ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Μελέτη και Προσομοίωση Τεχνικών Κωδικοποίησης Καναλιού για Συστήματα Ασυρμάτων Επικοινωνιών Νέας Γενιάς

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος – Ελευθέριος Χ. Λαμπρινάκος

Επιβλέπων : Φίλιππος Κωνσταντίνου
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2007

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Μελέτη και Προσομοίωση Τεχνικών Κωδικοποίησης Καναλιού για Συστήματα Ασυρμάτων Επικοινωνιών Νέας Γενιάς

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος – Ελευθέριος Χ. Λαμπρινάκος

Επιβλέπων : Φίλιππος Κωνσταντίνου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

.....	.....	.....
Φ. Κωνσταντίνου	Χ. Καψάλης	Ν. Ουζούνογλου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2007

.....
Γεώργιος – Ελευθέριος Χ. Λαμπρινάκος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου Φ. Κωνσταντίνου για τη δυνατότητα που μου έδωσε να ασχοληθώ με αυτόν τον πραγματικά ενδιαφέροντα τομέα των τηλεπικοινωνιών. Επίσης, ευχαριστώ ιδιαίτερα τον υποψήφιο διδάκτορα κ. Α. Γκότση για την αποδοτική και ευχάριστη συνεργασία μας. Τέλος, θα ήταν παράλειψη να μην ευχαριστήσω το οικογενειακό και φιλικό μου περιβάλλον που στάθηκε πολύτιμος αρωγός κατά την εκπόνηση αυτής της κοπιαστικής αλλά δημιουργικής εργασίας.

Γ. Λαμπρινάκος

Οκτώβριος 2007

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας είναι η μελέτη των LDPC (Low-Density parity-check) κωδίκων και η ανάπτυξη προσομοιωτή για την εκτίμηση της επίδοσης τους. Έτσι, γίνεται αρχικά μια γενική παρουσίαση των τεχνικών κωδικοποίησης καναλιού καθώς και της μεθοδολογίας της προσομοίωσης. Στη συνέχεια, αναλύουμε τη διαδικασία μοντελοποίησης του τηλεπικοινωνιακού συστήματος που θα προσομοιωθεί κάνοντας εκτενή ανάλυση στο κομμάτι της LDPC κωδικοποίησης και αποκωδικοποίησης. Παράλληλα, δίνουμε ιδιαίτερη έμφαση στις τεχνικές διαμόρφωσης και στον τύπο του τηλεπικοινωνιακού διαύλου. Ο κύριος στόχος της εργασίας επιτυγχάνεται στο κεφάλαιο 5, όπου παρουσιάζονται τα αποτελέσματα της προσομοίωσης μας. Με βάση αυτά τα αποτελέσματα, εξάγουμε χρήσιμα συμπεράσματα για τους LDPC κώδικες ενώ μελετάται σε βάθος η επίδοση τους συναρτήσει των διάφορων τεχνικών διαμόρφωσης και των διάφορων τύπων διαύλου. Επιπλέον, γίνεται αναφορά στις εφαρμογές των LDPC κωδίκων σε ασύρματα συστήματα επικοινωνιών νέας γενιάς και καταδεικνύεται γενικότερα η σημασία τους στις σύγχρονες τηλεπικοινωνίες. Τέλος, γίνεται σύνοψη της συμβολής αυτής της εργασίας και προτείνονται κατευθύνσεις για μελλοντική επέκταση της. Είναι άλλωστε κοινά αποδεκτό ότι οι LDPC κώδικες αφορούν και το παρόν αλλά και το μέλλον των τηλεπικοινωνιών.

Λέξεις – κλειδιά: LDPC, Κώδικας καναλιού, Κινητές Επικοινωνίες, Δορυφορικές Επικοινωνίες, BER, Προσομοίωση, Matlab

Abstract

The aim of this thesis is to study and evaluate by simulation the perfomance of Low-Density parity-check (LDPC) codes. First of all, we present in general the channel coding techniques and we examine the procedure of simulation. In addition, we analyze the modelization of the communication system that it will be simulated. From the parts of this communication system, an extended reference is made to LDPC codes and their decoding algorithms. We also emphasize the different modulation techniques and the different channel types. Our main goal is achieved in chapter 5, where we present the simulation results. Based on these results, we make useful conclusions about characteristics of LDPC codes and we study in depth how their performance is influenced by the modulation schema that is used and by the channel type. Moreover, we mention the applications of LDPC codes in next generation wireles communications systems and we explain why they are attracting much attention these days. Finally, we summarize the contribution of this thesis and we offer suggestions on the expansion of this project and on future research.

Key words: LDPC, Channel coding, Mobile Communications, Satellite Communications, BER, Simulation, Matlab

Πίνακας Περιεχομένων

Ευρετήριο Σχημάτων

1 Εισαγωγή

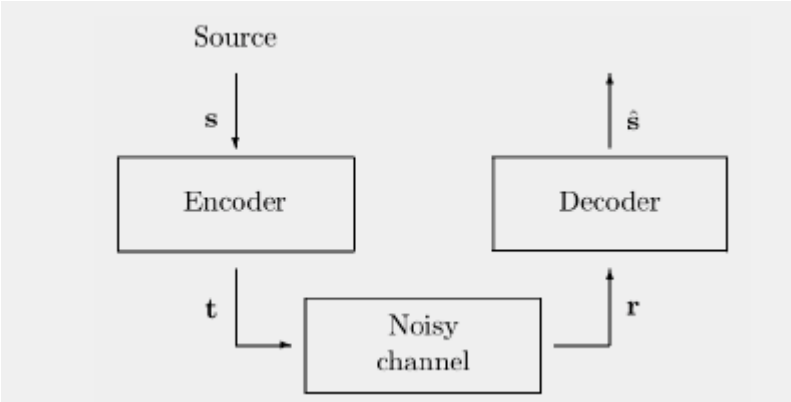
1.1 Γενική Περιγραφή

Με τον όρο κωδικοποίηση καναλιού (channel coding) αναφερόμαστε στη διαδικασία κατά την οποία το αρχικό σήμα πληροφορίας μετασχηματίζεται σε κωδικοποιημένο σήμα, με στόχο να αντιμετωπιστούν καλύτερα οι διάφορες επιβαρύνσεις που εισάγει ο τηλεπικοινωνιακός δίαυλος, όπως θόρυβος, παρεμβολές και διαλείψεις. Αδιαμφισβήτητα, οι επιβαρύνσεις αυτές είναι εντονότερες σε ασύρματα συστήματα επικοινωνιών. Η προαναφερθείσα διαδικασία κωδικοποίησης λαμβάνει χώρα στον πομπό του τηλεπικοινωνιακού συστήματος και ακολουθείται από την αντίστοιχη διαδικασία αποκωδικοποίησης που λαμβάνει χώρα στο δέκτη.

