αυτό θα χρησιμοποιήσουμε τον ίδιο συμβολισμό, αφού περιγράφουν ουσιαστικά τα ίδια μεγέθη. Στην περίπτωση των μη κανονικών κωδίκων, ο βαθμός των κόμβων περιγράφεται από ειδικά πολυώνυμα κατανομής βαθμού.

Στο σχήμα (5) φαίνεται ένας πίνακας ελέγχου ισοτιμίας και το αντίστοιχο διάγραμμα Tanner. Ο κώδικας του σχήματος είναι ένας (9, 2, 3) LDPC κώδικας. Η βασική ιδιότητα του συγκεκριμένου και κάθε LDPC κώδικα η ακόλουθη:

Όλα τα ψηφία που συνδέονται σε ένα κόμβο ελέγχου έχουν modulo-2 άθροισμα ίσο με το μηδέν ή, ισοδύναμα, έχουν αποτέλεσμα μηδέν στην πράξη XOR .

Αυτή η ιδιότητα ονομάζεται περιορισμός ισοτιμίας ελέγχου (parity check constraint), και κάθε λέξη των 9 bits για τον συγκεκριμένο κώδικα είναι έγκυρη λέξη αν και μόνο αν ικανοποιεί όλα τα parity check constraints, που στην συγκεκριμένη περίπτωση είναι 6 (βλέπε Σχέση (6)).

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



Σχήμα 5: Παράδειγμα διαγράμματος Tanner για συγκεκριμένο πίνακα ελέγχου ισοτιμίας H

Ένα πολύ σημαντικό χαρακτηριστικό ενός διαγράμματος Tanner είναι η ύπαρξη κύκλων στο διάγραμμα. Ένας κύκλος μήκους g είναι μία κλειστή διαδρομή η οποία περιλαμβάνει g το πλήθος ακμές. Η παρουσία κύκλων στο διάγραμμα Tanner ενός κώδικα είναι ανεπιθύμητη, διότι περιορίζει την απόδοση της αποκωδικοποίησης, πλην όμως ανέφικτη, καθώς η απουσία κύκλων προϋποθέτει απόλυτη ανεξαρτησία μεταξύ των bits που ελέγχει η κάθε γραμμή του πίνακα ισοτιμίας. Δεδομένου ότι η αποφυγή των κύκλων δεν είναι δυνατή, επιδιώκεται η παρουσία κύκλων με το μέγιστο δυνατό μήκος. Το μήκος όμως των κύκλων είναι ανάλογο των διαστάσεων του πίνακα ισοτιμίας γεγονός που θέτει όρια στο μέγιστο δυνατό τους μήκος. Ο μικρότερος κύκλος που μπορεί να εμφανισθεί είναι ένας κύκλος μήκους 4. Το μήκος του μικρότερου κύκλου ενός κώδικα είναι εκείνο που μας ενδιαφέρει κυρίως, και για αυτό το λόγο αποτελεί χαρακτηριστικό του κώδικα και ονομάζεται girth.

2.4 Πλεονεκτήματα - Μειονεκτήματα

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

Ένα ακόμη σημαντικό πλεονέκτημα των κωδίκων LDPC, είναι το γεγονός πως οι χρησιμοποιούμενοι αλγόριθμοι αποκωδικοποίησης, πέρα από την διόρθωση σφαλμάτων,

προσφέρουν σαφή και έγκυρη ανίχνευση των περιπτώσεων εκείνων που η αποκωδικοποίηση αποτυγχάνει. Είναι θεωρητικά πιθανό, με κατάλληλη επιλογή κώδικα, ότι μπορούμε να ελαττώσουμε κατά πολύ την πιθανότητα η αποκωδικοποίηση να καταλήξει σε κωδική λέξη η οποία να είναι μεν έγκυρη, χωρίς όμως να είναι εκείνη που μεταδόθηκε. Η πιθανότητα αυτή σχετίζεται άμεσα με την ελάχιστη απόσταση ανάμεσα σε δύο έγκυρες κωδικές λέξεις του κώδικα. Όσο μεγαλύτερη είναι η απόσταση αυτή, τόσο μεγαλύτερη είναι και η πιθανότητα αποφυγής αυτής της περίπτωσης.

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

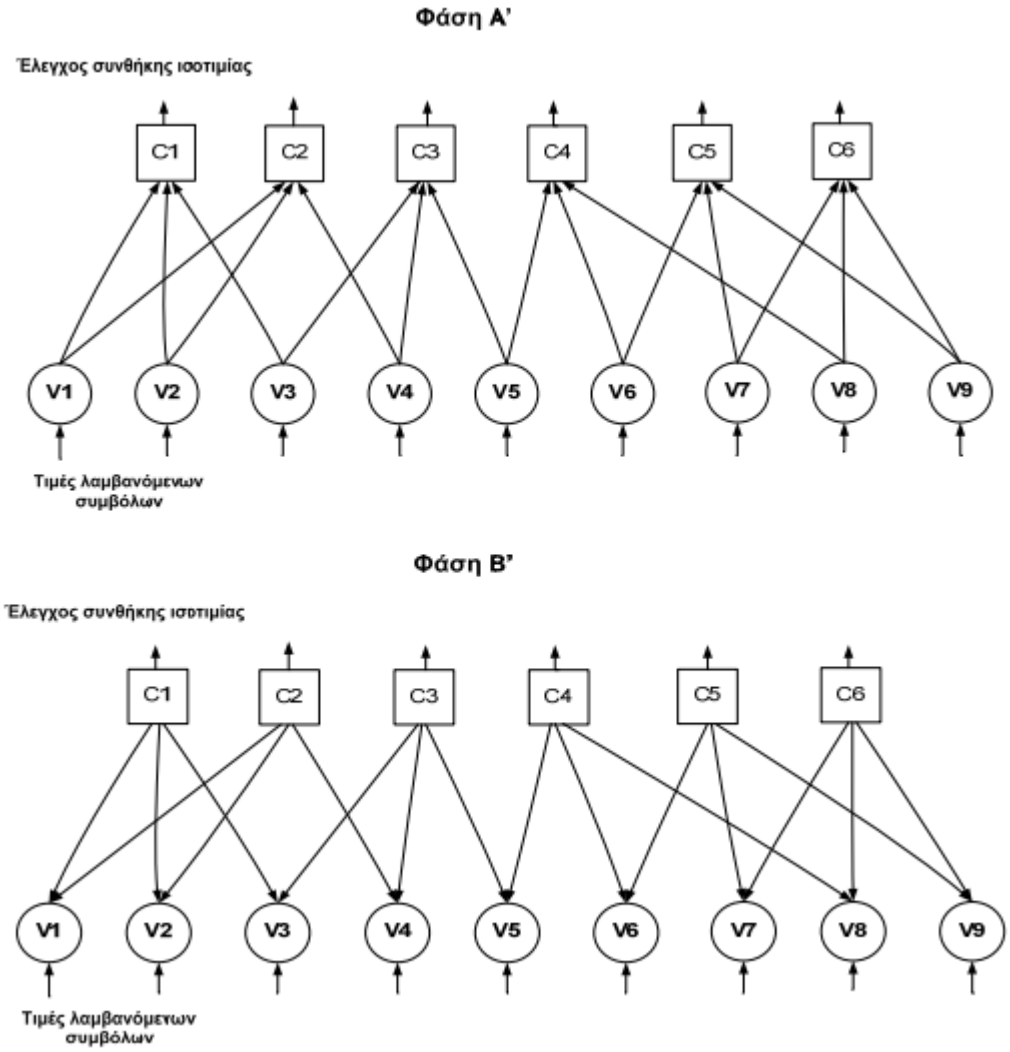
Το βασικό αρνητικό τους στοιχείο είναι πως για να επιτευχθεί αυτός ο ρυθμός μετάδοσης χωρίς λάθη πρέπει να χρησιμοποιηθεί κώδικας με πολύ μεγάλο μήκος λέξης. Αυτό σημαίνει πως απαιτείται πολύ μεγάλος πίνακας ελέγχου ισοτιμίας. Όσο αραιός και να είναι ένας τόσο μεγάλος πίνακας, ο αριθμός των μη μηδενικών στοιχείων είναι εξαιρετικά μεγάλος, με αποτέλεσμα η αποκωδικοποίηση να είναι εξαιρετικά απαιτητική από άποψη πολυπλοκότητας.

2.5 Αποκωδικοποίηση

Η αποκωδικοποίηση είναι μία διαδικασία κατά την οποία ο αποκωδικοποιητής αποφαίνεται για την ακολουθία συμβόλων που μεταδόθηκε, έχοντας ως δεδομένα την ακολουθία συμβόλων που έλαβε. Με βάση τον message-passing αλγόριθμο, ο οποίος αποτελεί κλασικό τρόπο αποκωδικοποίησης κωδικών LDPC και στον οποίο θα αναφερθούμε εκτενώς στη συνέχεια, μηνύματα (messages) στέλνονται μεταξύ των κόμβων του διαγράμματος Tanner, που περιγράφει τον κώδικα, και εκτελούνται κάποιοι υπολογισμοί στους κόμβους αυτούς. Όπως αναφέραμε στο προηγούμενο κεφάλαιο, κάθε κόμβος μεταβλητής αναπαριστά κάποιο σύμβολο δεδομένων και κάθε κόμβος ελέγχου μια εξίσωση ελέγχου ισοτιμίας.

Ας υποθέσουμε ότι κάποια από τα ψηφία-σύμβολα μιας κωδικής λέξης, καθώς αυτή διαδίδεται μέσω του καναλιού επικοινωνίας, έχουν υποστεί αλλοίωση, εξαιτίας της παρουσίας θορύβου στο κανάλι. Αφού φτάσει η αλλοιωμένη κωδική λέξη στον αποκωδικοποιητή, κάθε σύμβολο της μπαίνει ως είσοδος στον αντίστοιχο κόμβο μεταβλητής. Ο κάθε κόμβος μεταβλητής προκειμένου να πάρει μια απόφαση σχετικά με το αν το bit που έλαβε είναι σωστό ή λάθος, «ρωτά» όλους τους γειτονικούς του κόμβους ελέγχου, ποια είναι η «γνώμη» τους για την τιμή του συγκεκριμένου bit. Καθένας από αυτούς τους κόμβους ελέγχου ρωτά, στη συνέχεια, τους υπόλοιπους γειτονικούς του κόμβους μεταβλητής ποια είναι η τρέχουσα τιμή-εκτίμησή τους για τα δικά τους bits (Φάση Α) και στέλνει στον αρχικό κόμβο μεταβλητή μια απάντηση, η οποία είναι συνάρτηση των τιμών αυτών (Φάση Β). Στην ουσία, με αυτό τον τρόπο ελέγχεται η άρτια ή περιττή ισοτιμία των άσσων της ελεγχόμενης ομάδας ψηφίων. Επειδή, όμως, κάθε κόμβος μεταβλητής έχει περισσότερους από έναν γειτονικούς κόμβους ελέγχου, έχει και περισσότερες από μία γνώμες για το αν είναι σωστό το δικό του bit. Πρέπει με κάποιο τρόπο να επεξεργαστεί αυτές τις γνώμες και να βγάλει κάποιο συμπέρασμα για την τιμή του συγκεκριμένου bit. Για

παράδειγμα, θα μπορούσε να «πιστέψει» την πλειοψηφία. Η παραπάνω διαδικασία αποτελεί μία επανάληψη του message passing αλγορίθμου. Με κάθε επιπλέον επανάληψη που γίνεται, αυξάνεται η πιθανότητα να είναι σωστή η εκτίμηση των κόμβων μεταβλητής για τις τιμές των ψηφίων της κωδικής λέξης. Στο Σχήμα (6) απεικονίζεται η διαδικασία ανταλλαγής μηνυμάτων που μόλις περιγράψαμε, για τον (9,2,3)-LDPC κώδικα που αναφέραμε στο κεφάλαιο 2.3.



Σχήμα 6: Αποκωδικοποίηση με χρήση του Message-Passing αλγορίθμου δύο φάσεων.

Κατά τη μετάδοση δεδομένων, κάποια (ή όλα) από τα λαμβανόμενα σύμβολα διαφέρουν από εκείνα που εισήχθησαν στο κανάλι, ορισμένα πολύ και άλλα λιγότερο. Στην περίπτωση που έχουμε έναν δυαδικό κώδικα και διαμόρφωση BPSK (Binary Phase Shift Keying) τα μεταδιδόμενα σύμβολα είναι το +1 (λογικό 0) και το -1 (λογικό 1). Εξαιτίας του θορύβου στο κανάλι επικοινωνίας, το σήμα που λαμβάνει ο αποδιαμορφωτής κατά τη δειγματοληψία δεν είναι ίδιο με το αρχικό, αλλά παίρνει ενδιάμεσες τιμές ή ακόμα και ανεστραμμένες, αν υπάρχει πολύς θόρυβος. Υπάρχουν δύο προσεγγίσεις όσον αφορά τον τρόπο έκφρασης της τιμής κάθε λαμβανομένου συμβόλου, καθώς και των ενδιάμεσων εκτιμήσεων, κατά τη διάρκεια της αποκωδικοποίησης.

Η προφανής, η οποία ονομάζεται hard decision, είναι να θεωρήσουμε πως το πρόσημο του λαμβανομένου συμβόλου είναι αρκετό για να κάνουμε μία πρώτη εκτίμηση για το μεταδιδόμενο

σύμβολο. Αν, εφαρμόζοντας αυτή τη θεώρηση σε όλα τα σύμβολα της κωδικής λέξης, προκύπτει μη έγκυρη κωδική λέξη, τότε προσπαθούμε να διορθώσουμε τα όποια σφάλματα, διαχειριζόμενοι πλέον δυαδικά ψηφία. Επομένως αντιστοιχίζουμε ένα μόνο δυαδικό ψηφίο σε κάθε σύμβολο που λαμβάνουμε. Με τον τρόπο αυτό κβαντίζουμε άμεσα τα λαμβανόμενα σύμβολα και χάνουμε την όποια πληροφορία μπορούμε να αντλήσουμε από αυτά. Η αποκωδικοποίηση αυτή είναι χαμηλής πολυπλοκότητας, οδηγεί όμως σε σημαντική απώλεια ληφθείσας πληροφορίας. Για παράδειγμα, αν υποψιαζόμαστε ότι ένα από δύο λαμβανόμενα σύμβολα +0,13 και +1,2 είναι λανθασμένα, δηλαδή λόγω του θορύβου έχει αλλάξει το πρόσημό τους, τότε είναι πολύ λογικό να θεωρήσουμε πιθανότερο αυτό να έχει συμβεί στο πρώτο σύμβολο. Αυτή την εκτίμηση δεν θα μπορέσουμε να την κάνουμε αν εξαρχής εξισώσουμε τα δύο λαμβανόμενα σύμβολα θεωρώντας ότι είναι +1.