Η κωδικοποίηση καναλιού απαντάται μόνο σε ψηφιακά τηλεπικοινωνιακά συστήματα και είναι ένας από τους λόγους υπεροχής των ψηφιακών έναντι των αναλογικών συστημάτων, καθώς μπορεί να προσφέρει αυξημένη ποιότητα στο τηλεπικοινωνιακό σύστημα. Αυτή η αυξημένη ποιότητα έχει και το κόστος της, το οποίο μπορεί να είναι για παράδειγμα αύξηση του απαιτούμενου εύρους ζώνης συχνοτήτων. Συγκρίνοντας όμως την κωδικοποίηση καναλιού με άλλες τεχνικές βελτίωσης της ποιότητας του συστήματος, όπως χρήση πομποδεκτών μεγαλύτερης ισχύος, διαπιστώνεται ότι η κωδικοποίηση είναι από τις πλέον αποδοτικές και οικονομικές λύσεις, κυρίως χάρη στην εξέλιξη στον τομέα των ολοκληρωμένων κυκλωμάτων μεγάλης κλίμακας (LSI) και στις τεχνικές επεξεργασίας ψηφιακών σημάτων (DSP).

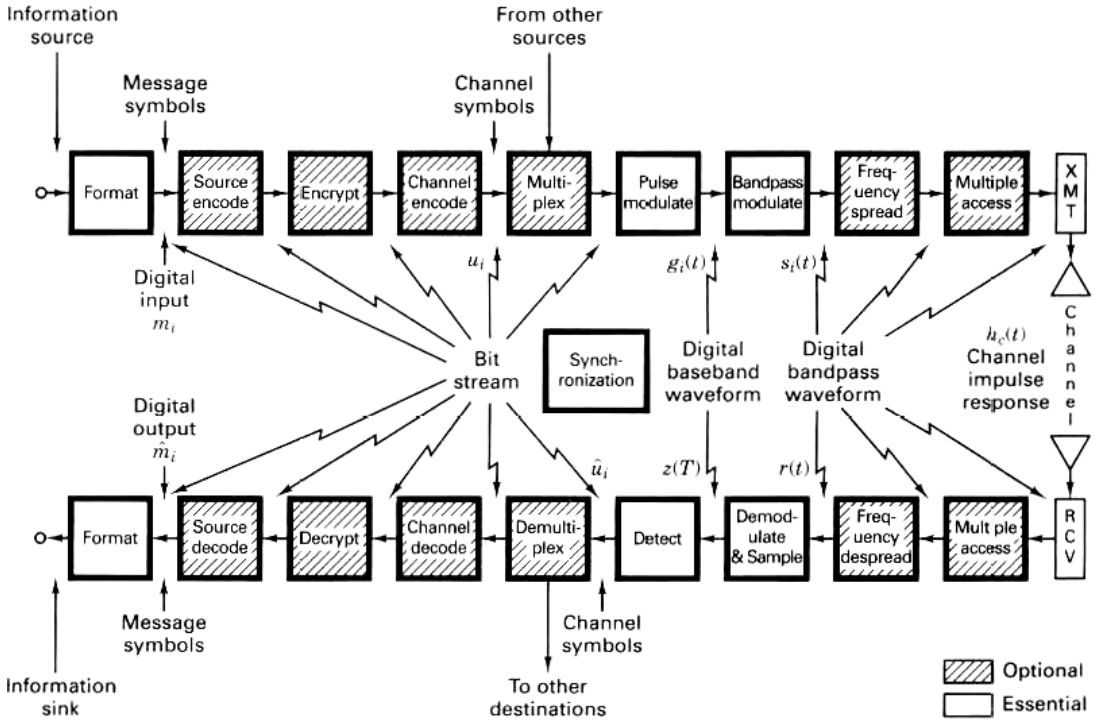
Ειδικά μάλιστα σε επικοινωνιακά συστήματα νέας γενιάς, π.χ. σε δορυφορικά δίκτυα υπολογιστών, όπου απαιτείται πολύ μικρό ποσοστό λαθών η κωδικοποίηση καναλιού θεωρείται απαραίτητη και η ανάλυση γίνεται για το ποια συγκεκριμένη τεχνική κωδικοποίησης θα χρησιμοποιηθεί λαμβάνοντας υπόψη τα χαρακτηριστικά και τις απαιτήσεις του συστήματος.

Πιο παραστατικά τώρα, δίνουμε το παρακάτω γενικό σχήμα για τις διαδικασίες κωδικοποίησης / αποκωδικοποίησης:



Σχήμα 1.1 Η διαδικασία κωδικοποίησης και αποκωδικοποίησης

Όσον αφορά τώρα τη φάση κατά την οποία πραγματοποιούνται οι εν λόγω διαδικασίες στον πομπό και στον δέκτη, παραθέτουμε το παρακάτω αναλυτικό μπλοκ διάγραμμα ενός γενικού ψηφιακού τηλεπικοινωνιακού συστήματος

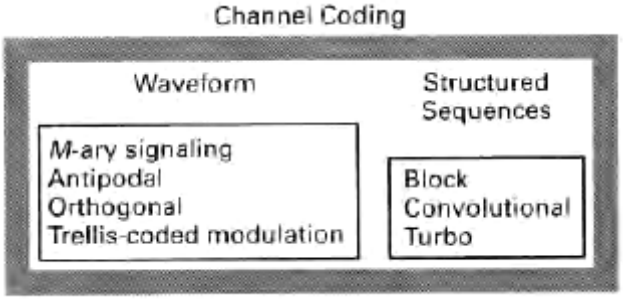


Σχήμα 1.2 Γενικό μπλοκ διάγραμμα ψηφιακού τηλεπικοινωνιακού συστήματος

Σε αυτό το σημείο να τονίσουμε ότι στο εξής ο όρος κωδικοποίηση θα αναφέρεται μόνο στην κωδικοποίηση καναλιού προς αποφυγή σύγχυσης με άλλου είδους κωδικοποιήσεις, όπως κωδικοποίηση πηγής (source coding).

1.2 Γενική Κατηγοριοποίηση Τεχνικών Κωδικοποίησης Καναλιού

Η πιο γενική κατηγοριοποίηση μεταξύ των τεχνικών κωδικοποίησης καναλιού συνίσταται στο διαχωρισμό τους σε κωδικοποιήσεις κυματομορφών (waveform coding) και κωδικοποιήσεις δομημένων ακολουθιών (structured sequences) ή αλλιώς δομημένου πλεονασμού (structured redundancy). Στην πρώτη κατηγορία, έχουμε μετατροπή των κυματομορφών σε νέες κυματομορφές με στόχο η διαδικασία εκτίμησης σήματος στο δέκτη να έχει πιθανοτικά λιγότερα λάθη. Στη δεύτερη κατηγορία, έχουμε εισαγωγή στα αρχικά δεδομένα (αρχικά bits) επιπλέον δεδομένων (επιπλέον bits). Μετά την εισαγωγή με δομημένο τρόπο των επιπλέον bits παίρνουμε το κωδικοποιημένο μήνυμα, που θα είναι και το μήνυμα που θα μεταδοθεί πάνω από το δίαυλο. Τα επιπλέον αυτά bits συνεισφέρουν στον εντοπισμό ή και τη διόρθωση σφαλμάτων, που εισάγονται από το δίαυλο, κατά τη φάση αποκωδικοποίησης στο δέκτη. Κοινό σημείο και των δύο κατηγοριών είναι ότι η διαδικασία κωδικοποίησης παράγει κωδικοποιημένα σήματα με καλύτερες ιδιότητες απόστασης ανάμεσας τους σε σχέση με τα ακωδικοποίητα. Η έννοια της απόστασης χρησιμοποιείται εδώ όπως αυτή ορίζεται στη θεωρία Πληροφορίας. Στο Σχήμα 1.3 φαίνεται αυτός ο αρχικός διαχωρισμός ανάμεσα στις τεχνικές κωδικοποίησης.



Σχήμα 1.3 Γενικός διαχωρισμός τεχνικών κωδικοποίησης

Στην εργασία αυτή, το βάρος θα δοθεί στη μελέτη κωδικοποίησης με δομημένες ακολουθίες ή αλλιώς με δομημένο πλεονασμό. Μάλιστα, οι LDPC (Low-density parity check) κώδικες που θα αποτελέσουν και το κύριο αντικείμενο μελέτης ανήκουν στην υπο-κατηγορία Block (συμπαγών) κωδίκων (βλ. Σχήμα 1.3).

1.3 Οργάνωση της εργασίας

Η εργασία θα ακολουθήσει την παρακάτω δομή:

- **Κεφάλαιο 2**: Παρουσίαση βασικών χαρακτηριστικών των τεχνικών κωδικοποίησης καναλιού. Έμφαση στη μελέτη γραμμικών συμπαγών κωδίκων αλλά και στον τρόπο αξιολόγησης των σχημάτων κωδικοποίησης γενικότερα.
- **Κεφάλαιο 3**: Θεωρητική ανάλυση της προσομοιωτικής διαδικασίας. Βήμα προς βήμα η διαδικασία μοντελοποίησης του τηλεπικοινωνιακού συστήματος που επιθυμούμε να προσομοιώσουμε με απώτερο στόχο τη μελέτη επίδοσης LDPC κωδίκων.
- **Κεφάλαιο 4**: Εκτενής παρουσίαση των LDPC κωδίκων. Ανάλυση των διαδικασιών κωδικοποίησης / αποκωδικοποίησης τους.
- **Κεφάλαιο 5:** Παρουσίαση και σχολιασμός των αποτελεσμάτων της προσομοίωσης μας. Δίνεται ιδιαίτερη έμφαση στην εξαγωγή συμπερασμάτων σχετικά με την επίδοση LDPC κώδικα συναρτήσει παραγόντων που αφορούν εσωτερικά χαρακτηριστικά του κώδικα και συναρτήσει εξωτερικών παραγόντων, όπως ο τύπος διαύλου και το σχήμα διαμόρφωσης.
- **Κεφάλαιο 6**: Παρουσίαση των κυριότερων εφαρμογών των LDPC κωδίκων στην αιχμή της τεχνολογίας. Νύξη για τα τρέχοντα ερευνητικά ζητήματα γύρω από τους LDPC κώδικες.
- **Κεφάλαιο 7**: Σύνοψη των πεδίων συμβολής της εργασίας και προτάσεις για μελλοντική έρευνα.