Η δεύτερη προσέγγιση χρησιμοποιεί περισσότερα από ένα δυαδικά ψηφία, τα λεγόμενα soft bits, για την αναπαράσταση των λαμβανόμενων συμβόλων και των μηνυμάτων που ανταλλάσσονται μεταξύ κόμβων ελέγχου και μεταβλητής. Η ληφθείσα πληροφορία, η οποία προκύπτει από την ακριβή τιμή του λαμβανόμενου συμβόλου, διατηρείται καθ' όλη τη διάρκεια της αποκωδικοποίησης και αξιοποιείται κατά τον καλύτερο δυνατό τρόπο. Προκειμένου να εκτιμηθεί, κατά την αποκωδικοποίηση, αν η κωδική λέξη στην οποία αυτή καταλήγει είναι έγκυρη (στο τέλος κάθε επανάληψης ή κάποιου αριθμού επαναλήψεων) γίνεται μία προσωρινή κβάντιση των εκτιμώμενων soft bits των συμβόλων, χωρίς όμως αυτά να χάνονται. Αντίθετα, αν η κωδική λέξη δεν είναι έγκυρη, η αποκωδικοποίηση συνεχίζεται με τις «soft» τιμές των συμβόλων. Η εκτίμηση αυτή αποκαλείται soft decision και επιτυγχάνει αποτελεσματικότερη αποκωδικοποίηση. Το κόστος της, από άποψη υπολογιστικής πολυπλοκότητας, όμως. είναι σαφώς μεγαλύτερο από αυτό της hard decision εκτίμησης.

Πριν περάσουμε στην παρουσίαση κάποιων βασικών αλγορίθμων αποκωδικοποίησης κωδικών LDPC, είναι σκόπιμο να σχολιάσουμε πως από την περιγραφή του message-passing αλγορίθμου διαφαίνεται μία αντιστοιχία ανάμεσα στο διάγραμμα Tanner ενός κώδικα και στο κύκλωμα που υλοποιεί την αποκωδικοποίηση. Το κύκλωμα αυτό θα μπορούσε να αποτελείται από δύο ειδών υπολογιστικές μονάδες, έναν τύπο για τους κόμβους ελέγχου (Check Processing Unit - CPU) και έναν για τους κόμβους μεταβλητής (Variable Processing Unit - VPU). Κάθε κόμβος του διαγράμματος Tanner αντιστοιχεί σε μία επεξεργαστική μονάδα του αντίστοιχου τύπου. Μεταξύ των δύο τύπων υπολογιστικών μονάδων ανταλλάσσονται εκτιμήσεις-μηνύματα, τις οποίες και επεξεργάζονται. Η διαδικασία αυτή είναι επαναληπτική. Ο τρόπος επεξεργασίας των εκτιμήσεων καθορίζεται από τον χρησιμοποιούμενο αλγόριθμο αποκωδικοποίησης. Αναδεικνύεται, έτσι, η σπουδαιότητα της παρουσίας λίγων άσων στον πίνακα ισοτιμίας, το πλήθος των οποίων είναι ανάλογο με το πλήθος των ανταλλασσόμενων μηνυμάτων. Επιπλέον, οι διαστάσεις του πίνακα καθορίζουν το πλήθος των υπολογιστικών μονάδων και, κατ’ επέκταση, την πολυπλοκότητα του κυκλώματος αποκωδικοποίησης.

2.6 Ο αλγόριθμος Belief Propagation(BP)

Στην υποενότητα αυτή θα περιγραφεί ένας θεμελιώδης τρόπος αποκωδικοποίησης, ο οποίος βασίζεται στα διμερή διγραφήματα, Tanner Graphs, τα οποία παρουσιάστηκαν προηγουμένως. Ο ακόλουθος αλγόριθμος, λειτουργεί επιτυχώς για αραιούς (sparse) κώδικες, των οποίων οι parity check πίνακες περιέχουν μικρό αριθμό μη μηδενικών στοιχείων, τα οποία

περιορίζονται στο 10% των συνολικών στοιχείων του πίνακα. Στην κατηγορία των sparse κωδίκων φυσικά, ανήκουν και οι LDPC κώδικες τους οποίους μελετάμε στην παρούσα εργασία. Σε επόμενες ενότητες αναφερόμαστε σε αλγορίθμους αποκωδικοποίησης, οι οποίοι βασίζονται στον Belief Propagation, στην παρούσα φάση όμως θα εξηγήσουμε τη γενική φιλοσοφία λειτουργίας του BP, καθώς έτσι θα φανεί και η σημασία των δομών Tanner Graphs. Για την περιγραφή της λειτουργίας του αλγορίθμου, θα βασιστούμε στη μορφή του Tanner Graph του Σχήματος (5).

Ο αλγόριθμος BP είναι επαναληπτικός και κάθε επανάληψή του απαρτίζεται από δύο στάδια: πρώτον, οι variable κόμβοι του γράφου, ανανεώνονται και στέλνουν την πληροφορία τους στους check κόμβους με τους οποίους συνδέονται και δεύτερον: οι check κόμβοι ανανεώνονται και στέλνουν πληροφορία πίσω στους variable κόμβους. Αναλυτικότερα, σε πρώτη φάση, αρχικοποιείται κάθε variable κόμβος c_i με μία τιμή, η οποία αντιστοιχεί στην πιθανοφάνεια (likelihood) ο συγκεκριμένος κόμβος να φέρει την τιμή 0 ή 1. Πιο συγκεκριμένα η πιθανοφάνεια αυτή, εκφράζεται με τον λόγο $\frac{P(x=0)}{P(x=1)}$, ή συχνότερα με τη λογαριθμημένη έκφραση $log\left(\frac{P(x=0)}{P(x=1)}\right)$, η οποία καλείται LLR. Αφού κάθε variable κόμβος λάβει τις τιμές αρχικοποίησης μέσω των LLRs, αναλαμβάνει να αποστείλει την πληροφορία αυτή στους check κόμβους b_i. Οι check κόμβοι στη συνέχεια, αντιστοιχούν τις τιμές που έλαβαν σε bits και ελέγχουν αν έλαβαν τη σωστή λέξη. Αν η ληφθείσα λέξη δεν είναι η σωστή, ξεκινάει η επαναληπτική διαδικασία: Οι check κόμβοι εκτελούν συγκεκριμένους υπολογισμούς, οι οποίοι διαφέρουν ανάλογα με τον αλγόριθμο υλοποίησης και στη συνέχεια βάσει των υπολογισμών αυτών, ενημερώνουν τους variable κόμβους, ώστε αυτοί να βελτιώσουν τις εκτιμήσεις τους (πιθανοφάνειες). Η διαδικασία αυτή, όπως θα δούμε και στους αλγορίθμους που ακολουθούν, ονομάζεται οριζόντιο βήμα. Οι variable κόμβοι, λαμβάνοντας υπόψιν τα μηνύματα που δέχθηκαν από τους check κόμβους, κάνουν νέους υπολογισμούς συνδυάζοντας και τα αρχικά LLRs και ενημερώνουν ξανά τους check κόμβους. Μετά από κάθε επανάληψη ανταλλαγής μηνυμάτων μεταξύ των δύο σετ κόμβων, εξετάζεται αν τα checks ικανοποιούνται, δηλαδή αν το XOR των εισόδων τους είναι ίσο με το 0. Στην ουσία λοιπόν, εφαρμόζεται η σχέση ελέγχου 5. Όσο οι check κόμβοι δεν ικανοποιούνται, ο αλγόριθμος εκτελείται επαναληπτικά.

2.7 *Ο αλγόριθμος Sum-Product*

Ο αλγόριθμος είναι βασισμένος στη λογική του Belief Propagation και συνεπώς στην ανταλλαγή μηνυμάτων ανάμεσα στα δύο σετ κόμβων στα διγραφήματα Tanner. Γενικά, στις επόμενες ενότητες, συμβολίζονται με Q τα μηνύματα που στέλνουν οι variable στους check κόμβους, ενώ με R τα μηνύματα των check στους variable κόμβους[15]. Επίσης, παρακάτω θα χρησιμοποιηθούν οι συμβολισμοί: n: σύνολο συμβόλων κωδικής λέξης, m: αριθμός coded συμβόλων και k = n − m τα information σύμβολα. Αρχικά, το ληφθέν σύμβολο μπορεί να λάβει είτε την τιμή 0 είτε την τιμή 1. Όπως αναφέραμε στο τέλος του προηγούμενου κεφαλαίου, από τη φάση της αποδιαμόρφωσης εξάγονται οι πιθανότητες το ληφθέν σύμβολο να έχει την τιμή 0 ή την τιμή 1. Με βάση τις πιθανότητες αυτές, υπολογίζουμε τις πιθανοφάνειες (κανονικοποιημένες πιθανότητες), οι οποίες εισάγονται σε έναν πίνακα f διαστάσεων 2 × n. Παρακάτω φαίνεται η μορφή του πίνακα f:

$$f = \begin{bmatrix} g_1^{(0)} & g_2^{(0)} & \dots & g_n^{(0)} \\ g_1^{(1)} & g_2^{(1)} & \dots & g_n^{(1)} \end{bmatrix} \quad (9)$$

Έστω $y_n^{(x)}$, η πιθανότητα το n-οστό σύμβολο τα λάβει την τιμή x, όπου στην περίπτωση των binary αποκωδικοποιητών, το x μπορεί να λάβει τις τιμές 0 ή 1. Οι τιμές $y_n^{(x)}$, προκύπτουν από τη γκαουσσιανή συνάρτηση πυκνότητας πιθανότητας, ουσιαστικά υπολογίζοντας την ευκλείδεια απόσταση από κάθε σημείο του αστερισμού. Καταλαβαίνουμε λοιπόν, πως στην ουσία δεν αποτελούν τιμές πιθανοτήτων που είναι "φραγμένες" ανάμεσα στο 0 και το 1. Θεωρώντας διαμόρφωση BPSK η μορφή των πιθανοτήτων $y_n^{(x)}$ θα είναι η εξής:

$$y_n^{(0)} = \left(\frac{1}{2\sqrt{2\pi}}\right) e^{-[\left(r_n^{(I)}-1\right)]^2/2s^2} \quad (10)$$

$$y_n^{(1)} = \left(\frac{1}{2\sqrt{2\pi}}\right) e^{-[\left(r_n^{(I)}+1\right)]^2/2s^2} \quad (11)$$

Ένας απλοϊκός τρόπος υπολογισμού των πιθανοφανειών, είναι αρχικά, να αθροίσουμε τις τιμές $y_n^{(0)}$ και $y_n^{(1)}$ και έπειτα να υπολογίσουμε τις πιθανοφάνειες (Likelihoods). Στην ουσία δηλαδή κανονικοποιούμε τις πιθανότητες ως εξής:

$$g_n^{(0)} = \frac{y_n^{(0)}}{y_n^{(0)} + y_n^{(1)}} \quad (12)$$

$$g_n^{(1)} = \frac{y_n^{(1)}}{y_n^{(0)} + y_n^{(1)}} \quad (13)$$

Οι παραπάνω κανονικοποιημένες πιθανότητες, στα ερχόμενα κεφάλαια θα αναφέρονται ως πιθανοφάνειες. Βέβαια οι πιθανοφάνειες αυτές, διαφέρουν από τα Likelihoods με τη συνήθη έννοια. Ο επικρατέστερος ίσως υπολογισμός που αφορά τα Likelihoods, είναι το LLR, το οποίο αναφέρθηκε στη σύντομη περιγραφή του αλγορίθμου Belief Propagation. Στο σημείο αυτό, υποθέτουμε πως ο πίνακας ελέγχου ισοτιμίας, έχει διαστάσεις m × n. Συνεχίζοντας, ο πίνακας πιθανοτήτων f, αρχικοποιεί έναν πίνακα Q, ο οποίος ορίζεται όπως φαίνεται παρακάτω:

$$Q = \begin{bmatrix} g_1^{(0)} & g_2^{(0)} & \cdots & g_1^{(0)} \\ g_1^{(1)} & g_2^{(1)} & & g_1^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{(0)} & g_2^{(0)} & \cdots & g_1^{(0)} \\ g_1^{(1)} & g_1^{(1)} & & g_1^{(1)} \end{bmatrix} \quad (14)$$

Ο πίνακας αυτός, όπως ήδη αναφέραμε, αντιπροσωπεύει τα μηνύματα που στέλνει κάθε variable κόμβος (στήλη) προς κάθε αντίστοιχο check κόμβο (γραμμή). Η μορφή του πίνακα Q, βασίζεται στη μορφή του πίνακα ελέγχου ισοτιμίας. Στις θέσεις που ο parity check matrix έχει μη μηδενικά στοιχεία, ο πίνακας Q περιέχει τις πιθανότητες (υποπίνακες των 2 γραμμών) που είναι αποθηκευμένες στον πίνακα f. Αντίθετα, στις θέσεις που ο πίνακας ελέγχου ισοτιμίας έχει μηδενική τιμή, ο πίνακας Q, περιέχει μηδενικούς υποπίνακες. Τέλος ο πίνακας Q αντιστοιχίζεται κατά στήλη με τον πίνακα f, δηλαδή τα μη μηδενικά δεδομένα κάθε στήλης του πίνακα Q είναι ίδια με τα δεδομένα της αντίστοιχης στήλης του πίνακα f. Κατανοούμε λοιπόν, πως στο βήμα της αρχικοποίησης, οι variable κόμβοι, στέλνουν στους check κόμβους τις τιμές των πιθανοτήτων που έλαβαν από το κανάλι.