2 *Τεχνικές Κωδικοποίησης Καναλιού*

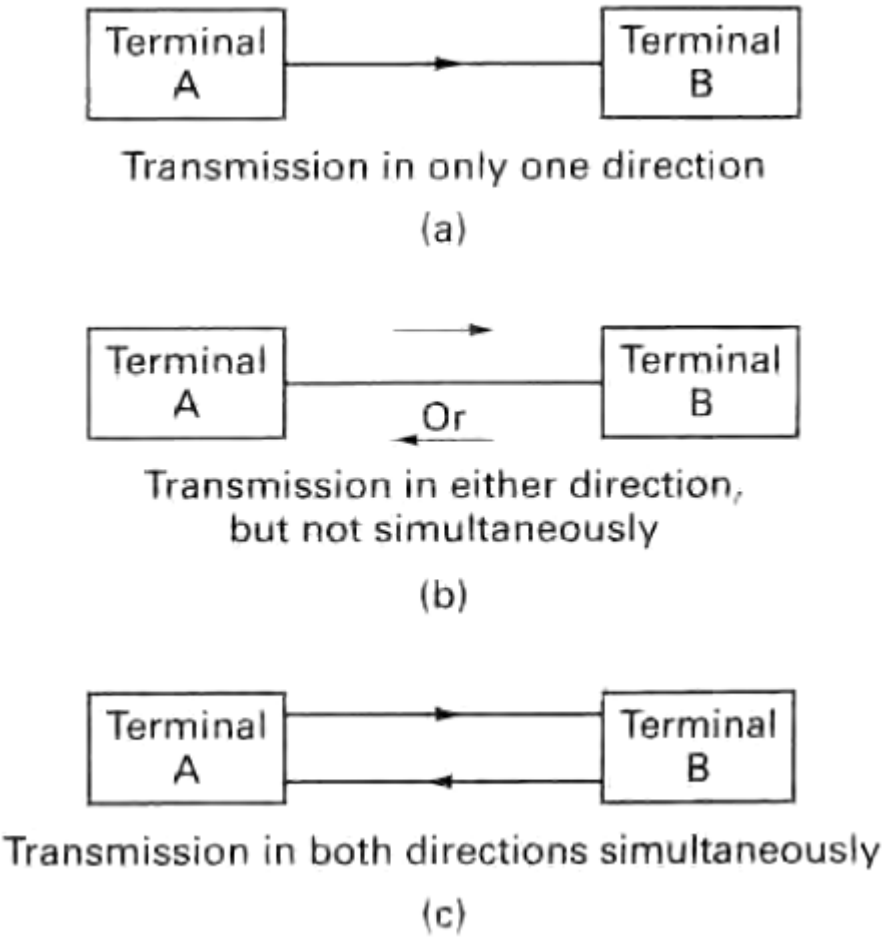
2.1 Γενικά

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα βασικά χαρακτηριστικά των τεχνικών κωδικοποίησης καναλιού. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, η ανάλυση μας θα επικεντρωθεί στην κωδικοποίηση με δομημένο πλεονασμό. Έχει ήδη αναφερθεί ότι στόχος της κωδικοποίησης καναλιού είναι η όσο το δυνατόν καλύτερη αντιμέτωπιση των σφαλμάτων που εισάγει ο δίαυλος πάνω στο μεταδιδόμενο σήμα. Πάνω σε αυτή τη βασική αρχή υπάρχουν δύο μεγάλες κατηγορίες όσον αφορά τον τρόπο αντιμετώπισης – ελέγχου των λαθών. Στην πρώτη κατηγορία, εντάσσουμε τη μέθοδο κατά την οποία γίνεται εντοπισμός των λαθών και επανεκπομπή (error detection and retransmission). Αναλυτικότερα, προστίθενται στα αρχικά bits πληροφορίας bits ισοτιμίας (parity bits) που καθιστούν δυνατό τον εντοπισμό πιθανών λαθών. Σε περίπτωση εντοπισμού λάθους ή λαθών, ο δέκτης δεν κάνει προσπάθεια διόρθωσης τους αλλά στέλνει στον πομπό αίτηση επανεκπομπής των δεδομένων που ελήφθησαν εμπεριέχοντας λάθη. Από την παραπάνω περιγραφή γίνεται κατανοητό ότι η μέθοδος ‘error detection and retransmission’ (στη συνέχεια θα τη συναντήσουμε ως ARQ) απαιτεί διαθεσιμότητα αμφίδρομου καναλιού. Η δεύτερη μεγάλη κατηγορία ελέγχου λαθών είναι η πρόσθια διόρθωση λαθών (forward error correction, FEC). Σε αυτή την περίπτωση, τα επιπλέον bits γνωστά και ως bits ισοτιμίας, σχεδιάζονται έτσι ώστε όχι μόνο να εντοπίζουν τυχόν σφάλματα αλλά να τα διορθώνουν κιόλας. Να τονίσουμε βέβαια ότι ένας κώδικας δεν μπορεί να διορθώσει γενικά όλα τα σφάλματα, καθώς υπάρχουν περιορισμοί στη λεγόμενη διορθωτική ικανότητα του όπως θα δούμε στη συνέχεια. Στα συστήματα που χρησιμοποιούν FEC βλέπουμε ότι δεν στέλνονται δεδομένα από το δέκτη στον πομπό, δεν υπάρχει δηλαδή ανάγκη διαθεσιμότητας καναλιού ανάδρασης αλλά αρκεί μονόδρομο κανάλι. Επειδή πολλές φορές η μέθοδος ελέγχου σφαλμάτων που θα χρησιμοποιηθεί, εξαρτάται από το αν το κανάλι δίνει τη δυνατότητα αμφίδρομης επικοινωνίας ή όχι κρίνεται σκόπιμη η αναλυτικότερη παρουσίαση των διαφόρων τύπων καναλιού όσον αφορά το εν λόγω ζήτημα.

2.1.1 Τύποι σύνδεσης τερματικών

Σε συνέχεια της προηγούμενης ανάλυσης, παραθέτουμε τους βασικούς τρόπους συνδετικότητας μεταξύ τερματικών σταθμών. Όπως φαίνεται και στο Σχήμα 2.1 έχουμε τρεις βασικούς τύπους καναλιού σε σχέση με τη σύνδεση δύο τερματικών: μονόδρομο (simplex), ημι-αμφίδρομο (half-duplex) και πλήρως αμφίδρομο (full duplex). Στην περίπτωση μονόδρομου καναλιού (Σχήμα 2.1a) δεδομένα στέλνονται μόνο από το τερματικό Α προς το Β και ποτέ κατά την αντίθετη κατεύθυνση. Ημι-αμφίδρομο κανάλι (Σχήμα 2.1b) σημαίνει ότι δεδομένα μπορούν να στέλνονται και προς τις δύο κατευθύνσεις, όχι όμως ταυτόχρονα. Τέλος, το πλήρως αμφίδρομο

κανάλι (Σχήμα 2.1c) παρέχει τη δυνατότητα ταυτόχρονης αμφίδρομης ανταλλαγής δεδομένων.