Horizontal step: Μετά τις αρχικοποιήσεις, ακολουθεί η διαδικασία του οριζόντιου βήματος (horizontal step), στην οποία υπολογίζεται ο πίνακας R, δηλαδή υπολογίζονται τα μηνύματα που κάθε check κόμβος (γραμμή) στέλνει προς κάθε αντίστοιχο variable κόμβο (στήλη). Η διαδικασία υπολογισμού, συνοψίζεται ως εξής:

Ο αλγόριθμος, υπολογίζει την πιθανότητα r_{mn}:

$$r_{mn}(x) = \sum_{C_n = x} P(z_m|c) P(c|c_n = x) \quad (15)$$

όπου $c_n = x$, είναι το n-οστό coded bit, του οποίου η τιμή είναι γνωστή και συμβολίζεται με x. Το c συμβολίζει το διάνυσμα των υπολοίπων coded bits, τα οποία αν θεωρήσουμε ότι ο πίνακας ελέγχου ισοτιμίας έχει βάρη γραμμών και στηλών, a και b αντίστοιχα, το c μπορεί να λάβει 2^{b-1} διαφορετικές τιμές. Από τις τιμές αυτές όμως, δεν ικανοποιούν όλες την εξίσωση ελέγχου ισοτιμίας $z_m = 0$. Αναζητούμε δηλαδή όλες τις πιθανές δυαδικές ακολουθίες που ικανοποιούν την εξίσωση αυτή. Γενικά η εξίσωση (15), απλοποιείται στην ακόλουθη μορφή:

$$r_{mn}(x) = \sum_{C_n = x} P(z_m|c) \prod_{n' \in N_m \backslash n} q_{mn'}(x) \quad (16)$$

Όπου $P(z_m|c)$ είναι είτε 0 είτε 1. Οι πιθανότητες $r_{mn}(x)$ τοποθετούνται σε έναν πίνακα R:

$$R = \begin{bmatrix} r_{11}^{(0)} & r_{12}^{(0)} & \cdots & r_{1n}^{(0)} \\ r_{11}^{(1)} & r_{12}^{(1)} & \cdots & r_{1n}^{(0)} \\ \vdots & \vdots & \ddots & \cdots \\ r_{m1}^{(0)} & r_{m2}^{(0)} & \cdots & r_{mn}^{(0)} \\ r_{m1}^{(1)} & r_{m2}^{(1)} & \cdots & r_{mn}^{(1)} \end{bmatrix} \quad (17)$$

Όπου κάθε στοιχεία στον πίνακα ελέγχου ισοτιμίας έχει ένα αντίστοιχο 1×2 διάνυσμα στήλης που περιέχει τις πιθανότητες $r_{mn}(0)$ και $r_{mn}(1)$.

Vertical step: Επόμενο βήμα του αλγορίθμου είναι το κάθετο βήμα(vertical step). Το βήμα αυτό ανανεώνει τις πιθανότητες (μηνύματα variable προς check κόμβους) q_{mn} του πίνακα Q, εφαρμόζοντας τον κανόνα του Bayes ως εξής:

$$q_{mn}(x) = p(c_n = x | z_m = 0, m' \in M_n/m) \\ = \frac{P(c_n = x) P(\{z_m = 0, m' \in M_n/m\} | c_n = x)}{P(\{z_m = 0, m' \in M_n/m\})} \quad (18)$$

όπου από τον αρχικό πίνακα πιθανοτήτων εξάγουμε το συμπέρασμα πως $f_n^x = P(c_n = x)$, άρα μετασχηματίζουμε την προηγούμενη εξίσωση στην εξής τελική σχέση :

$$q_{mn}(x) = \beta_{mn} f_n^x \prod_{m' \in M_n/m} r_{m'n}(x) \quad (19)$$

Όπου β_{mn} είναι μία σταθερά που χρησιμοποιείται για να εξασφαλίσει ότι ισχύει $\sum q_{mn}(x) = 1$ και είναι:

$$\beta_{mn} = \frac{1}{\sum f_n^x \prod_{m' \in M_n/m} r_{m'n}(x)} \quad (20)$$

Τα νέα στοιχεία q_{mn}, που υπολογίστηκαν μέσω της εξίσωσης (19), θα ανανεώσουν τώρα τον πίνακα Q, ο οποίος θα έχει ίδια ακριβώς μορφή με τον πίνακα της εξίσωσης (14). Δηλαδή κάθε στοιχείο του q_{mn} θα αποτελεί έναν υποπίνακα 1 × 2 όπως φαίνεται στην εξίσωση (21):

$$Q = \begin{bmatrix} q_{11}(0) & q_{12}(0) & \cdots & q_{1(n-1)}(0) & q_{1n}(0) \\ q_{11}(1) & q_{12}(1) & \cdots & q_{1(n-1)}(1) & q_{1n}(1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{m1}(0) & q_{m2}(0) & \cdots & q_{m(n-1)}(0) & q_{mn}(0) \\ q_{m1}(1) & q_{m2}(1) & \cdots & q_{m(n-1)}(1) & q_{mn}(1) \end{bmatrix} \quad (21)$$

Ο πίνακας αυτός, χρησιμοποιείται στο οριζόντιο βήμα της επόμενης επανάληψης (iteration) του αλγορίθμου αποκωδικοποίησης.

Μετά από την εύρεση του πίνακα R, είμαστε έτοιμοι να υπολογίσουμε τον τελικό πίνακα πιθανοτήτων Q' , από τον οποίο θα μπορέσουμε να ανιχνεύσουμε το μήνυμα που στάλθηκε. Συγκεκριμένα, χρησιμοποιούμε τον πίνακα R στη σχέση:

$$q_n(x) = \beta_n f_n^x \prod_{m \in M_n} r_{mn}(x) \quad (22)$$

Ο πίνακας Q' που παράγεται έχει την ίδια μορφή με τον αρχικό πίνακα πιθανοτήτων f, δηλαδή είναι ως εξής :

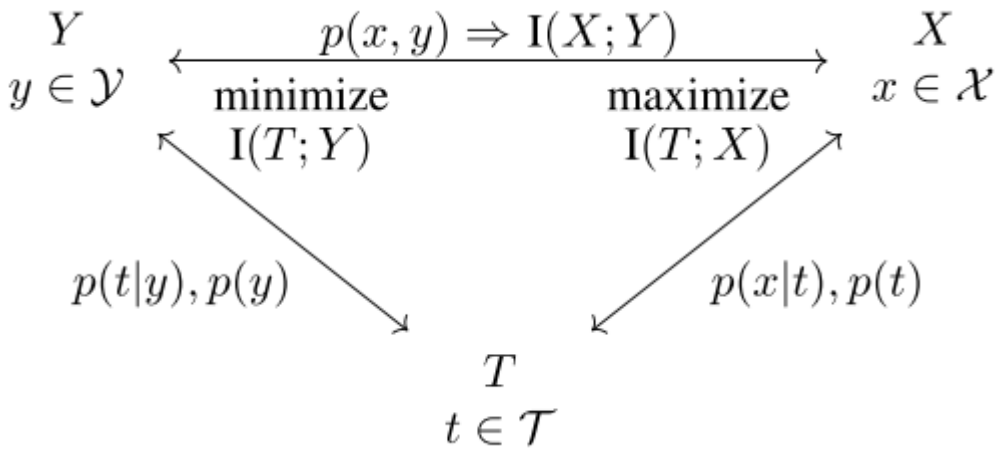
$$Q' = \begin{bmatrix} q_1^{(0)} & q_2^{(0)} & \cdots & q_n^{(0)} \\ q_1^{(1)} & q_2^{(1)} & \cdots & q_n^{(1)} \end{bmatrix} \quad (23)$$

Ουσιαστικά, στον πίνακα αυτό καλούμαστε να εντοπίσουμε την θέση που εμφανίζεται η μεγαλύτερη πιθανότητα εκ των δύο πιθανοτήτων κάθε στήλης, όπου υπενθυμίζουμε πως ο αριθμός των στηλών εκφράζει το μήκος της κωδικής λέξης. Ουσιαστικά, δηλαδή θα χρησιμοποιήσουμε την απλή εξίσωση argmax σε κάθε στήλη του πίνακα Q' , ώστε να βρούμε τη θέση της μέγιστης πιθανότητας. Αν δεν προκύψει το επιθυμητό αποτέλεσμα, ο αλγόριθμος εκτελείται επαναληπτικά, ανανεώνοντας μέσω του κάθετου βήματος τον πίνακα Q. Ένας έγκυρος έλεγχος τερματισμού του αλγορίθμου, είναι να δούμε αν η λέξη που αποκωδικοποιήσαμε, είναι κωδική. Αυτός ο έλεγχος θα πραγματοποιηθεί μέσω της σχέσης H · c T (5), η οποία αν είναι ίση με το μηδενικό διάνυσμα, η λέξη που παράχθηκε από τον decoder είναι κωδική. Συνήθως ένας μέγιστος αριθμός επαναλήψεων για τον αποκωδικοποιητή είναι 5, 10 ή 50, ενώ παράλληλα πραγματοποιείται και έλεγχος τερματισμού μέσω της (5). Αν ο αλγόριθμος δεν καταλήξει σε κωδική λέξη μετά το πέρας του μέγιστου αριθμού των επαναλήψεων που θέσαμε, τότε θεωρούμε πως η αποκωδικοποίηση απέτυχε.

3 Information Bottleneck

3.1 Γενική περιγραφή της μεθόδου Information Bottleneck

Η μέθοδος Information Bottleneck εισήχθη από τον N. Tishby [10]. Είναι μία μαθηματική μέθοδος που συνδέεται με το rate-distortion theory, αλλά παρά κάποιες κοινές ιδέες μεταξύ τους, η μέθοδος Information Bottleneck και το rate-distortion theory έχουν αρκετές διαφορές. Ενώ το rate-distortion theory στοχεύει στην ελαχιστοποίηση ενός αναμενόμενου μέτρου παραμόρφωσης με την συμπίεση των αρχικών δεδομένων, η μέθοδος Information Bottleneck χρησιμοποιεί την έννοια της συνάφειας μέσω μιας άλλης μεταβλητής η οποία θα πρέπει να διατηρηθεί υπό συμπίεση. Η γενική ιδέα της μεθόδου Information Bottleneck φαίνεται στο σχήμα(7)



Σχήμα 7: Αναπαράσταση της γενικής ιδέας της μεθόδου Information Bottleneck

Έστω διακριτές τυχαίες μεταβλητές X και Y, όπου Y παρατηρείται ενώ το X όχι. Οι μεταβλητές X και Y μοιράζονται αμοιβαίες πληροφορίες:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \qquad (24)$$

Επιπλέον εισάγεται μία τρίτη μεταβλητή T. Η μεταβλητή T θα πρέπει να είναι μία συμπιεσμένη αναπαράσταση του Y. Οι μεταβλητές X → Y → T σχηματίζουν μία αλυσίδα Markov. Στην μέθοδο Information Bottleneck η μεταβλητή Y ονομάζεται παρατηρούμενη μεταβλητή. Η X ονομάζεται σχετική μεταβλητή και το T είναι η μεταβλητή συμπίεσης. Η βασική ιδέα είναι να συμπιέσουμε το Y στο T και έτσι να ελαχιστοποιήσουμε το I(Y;T) με τρόπο τέτοιο ώστε να διατηρείται η μέγιστη

δυνατή αμοιβαία πληροφορία I(T;X). I(X;Y) είναι το πάνω όριο για I(T;X) αφού η επεξεργασία του Y δεν μπορεί να παράγει καμία επιπλέον πληροφορία για το X. Η σχέση συμπίεσης μεταξύ Y και T περιγράφεται από την υπό όρους κατανομή p(t|y). Ο Tishby χειρίστηκε το περιορισμένο πρόβλημα βελτιστοποίησης εύρεσης ενός κατάλληλου p(t|y) το οποίο μεγιστοποιεί το I(T;X) και ελαχιστοποιεί το I(Y;T) χρησιμοποιώντας την μέθοδο Lagrance. Ο Lagrangian πολλαπλασιαστής β ≥ 0 είναι ο περιορισμός πληροφορίας για το I(T;X) και συχνά αναφέρεται ως "trade-off" παράμετρος της μεθόδου Information Bottleneck. Ο αντίστοιχος Lagrangian είναι:

$$L\{p(t|y)\} = I(T;Y) - \beta I(T;X) \qquad (25)$$

Και πρέπει να είναι ελαχιστοποιημένο για το σύνολο όλων των έγκυρων κατανομών p(t|y) για ένα σταθερό β ≥ 0. Η επιλογή του β ≥ 0 επιτρέπει την διατήρηση της σχετικής πληροφορίας I(T;X) για συμπίεση. Αν υποθέσουμε ότι το $\beta \to +\infty$ έχει ως αποτέλεσμα η επιθυμητή μέγιστη διατήρηση της σχετικής πληροφορίας υπό τον περιορισμό της καρδινότητας ενός χώρου του T. Οποιαδήποτε ελαχιστοποίηση της λύσης (25) πρέπει να έχει πεπλεγμένη μορφή:

$$p(t|y) = \frac{p(t)}{Z(y,\beta)} \exp\left(-\beta D_{KL}\{p(x|y)|p(x|t)\}\right) \qquad (26)$$

Όπου Z(y,β) είναι μία συνάρτηση κανονικοποίησης που διασφαλίζει ότι $\sum_t p(t|y) = 1 \forall y \in Y$ και

$$D_{KL}\{p(x)|Q(x)\} = \prod_{x\in X} p(x) \log \frac{p(x)}{q(x)} \qquad (27)$$

Είναι η Kullback-Leibler απόκλιση ανάμεσα σε p(x) και q(x). Η λύση της (26) είναι πεπλεγμένη διότι λόγω της σχέσης της αλυσίδας Markov X → Y → T, η κοινή κατανομή p(x,y,t)=p(t|y)p(x,y) και συνεπώς και το p(x|t) εξαρτάται από το p(t|y), δηλαδή

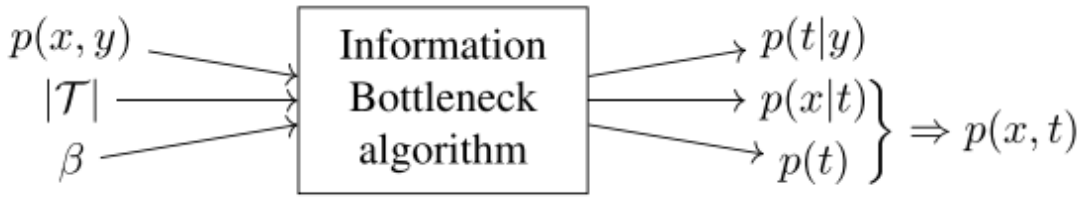
$$p(x|t) = \frac{1}{p(t)} \sum_{y\in Y} p(t|y)p(x,y) \qquad (28)$$

Επιπρόσθετα, το p(t) δίνεται από την σχέση:

$$p(t) = \sum_{y\in Y} p(t|y)p(y) \qquad (29)$$

Αρκετοί υπάρχοντες αλγόριθμοι Information Bottleneck, για παράδειγμα, ο iterative Information Bottleneck algorithm[10] kai o sequential Information Bottleneck algorithm[11], επαναλαμβάνουν

μεταξύ p(t|y), p(t), p(x,y) με σκοπό την ελαχιστοποίηση του L{p(t|y)}. Οι είσοδοι και οι έξοδοι ενός Information Bottleneck αλγόριθμου φαίνονται στο παρακάτω σχήμα (8):



Σχήμα 8: Είσοδοι και έξοδοι ενός Information Bottleneck αλγορίθμου

Ο αλγόριθμος χρησιμοποιεί κοινή κατανομή πιθανότητας p(x,y) που συνδέει την σχετική τυχαία μεταβλητή και την παρατήρηση ως είσοδο. Επιπλέον, εισάγει την καρδινότητα |T| του χώρου συμβάντων της συμπιεσμένης μεταβλητής και της trade-off παραμέτρου β. Ο αλγόριθμος μας δίνει το p(t|y), την οπίσθια κατανομή της σχετικής μεταβλητής δεδομένης της μεταβλητής συμπίεσης δηλαδή το p(x|t) και την κατανομή πιθανότητας p(t). Αφού ισχύει η σχέση p(x, t) = p(x|t)p(t) ουσιαστικά ο αλγόριθμος δίνει και την τιμή της κοινής κατανομής p(x|t) όπως φαίνεται και στο σχήμα(8).

Ο πρώτος iterative Information Bottleneck αλγόριθμος περιγράφεται στο [10]. Είναι άμεσα σχετικός με τον αλγόριθμο Blahut-Arimoto[12] ο οποίος λύνει τα προβλήματα του rate-distortion.

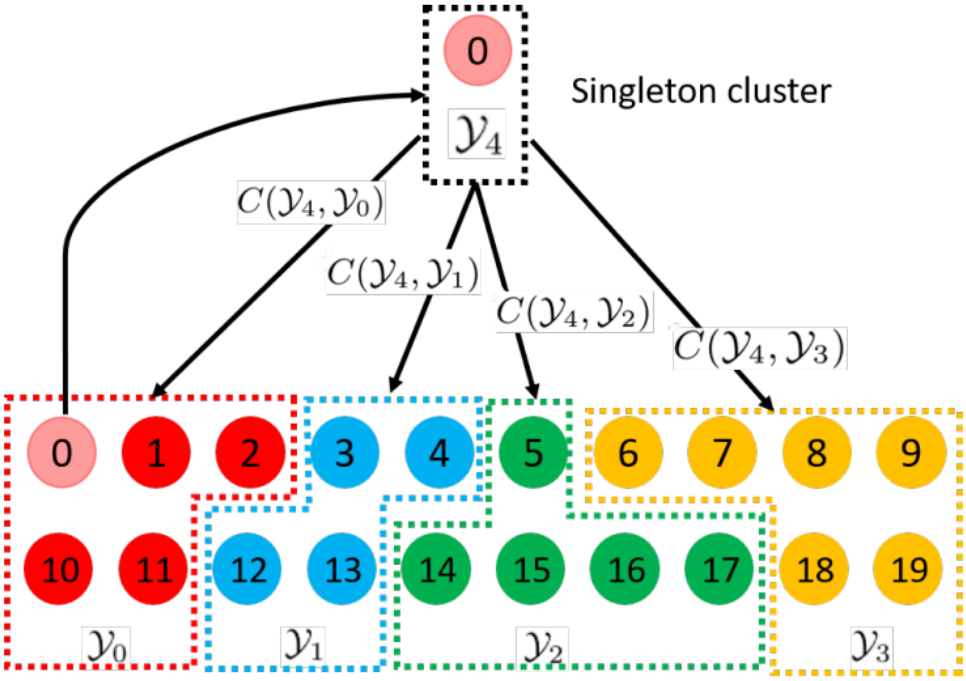
3.2 Περιγραφή του sequential Information Bottleneck αλγορίθμου

Ο sequential Information Bottleneck αλγόριθμος είναι αυτός που θα χρησιμοποιήσουμε στα πλαίσια αυτής της εργασίας. Χωρίς να χάνεται η γενικότητα θεωρούμε τους συγκεκριμένους χώρους συμβάντων της παρατηρήσιμης και της συμπιεσμένης τυχαίας μεταβλητής:

$$\mathcal{Y} = \{0,1,\dots,|\mathcal{Y}|-1\} \quad (30)$$

$$T = \{0,1,\dots,|T|-1\} \quad (31)$$

Η διαδικασία της sequential Information Bottleneck μεθόδου φαίνεται στο σχήμα(9).



Σχήμα 9: Η διαδικασία της sequential Information Bottleneck μεθόδου

Το συγκεκριμένο παράδειγμα αναφέρεται σε |Y| = 20 και |T| = 4. Αρχικά, ο sequential Information Bottleneck αλγόριθμος ταξινομεί |Y| στοιχεία σε |T| ομάδες τυχαία. Τα 20 αυτά στοιχεία y ∈ Y κατηγοριοποιούνται σε Y_t (t ∈ T) ομάδες. Κάθε ένα από τα χρωματιστά πολύγωνα στο σχήμα (9) υποδηλώνει μία ομάδα εδώ συγκεκριμένα τις Y_0,Y_1, Y_2 και Y_3. Τα στοιχεία y ∈ Y φαίνονται ως κύκλοι με την ένδειξη ενός ακέραιου αριθμού. Η σχέση μεταξύ y και t παρουσιάζεται από την συνάρτηση χαρτογράφησης p(t|y).

Στην συνέχεια ο αλγόριθμος εξάγει ένα στοιχείο y του Y_t (στο σχήμα t=0, y=0) και δημιουργεί μία νέα μεμονωμένη ομάδα(singleton cluster στο σχήμα), $Y_{|T|}$ (Y_4 στο σχήμα) η οποία αποτελείται από ένα μόνο στοιχείο. Αυτός ο χειρισμός αλλάζει την συνάρτηση χαρτογράφησης p(y|t) . Η συνολική ομαδοποίηση πρέπει να αποτελείται από |T| ομάδες αλλά με την εισαγωγή της μεμονωμένης ομάδας έχουμε συνολικά |T|+1 ομάδες. Επομένως, μια νέα ομάδα $Y_{|T|}$ έχει προστεθεί και έτσι το t μπορεί να πάρει μία επιπλέον τιμή. Αυτό έχει ως αποτέλεσμα, οι τιμές των p(t|y) και p(t) να πρέπει να ενημερωθούν σύμφωνα με τις σχέσεις (28) και (29) αφότου το στοιχείο y έχει μετακινηθεί σε μεμονωμένη ομάδα.

Μετά από αυτό το βήμα, ο sequential Information Bottleneck αλγόριθμος συγχωνεύει την μεμονωμένη ομάδα με μία από τις αρχικές ομάδες Y_t ώστε ο συνολικός αριθμός των ομάδων να μειωθεί σε |T|. Όλες οι αρχικές ομάδες Y_t, t ∈ {0,1,…,|T|-1} είναι υποψήφιες για να γίνει η συγχώνευση μαζί τους. Στο σχήμα (9) οι πιθανότητες συγχώνευσης φαίνονται σαν βέλη που συνδέουν την μεμονωμένη ομάδα με τις πιθανές ομάδες στόχους για συγχώνευση. Η επιλογή της ομάδας με την οποία θα γίνει η συγχώνευση γίνεται σύμφωνα με τα λεγόμενα κόστη συγχώνευσης $C(Y_{|T|},Y_t)$, τα οποία πρέπει να υπολογιστούν για όλες τις πιθανές ομάδες στόχους και επιλέγεται τελικά αυτή με το μικρότερο κόστος. Σύμφωνα με τον ορισμό του κόστους[13] πρέπει να διαιρέσουμε την σχέση (25) της Lagrangian με -β<0. Αυτό έχει ως αποτέλεσμα μια τροποποιημένη σχέση για την Lagrangian:

$$L'\{p(t|y)\} = I(T;Y) - \beta^{-1} I(T;X) \quad (32)$$

η οποία θέλουμε να μεγιστοποιηθεί.

Τελικά το κόστος συγχώνευσης υπολογίζεται από την ακόλουθη σχέση:

$$C\left(Y_{|T|}, Y_t\right) = (p(y) + p(t) \cdot JS(p(x|y), p(x|t)) \quad (33)$$

όπου JS(p,q) δηλώνει την Jensen-Shannon απόκλιση που ορίζεται ως:

$$JS(p,q) = \pi_0 D_{KL}(p|\bar{p}) + \pi_1 D_{KL}(q|\bar{p}) \quad (34)$$

Όπου $D_{KL}\{p(x)|q(x)\} = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$

$$\{p,q\} \equiv \{p(x|y), p(x|t) \quad (35)$$

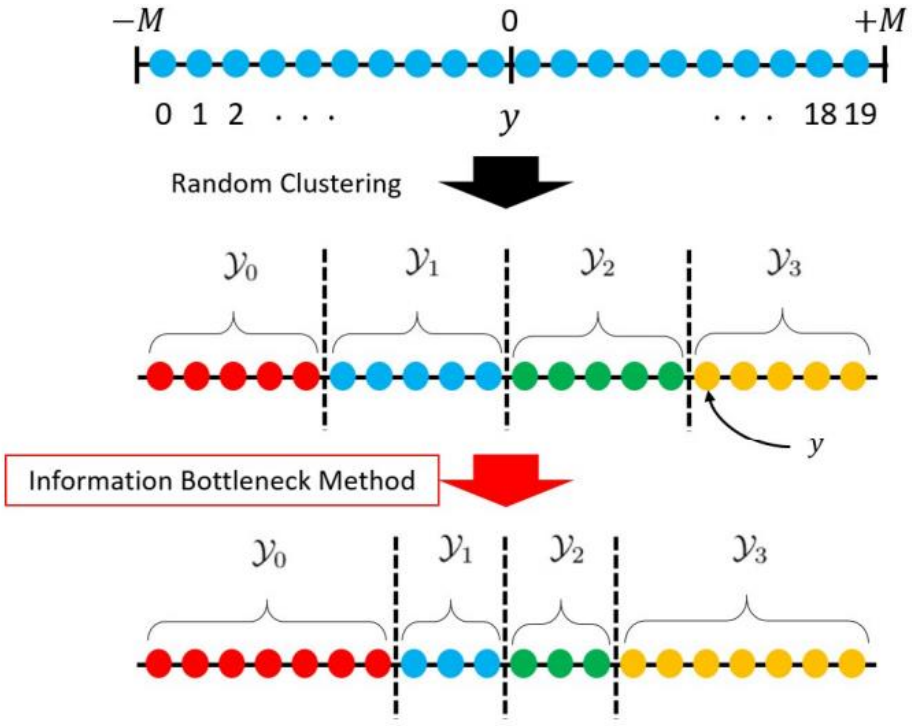
$$\{\pi_0, \pi_1\} \equiv \left\{\frac{p(y)}{p(y)+p(t)}, \frac{p(t)}{p(y)+p(t)}\right\} \quad (36)$$

$$\bar{p} = \pi_0 p(x|y) + \pi_1 p(x|t) \quad (37)$$

Ο αλγόριθμος επαναλαμβάνει τα βήματα για την εξαγωγή και την συγχώνευση για όλα τα y ∈ Y συνεχόμενα (“sequentially”) και σταματάει όταν οι ομάδες δεν αλλάζουν πλέον. Με αυτό τον τρόπο δεν είναι εγγυημένο ότι θα βρεθεί το γενικό μέγιστο της σχέσης (32). Αυτός είναι ο λόγος που ο αλγόριθμος πρέπει να τρέξει για αρκετές διαφορετικές αρχικοποιήσεις των ομάδων. Τελικά επιλέγεται το p(t|y) που αντιστοιχεί στο μεγαλύτερο L’{p(t|y)}.

3.3 *Σχεδιασμός Quantizer*

Ο σχεδιασμός του quantizer είναι βασισμένος στον τροποποιημένο sequential Information Bottleneck αλγόριθμο όπως φαίνεται στο [14]: Algorithm 1. Αρχικά οι έξοδοι του καναλιού είναι διακριτοποιημένες όπως φαίνεται στο σχήμα(10).



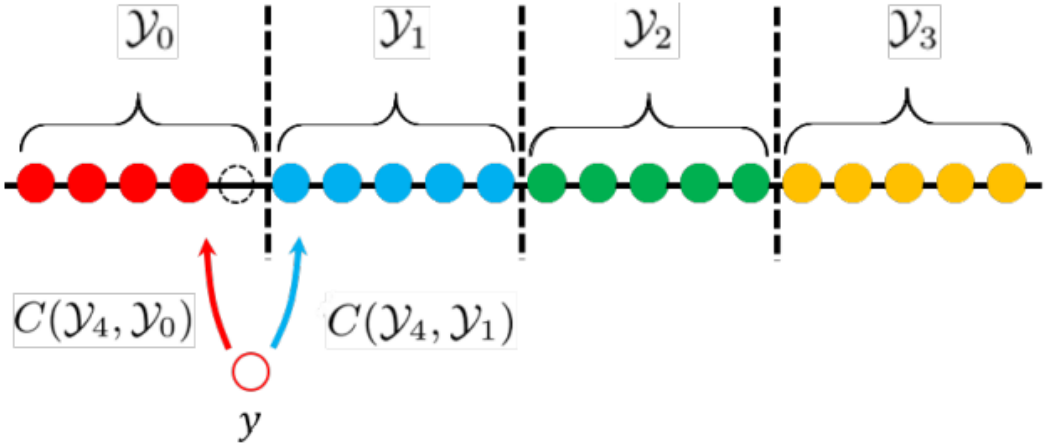
Σχήμα 10: Το quantization των εξόδων του καναλιού. Σε αυτό το παράδειγμα, οι έξοδοι είναι διακριτές σε $|y| = 20$ τιμές και έχουν ομαδοποιηθεί σε $|T| = 4$ ομάδες.