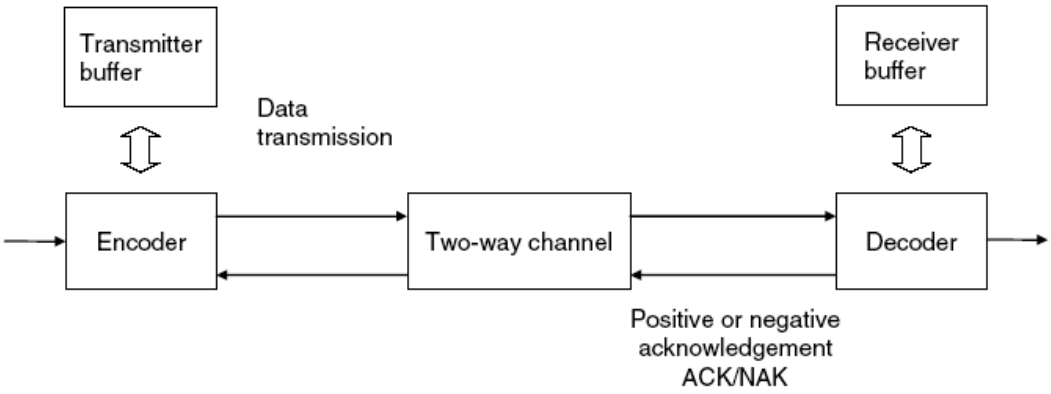
Σχήμα 2.1 Τύποι συνδετικότητας. (a) Μονόδρομο κανάλι. (b) Ημι-αμφίδρομο κανάλι. (c) Πλήρως αμφίδρομο κανάλι

2.2 Αυτόματη αίτηση επανάληψης (ARQ)

Όταν η διαδικασία ελέγχου σφαλμάτων συνίσταται μόνο στον εντοπισμό τους και όχι και στη διόρθωση τους, τότε υπάρχει ανάγκη να σταλεί από το δέκτη στον πομπό ειδοποίηση για το εντοπισμένο σφάλμα και ενημέρωση ότι απαιτείται επανεκπομπή. Τέτοιες μέθοδοι είναι ευρέως γνωστές ως μέθοδοι αυτόματης αίτησης επανάληψης (automatic repeat request) ή αυτόματης αίτησης επανεκπομπής (automatic retransmission query, ARQ). Η γενική λειτουργία ARQ συστημάτων απεικονίζεται στο Σχήμα 2.2.

Συνοπτικά μπορούμε να πούμε ότι μετά την δημιουργία νέου κωδικοποιημένου διανύσματος, αυτό αποθηκεύεται πρώτα σε έναν buffer και ύστερα μεταδίδεται μέσω του καναλιού. Το κανάλι όπως γνωρίζουμε εισάγει θόρυβο, ο οποίος είναι δυνατόν να αλλοιώσει το αρχικό διάνυσμα. Έτσι, γίνεται έλεγχος από τον αποκωδικοποιητή στο δέκτη αν το ληφθέν διάνυσμα είναι έγκυρο ή όχι. Αν προκύψει έγκυρο, ο αποκωδικοποιητής θεωρεί ότι δεν υπάρχει κανένα σφάλμα και στέλνεται πίσω στον

πομπό θετική επιβεβαίωση (ACK). Σε περίπτωση μη εγκυρότητας, συνεπάγεται ότι κατά τη μετάδοση έχουν εισαχθεί σφάλματα και στέλνεται πίσω στον πομπό σήμα αρνητικής επιβεβαίωσης (NAK). Αυτόματα τότε, ξεκινάει η διαδικασία επανεκπομπής του κωδικοποιημένου διανύσματος το οποίο είναι, όπως προαναφέραμε, αποθηκευμένο στον buffer του πομπού.

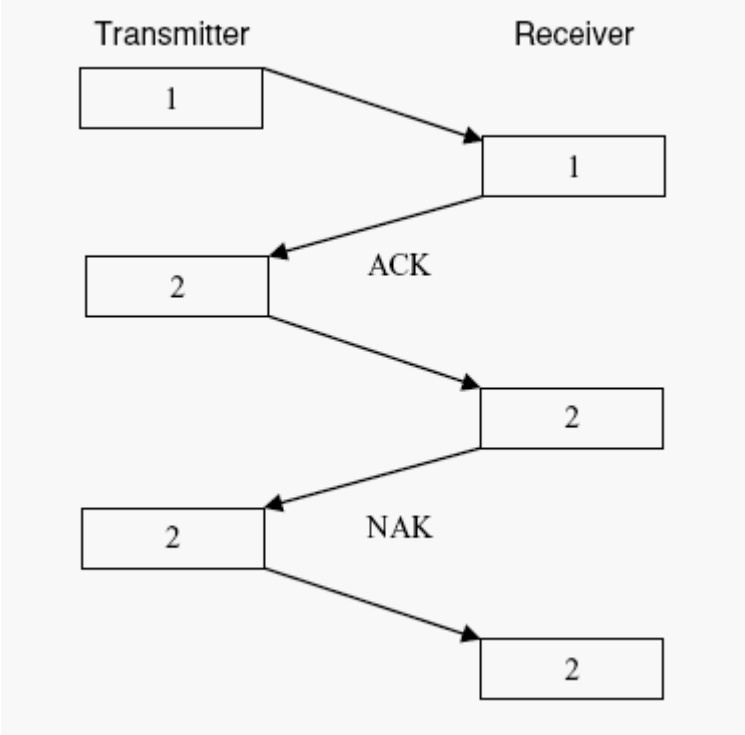


Σχήμα 2.2 Μπλοκ διάγραμμα ARQ συστήματος

Στη συνέχεια θα δώσουμε μια σύντομη περιγραφή των πλέον γνωστών και χρησιμοποιούμενων πρωτοκόλλων τύπου ARQ.