Έστω ένα κωδικό bit x ∈ {0,1} από μία δυαδική κωδική λέξη LDPC(codeword) που μεταδίδεται μέσω ενός συμμετρικού AWGN καναλιού με binary phase shift keying (BPSK) modulation. Τα bit πρώτα αντιστοιχίζονται σε ένα διπολικό σύμβολο διαμόρφωσης s(x) = -2x+1. Το σύμβολο s(x)∈{-1,+1} περνάει από το AWGN κανάλι και η έξοδος του καναλιού (Y) παρατηρείται. Η προηγούμενη κατανομή p(x) θεωρείται πως είναι p(x)=1/2 και για τις δύο πιθανές τιμές του x∈{0,1}.

Στο παράδειγμά μας οι έξοδοι του καναλιού παίρνουν τιμές στο εύρος [-M,+M] και έχουν διακριτοποιηθεί σε |y| = 20 ακέραιες τιμές. Στην συνέχεια αυτές οι τιμές ομαδοποιούνται σε 4 ομάδες. Εδώ, για y μια λειφθήσα τιμή και x ένα μεταδιδόμενο bit, η κοινή πιθανότητα p(x,y) για την περίπτωση ενός AWGN καναλιού δίνεται από τη σχέση:

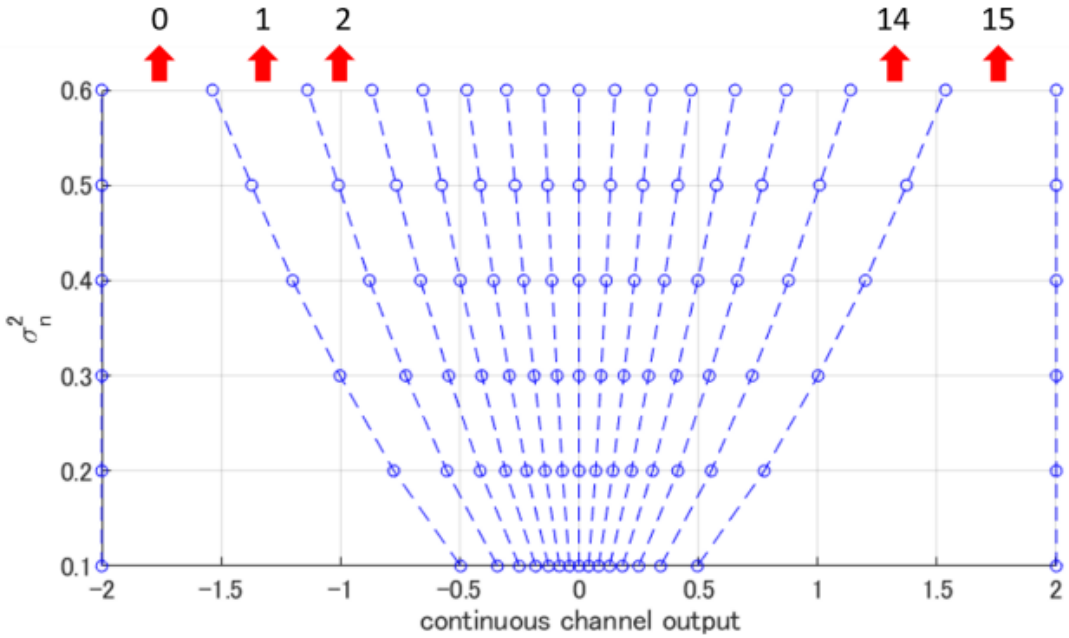
$$p(x,y) = \frac{1}{2\sqrt{2\pi}\sigma} \exp\left(-\frac{|y - s(x)|^2}{2\sigma^2}\right) \qquad (38)$$

όπου σ^2 είναι η διακύμανση θορύβου. Η p(x,y) χρησιμοποιείται ως είσοδος του αλγόριθμου και έτσι το κόστος C υπολογίζεται. Ο τροποποιημένος sequential Information Bottleneck αλγόριθμος διαφέρει από τον πρωτότυπο στο γεγονός ότι λαμβάνει υπόψιν μόνο τις διπλανές ομάδες όταν υπολογίζει το κόστος όπως φαίνεται στο παρακάτω σχήμα(11)



Σχήμα 11: Διαδικασία της τροποποιημένης sequential Information Bottleneck μεθόδου

Ο αλγόριθμος εξάγει και συγχωνεύει τις διακριτοποιημένες εξόδους του καναλιού σύμφωνα με το κόστος C, το οποίο αλλάζει τα όρια της κβαντοποίησης. Επαναλαμβάνοντας αυτή τη διαδικασία μέχρι τα όρια να παραμείνουν ίδια, παίρνουμε τη συνάρτηση χαρτογράφησης p(t|y) όπως φαίνεται στο κάτω μέρος του σχήματος(10). Αυτή η p(t|y) έχει περιοχές κβαντοποίησης που μεγιστοποιούν το I(T;X) για την καρδινότητα T, γιατί θεωρείται $\beta \to \infty$ για το Lagrangian. Στο σχήμα (12) φαίνονται τα όριο κβαντοποίησης που παίρνουμε όταν ο quantizer χρησιμοποιεί την μέθοδο Information Bottleneck. Έχουμε χρησιμοποιήσει M = 2, |T| = 16, |Y| =2000 και έχουμε προσαρμόσει BPSK(Binary Phase Shift Keying) και ένα AWGN κανάλι και έχουμε τρέξει τον αλγόριθμο για 50 διαφορετικές αρχικοποιήσεις ομάδων. Όπως φαίνεται τα αποτελέσματα του καναλιού στην πιο αριστερή περιοχή με αριθμό 0, αυτά στην δεύτερη πιο αριστερά περιοχή στον αριθμό 1 και ούτω καθεξής.



Σχήμα 12: Όρια κβαντοποίησης της εξόδου του καναλιού με τον quantizer που χρησιμοποιεί την μέθοδο Information Bottleneck

Επιπλέον, με την μέθοδο Information Bottleneck υπολογίζεται η εκ των υστέρων πιθανότητα p(x|t) και έτσι μπορούμε να υπολογίσουμε τα LLRs όπως φαίνεται στην παρακάτω σχέση:

$$L = \frac{\Pr\,(x = 0|t)}{\Pr\,(x = 1|t)} \qquad (39)$$

Ο υπολογισμός αυτών των LLRs είναι πολύ σημαντικός για την υλοποίηση του decoder αυτής της εργασίας και θα χρησιμοποιηθούν στο επόμενο κεφάλαιο.

4 Υλοποίηση LDPC Decoder

4.1 Περιγραφή της υλοποίησης

Στην υλοποίηση μας θα προσπαθήσουμε να ελαχιστοποιήσουμε το Bit Error Rate του LDPC decoder της matlab με την χρήση της μεθόδου Information Bottleneck που περιγράψαμε στο προηγούμενο κεφάλαιο.

Αρχικά, βασισμένοι στον αλγόριθμο των Jan Lewandowsky και Gerald Bauch [14] θα βρούμε τα όρια κβαντοποίησης χρησιμοποιώντας M = 2, |T| = 16, |Y| =2000 και έχουμε προσαρμόσει BPSK(Binary Phase Shift Keying) και ένα AWGN κανάλι και τρέχουμε τον αλγόριθμο για 50 διαφορετικές αρχικοποιήσεις ομάδων. Βρίσκουμε αποτελέσματα για τιμές διακύμανσης θορύβου σ^2 = [0.1 ,0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2]. Τρέχουμε το πρόγραμμά μας με την χρήση της γλώσσας προγραμματισμού Python.

Στην συνέχεια αποθηκεύουμε τα αποτελέσματά μας και κρατάμε τις περιοχές κβαντοποίησης, συγκεκριμένα τα όρια των περιοχών καθώς και τις πιθανότητες LLRs που αντιστοιχούν σε κάθε περιοχή και θα είναι η είσοδος για τον LDPC decoder της matlab.

Έπειτα, θα περάσουμε 2000 διαφορετικές εισόδους πρώτα από LDPC encoder. Θα εφαρμόσουμε BPSK(Binary Phase Shift Keying) καθώς και θόρυβο μέσω ενός καναλιού AWGN για τις δώδεκα τιμές διακύμανσης θορύβου σ^2 = [0.1 ,0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2]. Για κάθε μία από αυτές τις τιμές έχουμε ένα σήμα με θόρυβο. Ελέγχουμε λοιπόν τις τιμές του σήματος που έχουν επηρεαστεί από θόρυβο και βρίσκουμε σε ποια από τις περιοχές που έχουμε χωρίσει με τον κβαντοποιητή ανήκει κάθε μία από αυτές. Ανάλογα με την ομάδα στην οποία ανήκει η κάθε τιμή, κρατάμε την τιμή του LLR που αντιστοιχίζεται στην συγκεκριμένη ομάδα όπως έχουμε βρει.

Πλέον είμαστε έτοιμοι να χρησιμοποιήσουμε τον LDPC decoder ο οποίος θα δέχεται σαν είσοδο αυτές τις τιμές των LLR και θα μας δίνει την αποκωδικοποιημένη τιμή του σήματος. Έτσι

μετράμε πόσα ψηφία του τελικού αποτελέσματος συμπίπτουν με αυτά του αρχικού σήματος ώστε να δούμε πόσο αποτελεσματικός είναι ο decoder μετρώντας την τιμή του Bit Error Rate.

Επιπρόσθετα, επιχειρούμε την ίδια διαδικασία όμως χωρίς τον χωρισμό των ομάδων μέσω της μεθόδου του Information Bottleneck. Σε αυτή την περίπτωση απλά χωρίζουμε τις ομάδες διαλέγοντας ως ακραίες τιμές για τα όρια (-1-3σ) ως κατώτατο όριο και (1+3σ) ως ανώτατο όριο και χωρίζοντας τις ομάδες σε ισαπέχουσες περιοχές και ελέγχουμε τα αποτελέσματα.

Την παραπάνω διαδικασία την επαναλαμβάνουμε για πέντε διαφορετικού πίνακες ισοτιμίας (parity check matrix), σύμφωνα με τους οποίους αλλάζει και το μέγεθος των λέξεων που χρησιμοποιούνται σαν είσοδοι και μετράμε τα αποτελέσματα για κάθε πίνακα ισοτιμίας.

4.2 Επιλογή Πινάκων Ισοτιμίας (Parity Check Matrices)

Για την υλοποίησή μας πρέπει να επιλέξουμε πίνακες ισοτιμίας με ρυθμό κώδικα (code rate), όπως φαίνεται και στην σχέση (8), που θα ισούται με 1/2. Για να το πετύχουμε αυτό θα διαλέξουμε πίνακες ισοτιμίας που έχει δημιουργήσει ο David MacKay [1] και είναι διαθέσιμοι στην ιστοσελίδα του[2].

Εμείς επιλέγουμε τους πίνακες

- 96.33.964 (N=96,K=48,M=48,R=0.5) ,
- 252.252.3.252 (N=504,K=252,M=252,R=0.5) ,
- 408.33.844(N=408,K=204,M=204,R=0.5),
- 10000.10000.3.361 (N=20000,K=10000,M=10000,R=0.5)

με βάρος στήλης (αριθμός άσσων σε στήλη) $d_u = 3$ και τον πίνακα

- 204.55.187 (N=204,K=102,M=102,R=0.5)

με βάρος στήλης $d_u = 5$.

4.3 Περιγραφή κώδικα matlab

Παρακάτω φαίνεται ο κώδικας matlab που χρησιμοποιήθηκε για την υλοποίηση του decoding και την εξαγωγή των αποτελεσμάτων:

```
load setsfinal3.mat
load limits3.mat
limits32 = zeros(12, 15);
for k = 1:12
  limits32(k,:) = linspace(-1-3*sqrt(k/10),1+3*sqrt(k/10),15);
end

H1 = struct2array(load ('Hsparse96.33.964.mat'));
H = sparse(H1);
ldpcEnc = comm.LDPCEncoder(H);
ldpcDec = comm.LDPCDecoder(H,'MaximumIterationCount',(10));
ber = comm.ErrorRate;

K=48;
s=zeros(1,K);
totalerrors1=zeros(12,1);
totalerrors2=zeros(12,1);
runs=2000;
for rngc = 1:runs
    rng shuffle
    disp(rngc)
    var_noise = linspace(0.1,1.2,12);
    sn2counter = 0;
    for sn2=var_noise
        sn2counter = sn2counter + 1;
        data = logical(randi([0 1], K, 1));

        encodedData1    = transpose(ldpcEnc(data));
        encodedData    = double(encodedData1);

        c=0;
        for i=encodedData
            c=c+1;
        sn2counter = sn2counter + 1;
        data = logical(randi([0 1], K, 1));

        encodedData1    = transpose(ldpcEnc(data));
        encodedData    = double(encodedData1);

        c=0;
        for i=encodedData
            c=c+1;
            s(c) = real(pskmod(i,2));
        end


        data_with_noise = s + randn(size(s))*sqrt(sn2);
        rxData = bpskDemodulator(transpose(data_with_noise));
        data_with_bpsk = transpose(rxData);

        [index,demodData1(1,:)] = quantiz(data_with_noise,limits3(sn2counter,:),setsfinal3(sn2counter,:));
        [index,demodData2(1,:)] = quantiz(data_with_bpsk,limits2(sn2counter,:),setsfinal2(sn2counter,:));
        demodSignal1 = transpose(demodData1);
        receivedBits1   = ldpcDec(demodSignal1);
        errorStats1     = ber(data, receivedBits1);
        totalerrors1(sn2counter,1)    = totalerrors1(sn2counter,1)  + biterr(data, receivedBits1);
        demodSignal2 = transpose(demodData2);
        receivedBits2   = ldpcDec(demodSignal2);
        errorStats2     = ber(data, receivedBits2);
        totalerrors2(sn2counter,1)    = totalerrors2(sn2counter,1)  + biterr(data, receivedBits2);
    end
end
```

Εικόνα 3: κώδικας matlab

Αρχικά φορτώνουμε τα απαραίτητα όρια και τιμές LLR που έχουμε υπολογίσει και υπολογίζουμε τα όρια για τις ομάδες διαλέγοντας ως ακραίες τιμές (-1-3σ) ως κατώτατο όριο και (1+3σ) ως ανώτατο όριο και χωρίζοντας τις ομάδες σε ισαπέχουσες περιοχές ώστε να κάνουμε την σύγκριση. Έπειτα διαλέγουμε τον πίνακα ισοτιμίας με τον οποίο θα εργαστούμε και

αρχικοποιούμε τον Encoder και τον Decoder έτσι ώστε να χρησιμοποιούν τον πίνακα ισοτιμίας που θέλουμε και διαλέγουμε στις πόσες επαναλήψεις θα τερματίζει ο decoder.

Στην συνέχεια για 2000 διαφορετικές τυχαία παραγόμενες εισόδους, το μέγεθος των οποίων εξαρτάται από τον πίνακα ισοτιμίας που έχουμε επιλέξει, τρέχουμε τον decoder για κάθε μία από τις διαφορετικές τιμές της διακύμανσης θορύβου σ^2 = [0.1 ,0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2]. Πρώτα περνάμε τις εισόδους από τον LDPC encoder και εφαρμόζουμε στην έξοδό του BPSK(Binary Phase Shift Keying). Έπειτα εφαρμόζουμε θόρυβο AWGN καναλιού στα δεδομένα. Είμαστε έτοιμοι να χρησιμοποιήσουμε τον quantizer με τις τιμές για τα όρια που έχουμε βρει με την μέθοδο Information Bottleneck. Μετά τον υπολογισμό της bpsk αποδιαμώρφοσης των δεδομένων με θόρυβο, που στην περίπτωσή μας είναι πολλαπλασιασμός επί 4, παίρνουμε τις τιμές LLR για τα δεδομένα και μπορούμε να τις χρησιμοποιήσουμε για να αντιστοιχίσουμε σε ομάδες με ακραίες τιμές (-1-3σ) ως κατώτατο όριο και (1+3σ) ως ανώτατο όριο. Αφού βρούμε σε ποια ομάδα από τις 16 αναλογούν τα δεδομένα με θόρυβο, το LLR που αντιστοιχίζεται στην συγκεκριμένη ομάδα είναι η μέση τιμή των ορίων αυτής της ομάδας. Φτάνουμε στο σημείο όπου χρησιμοποιούμε τον LDPC decoder και για τις δύο περιπτώσεις και μετράμε τα λάθη (totalerrors), δηλαδή το πόσα bit από την έξοδο του decoder (demodSignal) είναι διαφορετικά από τα bit της αρχικής τιμής των δεδομένων (data) για κάθε τιμή της διακύμανσης θορύβου.

Αφού τρέξουμε το πρόγραμμα για κάθε έναν από τους πίνακες ισοτιμίας που αναφέραμε παραπάνω και για αριθμό μέγιστων επαναλήψεων του decoder 5 και 10 είμαστε έτοιμοι να υπολογίσουμε το Bit Error Rate σε σχέση με το E_b/N_0 που στην περίπτωσή μας ισούται με το SNR=$1/\sigma^2$. Περνάμε λοιπόν στα αποτελέσματά μας.

5 Αποτελέσματα υλοποίησης

5.1 *Τρόπος υπολογισμού των αποτελεσμάτων*

Ο στόχος μας είναι να ελέγξουμε αν η διαδικασία της αποκωδικοποίησης με την χρήση κβαντισμού με την μεθόδου Information Bottleneck είναι πιο αποτελεσματική από την χρήση ενός συμβατικό κβαντισμού που περιγράψαμε στο Κεφάλαιο 4.

Για να κάνουμε αυτό τον έλεγχο πρέπει να υπολογίσουμε το Bit Error Rate σε σχέση με το E_b/N_0 που στην περίπτωσή μας ισούται με το SNR=$1/\sigma^2$. Πιο συγκεκριμένα ο αριθμός των Bit Errors είναι ο αριθμός των ληφθέντων δυαδικών ψηφίων μιας ροής δεδομένων μέσω ενός καναλιού επικοινωνίας που έχουν αλλάξει λόγω θορύβου, παρεμβολών, παραμορφώσεων ή σφαλμάτων συγχρονισμού. Το Bit Error Rate (BER) είναι ο αριθμός των εσφαλμένων bit ανά μονάδα χρόνου. Το E_b/N_0(db) ισούται με

$$\frac{E_b}{N_0}(dB) = 10\log_{10}\frac{T_{sym}}{T_{samp}} + SNR(dB) \qquad (40)$$

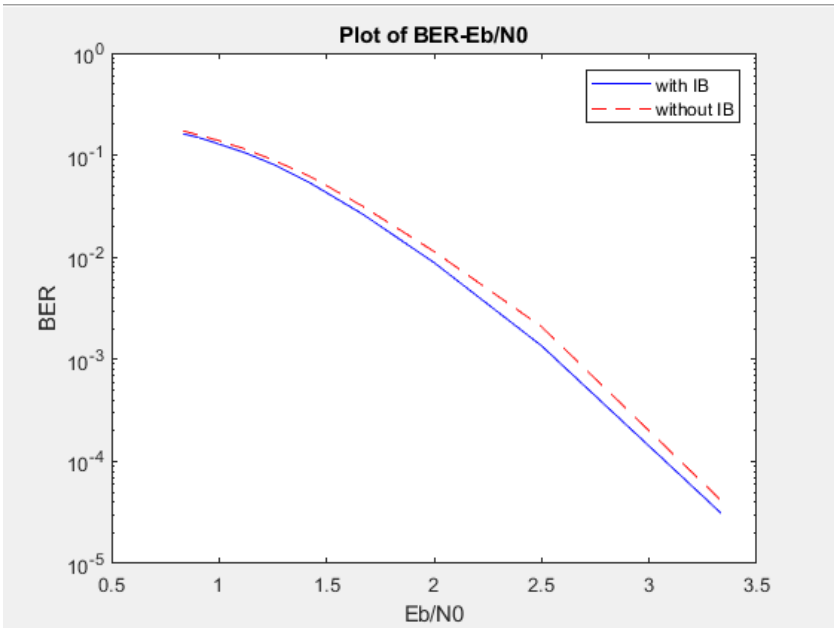
όπου T_{sym} είναι η περίοδος συμβόλων του σήματος και T_{samp} είναι η περίοδος δειγματοληψίας του σήματος [3]. Στην περίπτωσή μας T_{sym} = 1 και T_{samp} = 1 οπότε σύμφωνα με την σχέση (40) καταλήγουμε ότι:

$\frac{E_b}{N_0}(dB) = SNR(dB) = \frac{1}{\sigma^2}$.

5.2 *Παρουσίαση αποτελεσμάτων*

Αφού υπολογίσουμε αυτές τις τιμές για όλες μας τις υλοποιήσεις, δηλαδή, για κάθε πίνακα ισοτιμίας και με αριθμό διαφορετικών εισόδων 2000 για 5 και 10 επαναλήψεις ως ανώτατο όριο επαναλήψεων του LDPC decoder, καταλήγουμε στα παρακάτω αποτελέσματα:

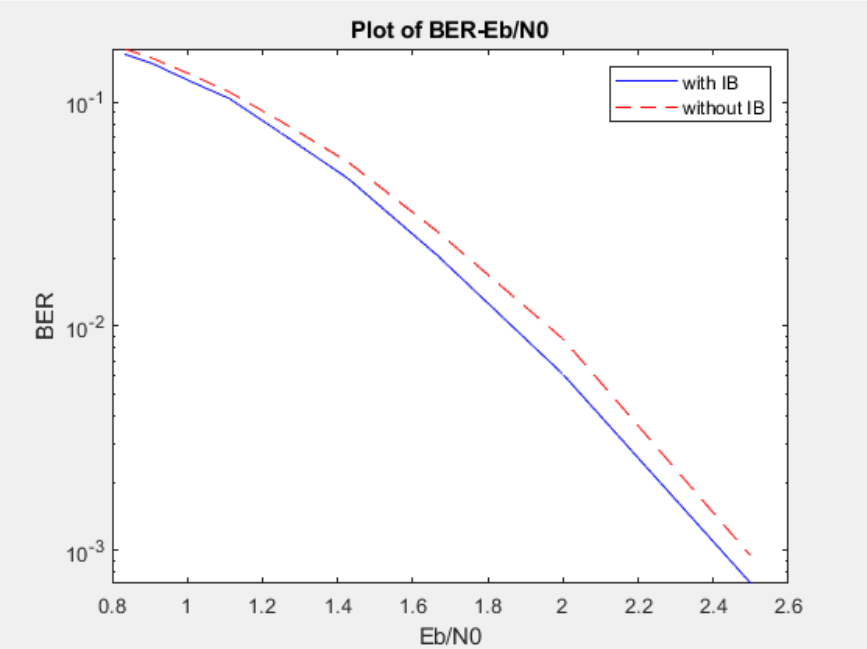
5.2.1 Πίνακας ισοτιμίας H96.33.964 για 5 και 10 επαναλήψεις



Σχήμα 13: H96.33.964 επαναλήψεις: 5

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	15575	16609	0.162240	0.173010
0.9	14181	15045	0.147719	0.156719
1.0	12357	13301	0.128719	0.138552
1.1	10318	11206	0.107479	0.116729
1.3	7842	8652	0.081687	0.090125
1.4	5060	5871	0.052708	0.061156
1.7	2545	3043	0.026510	0.031698
2.0	850	1093	0.008854	0.011385
2.5	130	200	0.001354	0.002083

3.3	3	4	0.000031	0.000042
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

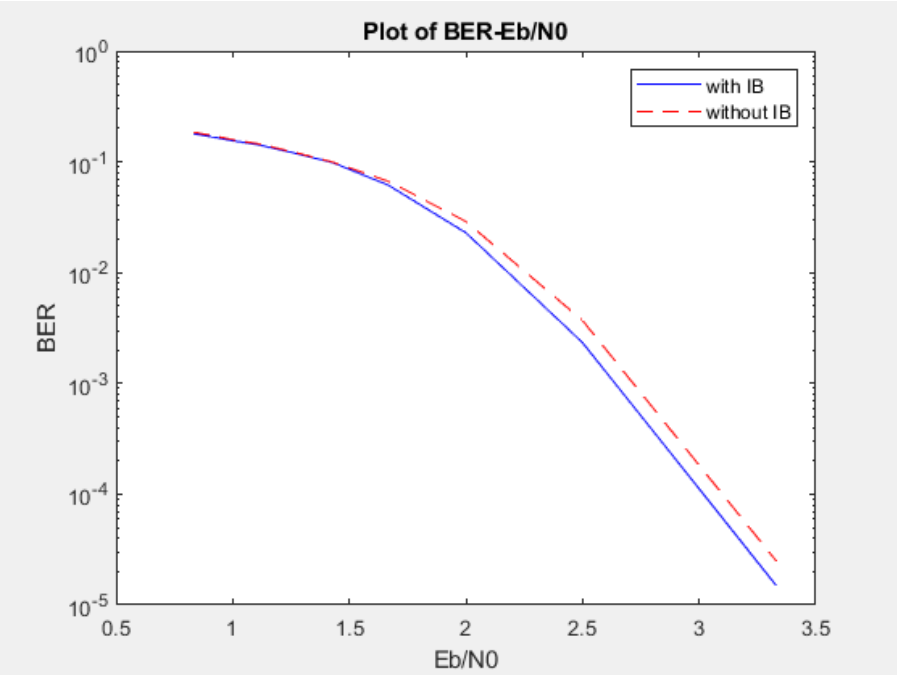


Σχήμα 14: H96.33.964 επαναλήψεις: 10

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	15788	16673	0.164458	0.173677
0.9	14272	15096	0.148667	0.157250
1.0	12109	12990	0.12613	0.135313
1.1	10011	10723	0.104281	0.111698

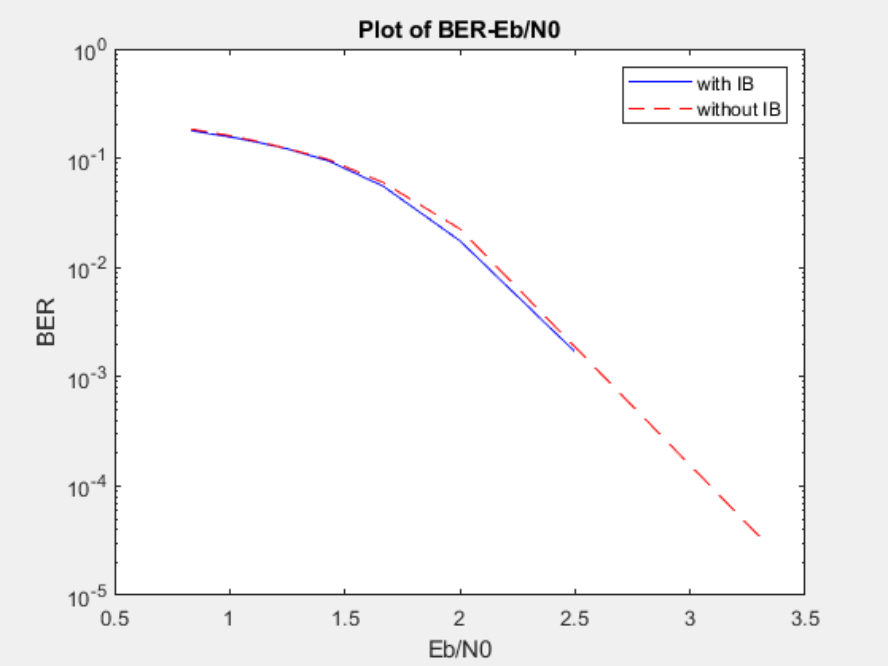
1.3	7001	7885	0.072927	0.082135
1.4	4382	5190	0.045646	0.054062
1.7	1981	2535	0.020635	0.026406
2.0	585	842	0.006094	0.008771
2.5	68	90	0.000708	0.000937
3.3	0	0	0.000000	0.000000
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