2.2.1 Stop and wait

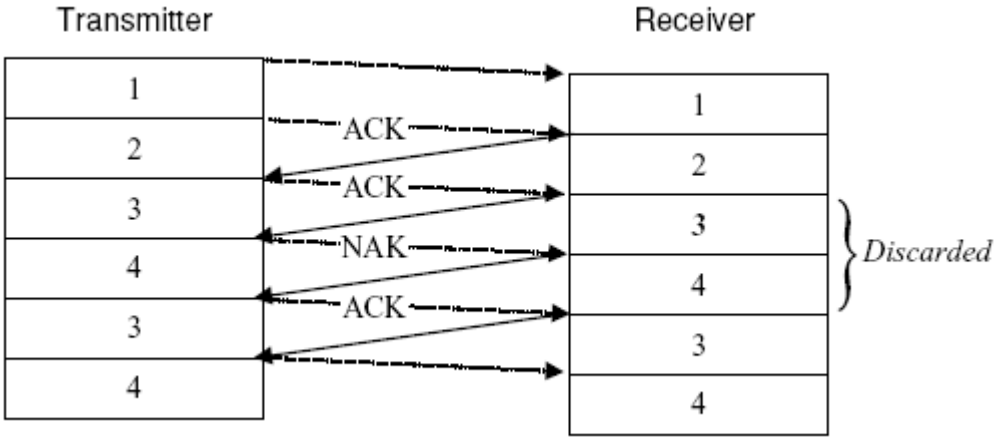
Η βασική αρχή του stop and wait σχήματος είναι ότι ο πομπός δεν μεταδίδει νέο πακέτο πληροφορίας αν δεν λάβει απάντηση (θετική ή αρνητική) για το προηγούμενο. Σαφώς, σε περίπτωση λήψης NAK έχουμε επανεκπομπή, ειδάλλως λήψη θετικής επιβεβαίωσης οδηγεί στην εκπομπή του επόμενου στη σειρά πακέτου πληροφορίας. Το Σχήμα 2.3 είναι αρκετά βοηθητικό, καθώς περιέχει και τις δύο πιθανές περιπτώσεις (ACK / NAK). Σε αυτό το πρωτόκολλο, γίνεται αποθήκευση μόνο ενός πακέτου κάθε φορά στον buffer, δηλαδή ο καταναλισκόμενος αποθηκευτικός χώρος είναι ο ελάχιστος δυνατός. Από την άλλη πλευρά, ο χρόνος αναμονής του πομπού D ενδέχεται να είναι αρκετά μεγάλος. Σε κάθε περίπτωση, για τον χρόνο αναμονής D ισχύει $D \geq 2T_d$, όπου T_d η καθυστέρηση διάδοσης από άκρη σε άκρη.



Σχήμα 2.3 Stop and wait ARQ πρωτόκολλο

2.2.2 Go-Back N

Σε αυτό το πρωτόκολλο τύπου ARQ έχουμε συνεχή εκπομπή πακέτων από τον πομπό, σε αντίθεση με το Stop and wait πρωτόκολλο που συναντήσαμε προηγουμένως. Σε περίπτωση λήψης NAK τώρα, ο πομπός γυρνά πίσω στο αντίστοιχο πακέτο και αρχίζει επανεκπομπή από αυτό το πακέτο. Κάτι τέτοιο απαιτεί, την δυνατότητα αποθήκευσης στον buffer του πομπού N πακέτων, όπου το N καθορίζεται από την καθυστέρηση πλήρους διαδρομής (round–trip delay) του συστήματος. Από την πλευρά του δέκτη τώρα, αγνοούνται τα υπόλοιπα N-1 πακέτα που λαμβάνονται μετά από το λανθασμένο πακέτο, ακόμα και αν αυτά δεν περιέχουν σφάλματα, με σκοπό να διατηρηθεί η σωστή ακολουθία πακέτων. Έτσι, στον buffer του δέκτη αρκεί αποθηκευτικός χώρος ίσος με ένα πακέτο. Το Σχήμα 2.4 περιγράφει τη διαδικασία με N=2.



Σχήμα 2.4 Παράδειγμα Go-Back N πρωτοκόλλου (N=2)

2.3 Κωδικοποίηση Δομημένων Ακολουθιών

Στο εξής θα ασχοληθούμε με κώδικες ελέγχου ισοτιμίας. Τέτοιοι κώδικες αναφέρονται ως κώδικες δομημένων ακολουθιών, καθώς κατά τη διαδικασία κωδικοποίησης προστίθενται με δομημένο τρόπο στα αρχικά δεδομένα επιπλέον δεδομένα, έτσι ώστε να γίνει εφικτός ο εντοπισμός ή η διόρθωση σφαλμάτων. Οι κώδικες με δομημένες ακολουθίες μπορούν να χωριστούν σε τρεις μεγάλες κατηγορίες, όπως φαίνεται και στο Σχήμα 1.3: συμπαγείς κώδικες (block), συνελικτικοί κώδικες (convolutional) και turbo κώδικες. Εμείς θα σταθούμε κυρίως στην περιγραφή των συμπαγών κωδίκων, μιας και το κύριο αντικείμενο μελέτης μας που είναι οι LDPC κώδικες ανήκουν σε αυτό το σύνολο.

2.3.1 Κώδικες Ελέγχου Ισοτιμίας

Πριν προχωρήσουμε σε περαιτέρω ανάλυση, κρίνεται σκόπιμο να εισάγουμε την έννοια του ρυθμού κώδικα, έννοια βασική για κάθε σχήμα κωδικοποίησης δομημένων ακολουθιών. Κατά τη φάση κωδικοποίησης λοιπόν, ο κωδικοποιητής λαμβάνει ως είσοδο k bits πληροφορίας ή με άλλα λόγια k bits δεδομένων. Σαν έξοδο παράγει n bits, όπου n>k. Τα n bits θα τα ονομάζουμε κωδικοποιημένα bits ή κωδικοποιημένα σύμβολα ή κωδική λέξη ως σύνολο. Τα (n-k) bits που προσθέτει ο κωδικοποιητής αποτελούν τα πλεονάζοντα bits και δεν φέρουν νέα ποσότητα πληροφορίας, αλλά προστίθενται για τον εντοπισμό ή τη διόρθωση σφαλμάτων όπως προαναφέρθηκε. Ένας τέτοιος κώδικας συμβολίζεται ως (n, k) κώδικας.Ο ρυθμός κώδικα ορίζεται ως ο λόγος των bits πληροφορίας προς τα συνολικά bits, δηλαδή εκφράζεται από το λόγο k/n.

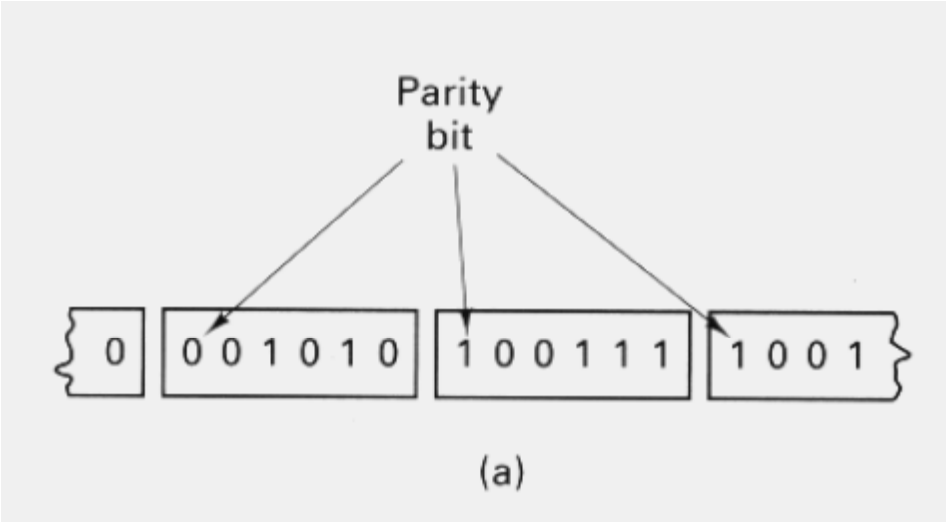
2.3.1.1 Κώδικες Μονού Ελέγχου Ισοτιμίας

Οι κώδικες ελέγχου ισοτιμίας (parity-check codes) χρησιμοποιώντας γραμμικά αθροίσματα των bits πληροφορίας, παράγουν bits ισοτιμίας (parity bits) προς εντοπισμό ή διόρθωση σφαλμάτων. Ένας κώδικας μονού ελέγχου ισοτιμίας