5.2.2 Πίνακας ισοτιμίας H204.55.187 για 5 και 10 επαναλήψεις



Σχήμα 15: H204.55.187 επαναλήψεις: 5

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	36532	37712	0.179078	0.184863
0.9	34316	35312	0.168216	0.173098
1.0	31636	32316	0.155078	0.158412
1.1	29045	29635	0.142377	0.145270
1.3	24869	25309	0.121907	0.124064
1.4	20087	20298	0.098466	0.099500
1.7	12575	13642	0.061642	0.066873
2.0	4663	5899	0.022858	0.028917
2.5	476	754	0.002333	0.003696
3.3	3	5	0.000015	0.000025
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

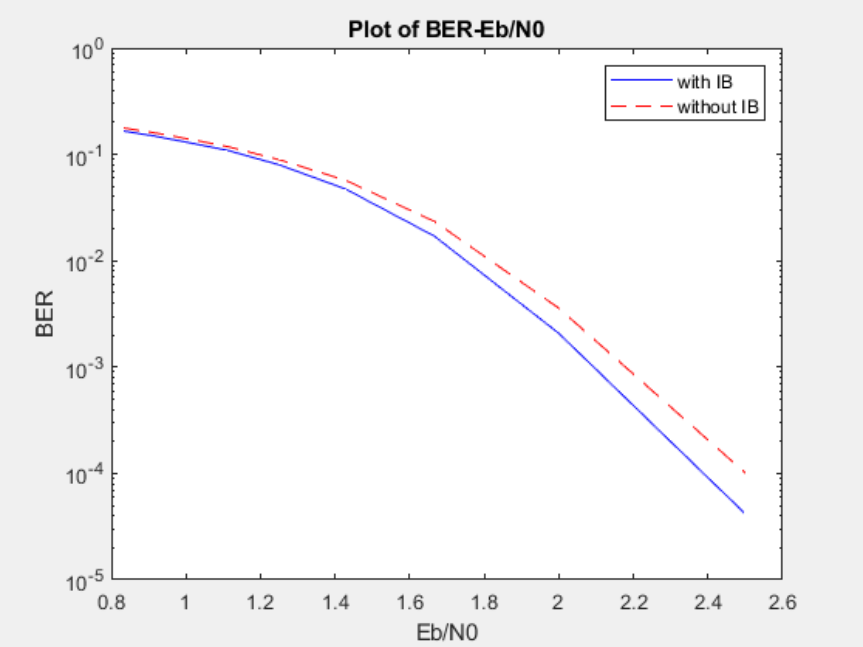


Σχήμα 16: H204.55.187 επαναλήψεις: 10

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	36625	37701	0.179534	0.184809
0.9	34403	35376	0.168642	0.173412
1.0	32014	32831	0.156931	0.160936
1.1	28860	29364	0.141471	0.143941
1.3	24865	25067	0.121887	0.122877
1.4	19305	19855	0.094632	0.097328
1.7	11315	12214	0.055466	0.059873
2.0	3580	4604	0.017549	0.022569
2.5	345	382	0.001691	0.001873

3.3	0	6	0.000000	0.000029
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

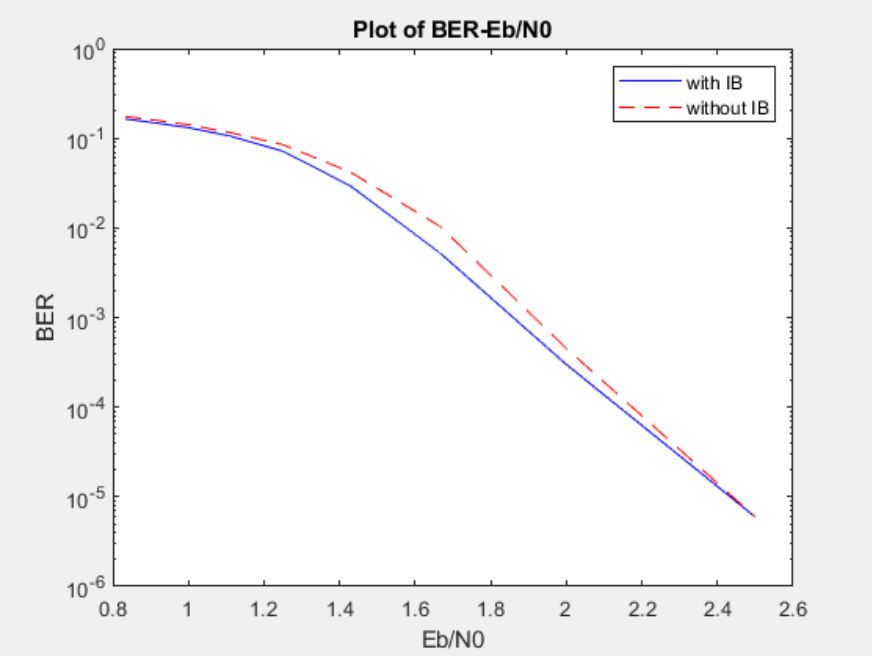
5.2.3 Πίνακας ισοτιμίας H252.252.3.252 για 5 και 10 επαναλήψεις



Σχήμα 17: H252.252.3.252 επαναλήψεις: 5

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	83333	89219	0.165343	0.177022
0.9	75723	81160	0.150244	0.161032
1.0	65758	71056	0.130472	0.140984
1.1	55058	59765	0.109242	0.118581

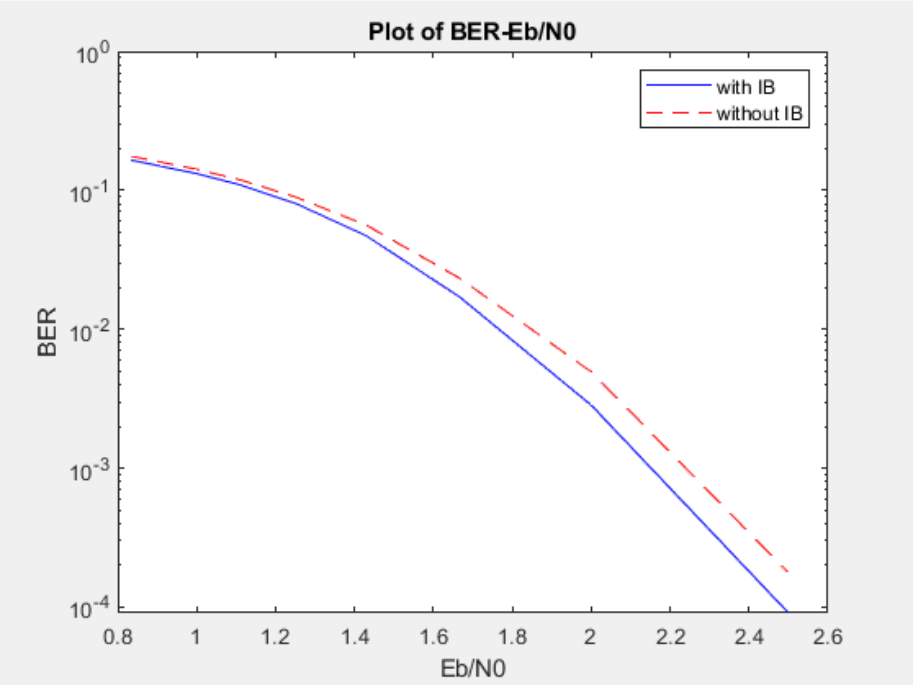
1.3	40220	44930	0.079802	0.089147
1.4	23775	28723	0.047173	0.056990
1.7	8561	11781	0.016986	0.023375
2.0	1046	1802	0.002075	0.003575
2.5	21	51	0.000042	0.000101
3.3	0	0	0.000000	0.000000
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000



Σχήμα 18: H252.252.3.252 επαναλήψεις: 10

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	82663	88246	0.164014	0.175091
0.9	74936	80420	0.148683	0.159563
1.0	66206	71285	0.131361	0.141438
1.1	53145	58192	0.105446	0.115460
1.3	36142	42741	0.071710	0.084804
1.4	14832	21146	0.029429	0.041956
1.7	2626	5210	0.005210	0.010337
2.0	151	228	0.000300	0.000452
2.5	3	3	0.000006	0.000006
3.3	0	0	0.000000	0.000000
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

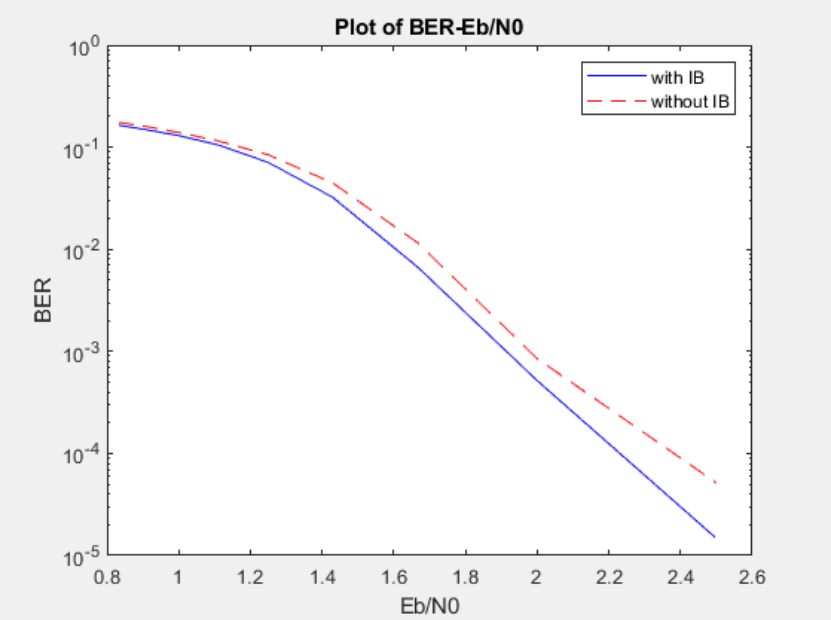
5.2.4 Πίνακας ισοτιμίας H408.33.844 για 5 και 10 επαναλήψεις



Σχήμα 19: H408.33.844 επαναλήψεις: 5

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	67185	71839	0.164669	0.176076
0.9	60821	65091	0.149071	0.159537
1.0	53764	57848	0.131775	0.141784
1.1	44560	48423	0.109216	0.118684
1.3	32843	36500	0.080498	0.089461
1.4	19288	22992	0.047275	0.056353
1.7	6956	9488	0.017049	0.023255
2.0	1159	2012	0.002841	0.004931

2.5	37	72	0.000091	0.000176
3.3	0	0	0.000000	0.000000
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

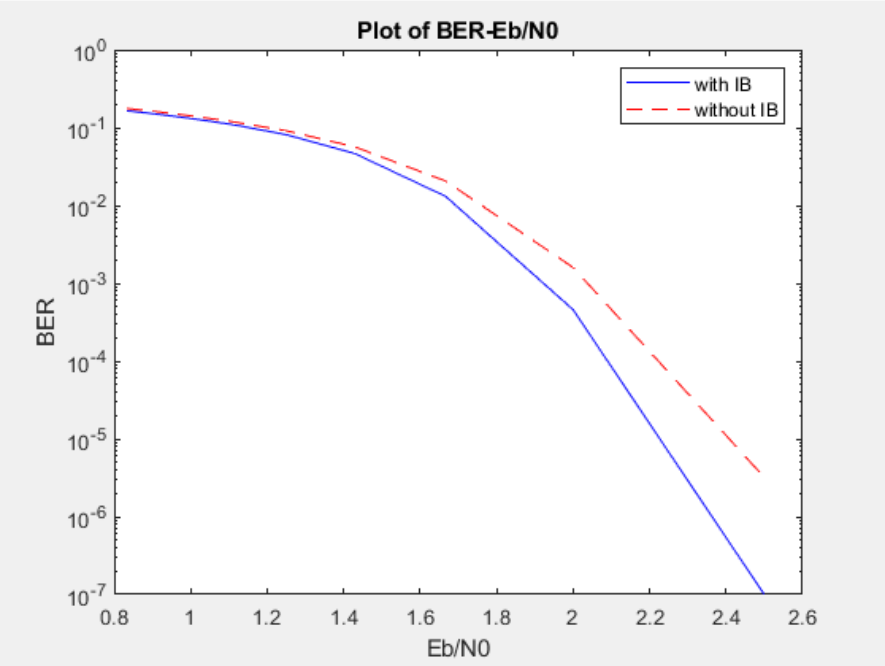


Σχήμα 20: H408.33.844 επαναλήψεις: 10

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	66989	71324	0.164189	0.174814

0.9	60557	65082	0.148424	0.159515
1.0	53038	56832	0.129995	0.139294
1.1	42834	46905	0.104985	0.114963
1.3	28901	34336	0.070836	0.084157
1.4	13258	18302	0.032495	0.044858
1.7	2728	4722	0.006686	0.011574
2.0	211	346	0.000517	0.000848
2.5	6	21	0.000015	0.000051
3.3	0	0	0.000000	0.000000
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

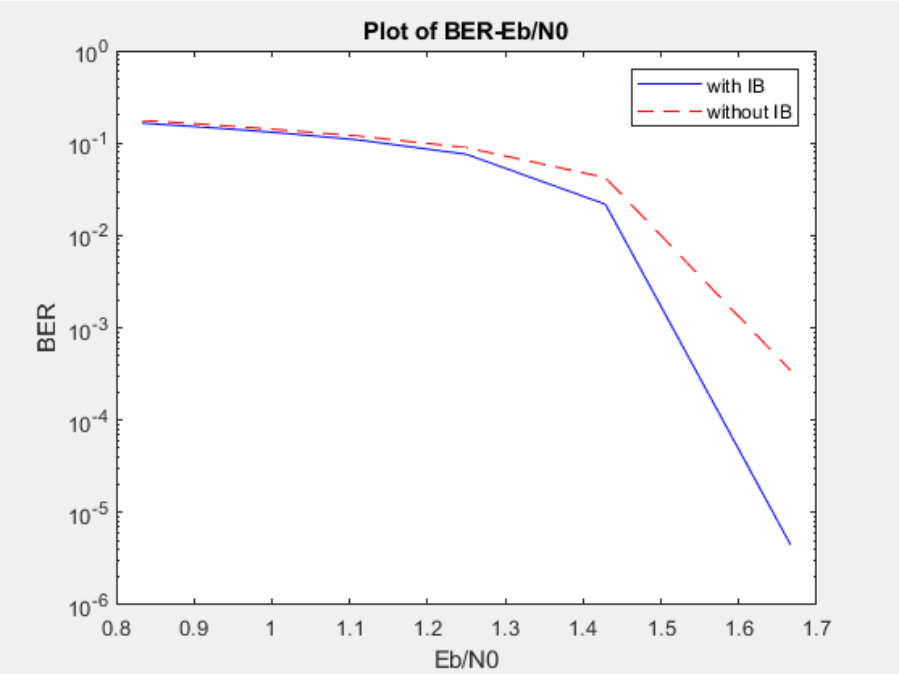
5.2.5 Πίνακας ισοτιμίας H10000.10000.3.631 για 5 και 10 επαναλήψεις



Σχήμα 21: H10000.10000.3.631 επαναλήψεις: 5

SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	3291366	3524795	0.164568	0.176240
0.9	2991601	3211635	0.149580	0.160582
1.0	2628803	2833115	0.131440	0.141656
1.1	2184326	2380357	0.109216	0.119018
1.3	1622511	1821386	0.081126	0.091069
1.4	931007	1134992	0.032495	0.056750
1.7	261561	411722	0.013078	0.020586
2.0	9153	31864	0.000458	0.001593

2.5	2	64	0.000000	0.000003
3.3	0	0	0.000000	0.000000
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000



Σχήμα 22: H10000.10000.3.631 επαναλήψεις: 10

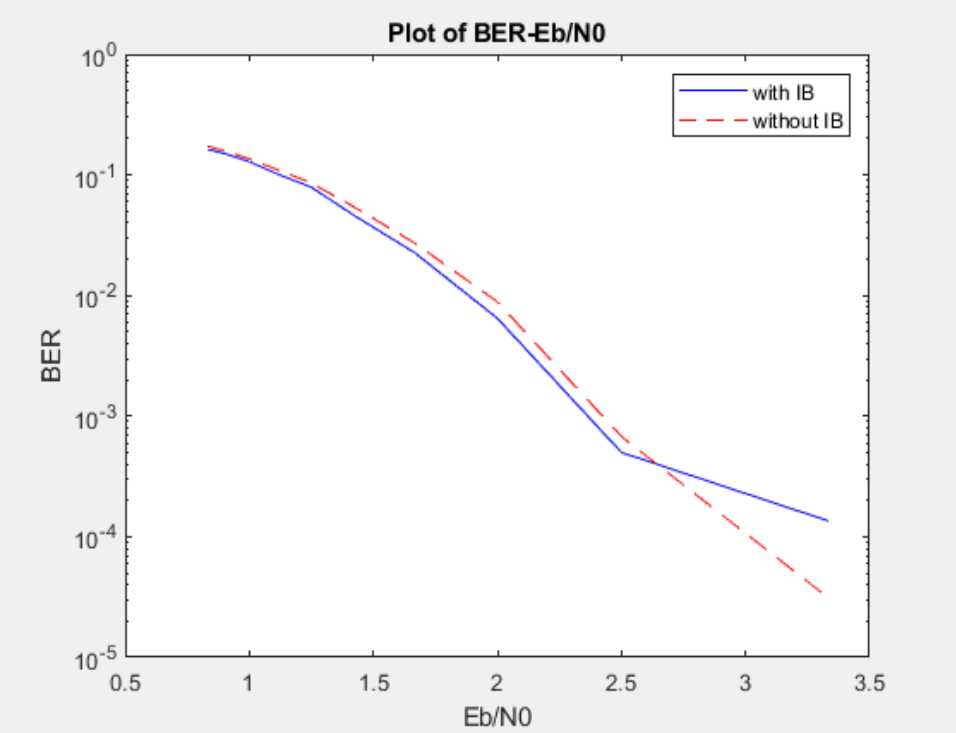
SNR	Errors με IB	Errors χωρίς IB	BER με IB	BER χωρίς IB
0.8	3289174	3523703	0.164459	0.176185
0.9	2987962	3208066	0.149398	0.160403
1.0	2622715	2830435	0.131136	0.141522

1.1	2163219	2379120	0.108161	0.118956
1.3	1511224	1784567	0.075561	0.089228
1.4	434929	844697	0.021746	0.042235
1.7	89	6870	0.000004	0.000344
2.0	0	0	0.000000	0.000000
2.5	0	0	0.000000	0.000000
3.3	0	0	0.000000	0.000000
5.0	0	0	0.000000	0.000000
10.0	0	0	0.000000	0.000000

5.3 Χρήση γραμμικού κβαντιστή με παραπάνω bit κβάντισης

Ο γραμμικός κβαντιστής που χρησιμοποιήσαμε μέχρι τώρα ήταν για 4 bit κβάντισης, δηλαδή χώριζε σε 16 ομάδες από 0000 μέχρι 1111. Επιχειρήσαμε να χρησιμοποιήσουμε γραμμικό κβαντιστή με περισσότερα bit. Και στην περίπτωση με περισσότερα bit χρησιμοποιούμε ισαπέχοντα όρια για την ομαδοποίηση με ακραίες τιμές (-1-3σ) ως κατώτατο όριο και (1+3σ) ως ανώτατο όριο, όπου σ η διακύμανση θορύβου, και επιλέγουμε τιμή LLR για κάθε ομάδα τη μέση τιμή των ορίων της ομάδας.

Μετά την δοκιμή για γραμμικούς κβαντιστές με περισσότερα bit, παρατηρήσαμε ότι σε σχέση με την υλοποίηση με τον κβαντιστή που χρησιμοποιεί την μέθοδο Information Bottleneck ο γραμμικός αρχίζει να έχει καλύτερα αποτελέσματα στην περίπτωση για 12 bit κβάντισης για μεγαλύτερες τιμές του E_b/N_0 όπως φαίνεται και στο σχήμα(23). Για περισσότερα bit κβάντισης υπάρχει βελτίωση, αλλά είναι μηδαμινή μπροστά στην υπολογιστική πολυπλοκότητα που προσθέτει στο κύκλωμα.

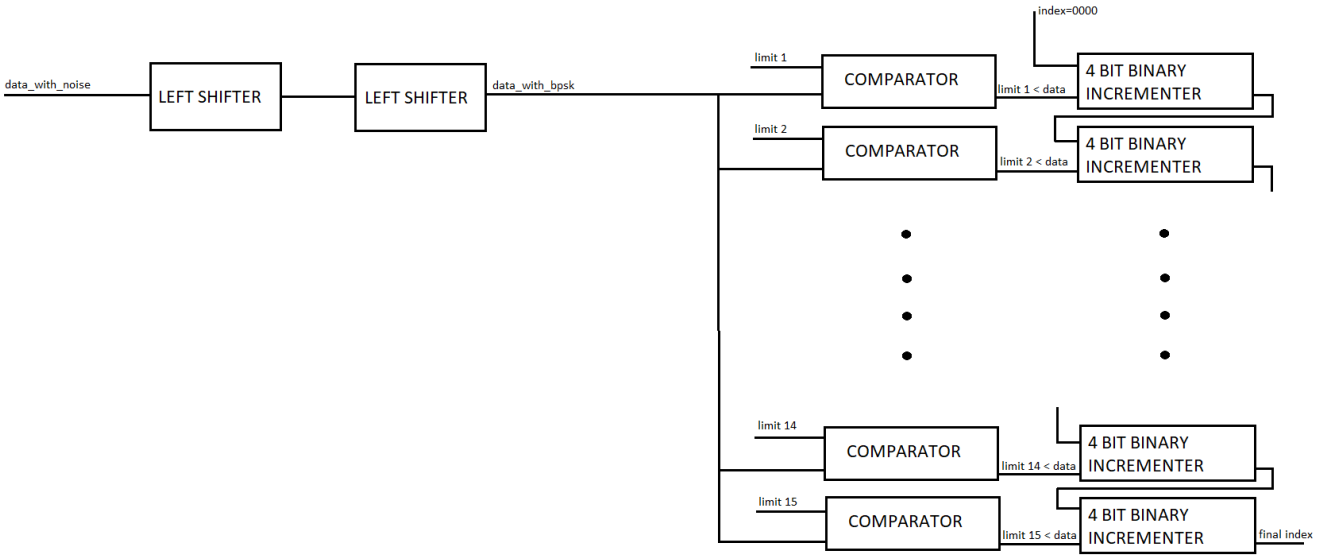


Σχήμα 23: H96.33.964 10 επαναλήψεις, κβαντιστής με IB 4 bit σε σχέση με γραμμικό κβαντιστή 12 bit

5.4 *Υλοποίηση του Κβαντιστή σε υλικό(hardware)*

Για τον quantizer παρουσιάζεται η περιγραφή σε υλικό στο σχήμα(24). Ελέγχουμε κάθε φορά αν η τιμή των δεδομένων είναι μεγαλύτερη από το κατώτατο όριο μέχρι το μεγαλύτερο με έναν comparator και κάθε φορά προσθέτουμε το αποτέλεσμα για να πάρουμε την τιμή index με την οποία θα επιλέξουμε σε ποια ομάδα ανήκει ώστε να δώσουμε καινούργια τιμή. Δηλαδή αν limit<data τότε αυξάνεται κατά 1 το index. Το index παίρνει τιμές από 0000 μέχρι 1111 μία για κάθε ομάδα. Στην συνέχεια η τιμή finalindex θα χρησιμοποιηθεί για την επιλογή της τιμής LLR που θα ανατεθεί ανάλογα με την ομάδα στην οποία αντιστοιχεί.

Για την γραμμική υλοποίηση το παραπάνω που έχουμε είναι οι δύο shifters που χρειάζονται για τον υπολογισμό της bpsk αποδιαμόρφωσης των δεδομένων με θόρυβο. Επιπλέον, αν βάλουμε περισσότερα bit στον γραμμικό quantizer αυξάνεται και ο αριθμός των bit για την τιμή index, άρα ένας half adder παραπάνω σε κάθε incrementer.



Σχήμα 24: Αναπαράσταση του κβαντιστή σε υλικό

5.5 *Συμπεράσματα – Παρατηρήσεις*

- Είδαμε ότι με την χρήση της μεθόδου information bottleneck μπορούμε να πετύχουμε καλύτερα αποτελέσματα αποκωδικοποίησης και μάλιστα με τη χρήση quantizer με πολύ λιγότερα bit από ότι με γραμμικό.
- Παρατηρήσαμε επίσης ότι με τη χρήση μεγαλύτερου πίνακα ισοτιμίας που έχει και ως αποτέλεσμα μεγαλύτερες κωδικές λέξεις έχουμε καλύτερη απόδοση του decoder σε όλες τις τιμές διακύμανσης θορύβου.
- Είδαμε ότι με περισσότερες επαναλήψεις του decoder, έχουμε ελάχιστα καλύτερα αποτελέσματα, που είναι λογικό.
- Η διαδικασία κωδικοποίησης με χρήση της μεθόδου information bottleneck διευκολύνει την υλοποίηση σε hardware καθώς απαιτούνται λιγότερα στοιχεία.

6 Βιβλιογραφία

[1] D. MacKay and R. Neal. Near Shannon Limit Performance of Low-Density Parity-Check Codes. Electronics Letters, 32(18):1645-1646, August 1996.

[2] http://www.inference.org.uk/mackay/codes/

[3] https://www.mathworks.com/help/comm/ug/awgn-channel.html

[4] C.Berrou, A.Glavieux, and P.Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. International Conference on Communication, May 23-26 1993.

[5] D.J.C. Mackay and R.M. Neal. Good Codes Based on Very Sparse Matrices. 5th IMA Conference on Cryptography and Coding, Berlin, Germany:Springer, 1995.

[6] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo: CA : Morgan Kaufmann, 1998.

[7] M.Sipser and D.A.Spielman. Expander Codes. IEEE Transactions on Information Theory, 42:1710–1722, November 1996.

[8] R.Tanner. A recursive approach to low complexity codes. IEEE Transactions on Information Theory, 27:533–547, September 1981.

[9] F.R.Kschischang and B.J. Frey. Iterative Decoding of Compound Codes by Probability Propagation in Graphical Models. Journal on Selected Areas in Communications, 16:219–230, 1998.

[10] N. Tishby, F. C. Pereira, and W. Bialek, ‘‘The information bottleneck method,’’ in Proc. 37th Allerton Conf. Commun. Comput., Monticello, VA, USA, 1999, pp. 368–377.

[11] N. Slonim, N. Friedman, and N. Tishby, ‘‘Unsupervised document classification using sequential information maximization,’’ in Proc. 25th ACM SIGIR, Tampere, Finland, 2002, pp. 129–136.

[12] N. Slonim, ‘‘The information bottleneck: Theory and applications,’’ Ph.D. dissertation, Interdiscipl. Center Neural Comput. (ICNC), Hebrew Univ. Jerusalem, Jerusalem, Israel, 2002

[13] R. Blahut, ‘‘Computation of channel capacity and rate-distortion functions,’’ IEEE Trans. Inf. Theory, vol. IT-18, no. 4, pp. 460–473, Jul. 1972

[14] Jan Lewandowsky, Gerhard Bauch, Information-Optimum LDPC Decoders Based on the Information Bottleneck Method, January 2018.

[15] Rolando Antonio Carrasco, Martin Johnston, Non-Binary Error Control Coding for Wireless Communication and Data Storage, pp.208-216, May 2005