κατασκευάζεται προσθέτοντας ένα μόνο bit ισοτιμίας σε κάθε μπλοκ (πακέτο) που φέρει bits πληροφορίας. Η τιμή που θα ανατεθεί στο bit ισοτιμίας θα είναι 0 ή 1 έτσι ώστε το άθροισμα modulo-2 των bits της κωδικής λέξης να είναι άρτιο ή περιττό, ανάλογα με τη σχεδίαση του κώδικα. Σε περίπτωση άρτιου αθροίσματος έχουμε κώδικες άρτιας ισοτιμίας και αναλόγως κώδικες περιττής ισοτιμίας. Η modulo-2 άθροιση ορίζεται ως εξής

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 0$$
$$1 \oplus 1 = 0$$

Στο Σχήμα 2.5 δείχνεται με ποιο τρόπο προστίθεται το bit ισοτιμίας μέσα σε κάθε μπλοκ, έτσι ώστε να κρατηθεί άρτια ισοτιμία εν προκειμένω.



Σχήμα 2.5 Περίπτωση κώδικα άρτιας ισοτιμίας

Στην πλευρά του δέκτη τώρα, ο αποκωδικοποιητής θα εξετάσει αν το άθροισμα των bits, για κάθε ληφθείσα κωδική λέξη ξεχωριστά, ικανοποιεί την απαίτηση του συστήματος για άρτια ισοτιμία. Εάν το άθροισμα για μια τυχαία κωδική λέξη προκύψει 0 θεωρείται λέξη δίχως σφάλματα, αλλιώς αν το άθροισμα προκύψει 1 συνεπάγεται ότι η κωδική λέξη περιέχει σφάλματα. Όπως προκύπτει, ένας τέτοιος κώδικας δεν μπορεί να διορθώσει σφάλματα καθώς σε περίπτωση εντοπισμού τους δεν υπάρχει καμμία παραπάνω πληροφορία για τη θέση του σφάλματος (ή των σφαλμάτων). Μάλιστα, αυτό που μπορεί να εντοπίσει είναι μόνο η παρουσία περιττού αριθμού λαθών ανά κωδική λέξη. Αντίθετα, σε περίπτωση άρτιου αριθμού σφαλμάτων έχουμε την περίπτωση μη εντοπισμένου σφάλματος. Παραδείγματος χάριν, αν η κωδική λέξη 001010 φτάσει στον αποκωδικοποιητή ως 001111 το σφάλμα δεν εντοπίζεται μιας και ικανοποιείται η συμφωνημένη συνθήκη άρτιας ισοτιμίας. Θεωρώντας τώρα ότι τα bits μεταδίδονται με ανεξάρτητο τρόπο και συμβολίζοντας p την πιθανότητα ένα bit να ληφθεί με σφάλμα, ισχύει η ακόλουθη έκφραση για την πιθανότητα να ληφθούν j λανθασμένα bits εντός πακέτου n bits

$$P(j,n) = \binom{n}{j} p^j (1-p)^{n-j} \tag{2.1}$$

όπου

$$\binom{n}{j} = \frac{n!}{j!(n-j)!}$$

Από την (2.1) προκύπτει ότι για κώδικες μονού ελέγχου ισοτιμίας, η πιθανότητα μη εντοπισμού σφάλματος δίνεται από τη σχέση

$$P_{nd} = \sum_{j=1}^{(n-1)/2} \binom{n}{2j} p^{2j} (1-p)^{n-2j} \tag{2.2}$$

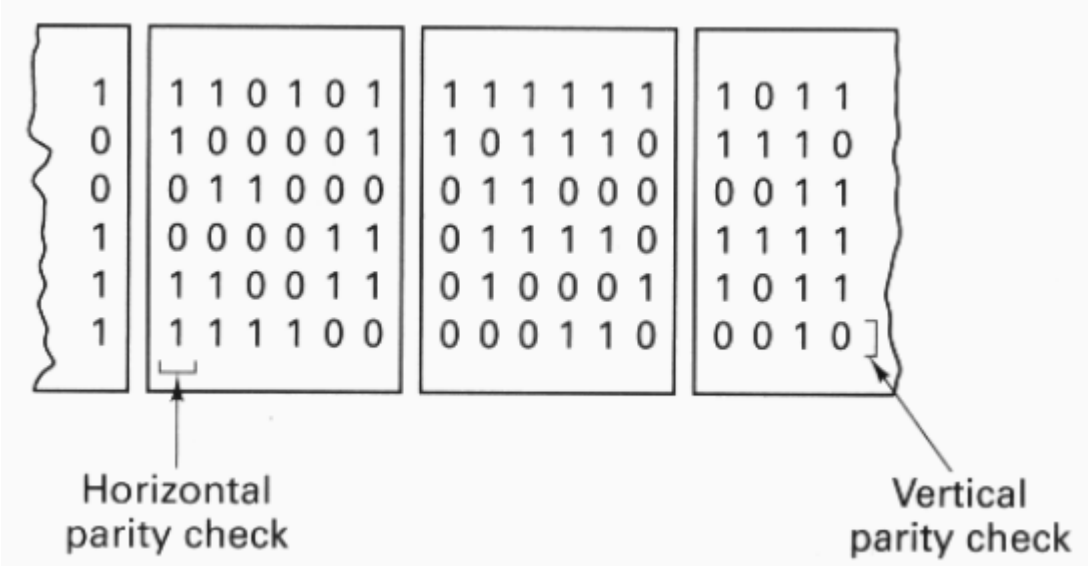
για n περιττό και

$$P_{nd} = \sum_{j=1}^{n/2} \binom{n}{2j} p^{2j} (1-p)^{n-2j} \tag{2.3}$$

για n άρτιο.

2.3.1.2 Ορθογώνιοι κώδικες

Οι ορθογώνιοι κώδικες αποτελούν τη 'διδιάστατη' εκδοχή των κωδίκων που μόλις περιγράφηκαν. Κατασκευαστικά, αρχικά οργανώνουμε τα bits πληροφορίας σε ορθογώνια διαστάσεων M x N. Ύστερα, εφαρμόζεται οριζόντιος και κατακόρυφος έλεγχος ισοτιμίας σε κάθε γραμμή και στήλη του M x N πίνακα, δημιουργώντας έτσι τον κωδικοποιημένο πίνακα διαστάσεων (M+1) x (N+1) και κάνοντας το ρυθμό κώδικα να ισούται με MN / ((M+1) (N+1)). Η διαδικασία κατασκευής ενός τέτοιου κώδικα δίνεται παραστατικά στο επόμενο σχήμα, όπου M=N=5 και εφαρμόζεται πάλι άρτια ισοτιμία.



Σχήμα 2.6 Παράδειγμα ορθογώνιου κώδικα ελέγχου ισοτιμίας

Στο παράδειγμα μας, σε περίπτωση ενός μόνο σφάλματος σε ένα από τα 36 bits του κωδικοποιημένου πίνακα θα προκληθεί αποτυχία ισοτιμίας σε μία μόνο γραμμή και σε μία μόνο στήλη του πίνακα. Αποτυχία όμως μόνο στη γραμμή i και μόνο στη στήλη j σημαίνει ότι το bit με σφάλμα βρίσκεται στη θέση (i, j). Αυτός ο ακριβής εντοπισμός της θέσης του μοναδικού σφάλματος σημαίνει και διορθωτική ικανότητα του κώδικα και είναι αυτό που τον ξεχωρίζει από τους μέχρι τώρα αναλυθέντες κώδικες. Όπως βλέπουμε όμως, η διορθωτική ικανότητα τέτοιων κωδίκων περιορίζεται μόνο στην περίπτωση ενός σφάλματος ανά κωδική λέξη. Δεν παύει όμως αυτός ο κώδικας να αποτελεί την εισαγωγή μας στην κύρια κατηγορία κωδίκων που θα μελετηθούν: τους κώδικες διόρθωσης λαθών.

2.4 Γραμμικοί Συμπαγείς Κώδικες

Στους γραμμικούς συμπαγείς κώδικες, η πληροφορία οργανώνεται σε πακέτα ή αλλιώς διανύσματα σταθερού μήκους k bits. Το διάνυσμα που φέρει κάθε φορά αυτή την πληροφορία ονομάζεται διάνυσμα πληροφορίας ή διάνυσμα μηνύματος και συμβολίζεται με **m.** Έτσι, $\mathbf{m=(m_0, m_1,...m_{k-1})}$ όπου τα στοιχεία του διανύσματος m θεωρούμε ότι είναι 0 ή 1, ο κώδικας δηλαδή ανήκει στο δυαδικό πεδίο GF(2). Γενικά, είναι δυνατή η κατασκευή κωδίκων που δεν περιορίζονται στο δυαδικό αλφάβητο, σε αυτή την εργασία όμως θα ασχοληθούμε μόνο με δυαδικούς κώδικες που αποτελούν και τη συνηθέστερη μορφή. Σε τέτοια περίπτωση, το διάνυσμα m μπορεί να πάρει 2^k διαφορετικές μορφές. Η διαδικασία της κωδικοποίησης έγκειται στη μετατροπή του m σε νέο διάνυσμα $\mathbf{c=(c_0, c_1,...c_{n-1})}$ μήκους n>k. Τότε, ο γραμμικός συμπαγής κώδικας θα συμβολίζεται (n, k).

Όταν οι τιμές των n, k είναι σχετικά μικρές η παραπάνω μετατροπή – αντιστοίχιση μπορεί να γίνει μέσω ενός κατάλληλου πίνακα που θα αντιστοιχίζει κάθε ένα από τα 2^k διανύσματα πληροφορίας σε μία κωδική λέξη μήκους n. Όταν όμως τα n, k είναι μεγάλα οι διαστάσεις ενός τέτοιου πίνακα είναι πρακτικά απαγορευτικές και έτσι υπάρχει η ανάγκη ανάπτυξης ενός γεννήτορα μηχανισμού για τη διαδικασία

κωδικοποίησης. Δεδομένης αυτής της ανάγκης, η ύπαρξη γραμμικότητας σε αυτόν τον μηχανισμό βοηθά πολύ στην απλοποίηση της διαδικασίας.

Ορισμός 2.1: Ένας συμπαγής κώδικας μήκους n και 2^k δυνατών μηνυμάτων πληροφορίας λέγεται γραμμικός συμπαγής κώδικας (n, k) εάν οι 2^k κωδικές του λέξεις σχηματίζουν ένα διανυσματικό υπόχωρο, διάστασης k, του διανυσματικού χώρου V_n ο οποίος περιλαμβάνει όλα τα διανύσματα μήκους n με στοιχεία στο πεδίο GF(2).

Σαν αποτέλεσμα του παραπάνω ορισμού, ένας γραμμικός συμπαγής κώδικας χαρακτηρίζεται από το ότι το μηδενικό διάνυσμα είναι πάντα μία από τις κωδικές λέξεις και από το ότι το άθροισμα δύο οποιωνδήποτε κωδικών λέξεων αποτελεί επίσης κωδική λέξη.

2.4.1 Περιγραφή μέσω γεννήτορα πίνακα G

Αφού ένας γραμμικός συμπαγής κώδικας (n, k) αποτελεί ένα διανυσματικό υπόχωρο του V_n , θα υπάρχουν k γραμμικώς ανεξάρτητα διανύσματα έτσι ώστε κάθε πιθανή κωδική λέξη να προκύπτει ως γραμμικός συνδυασμός αυτών:

$$\mathbf{c} = m_0 \bullet \mathbf{g_0} \oplus m_1 \bullet \mathbf{g}_1 \oplus \ldots \oplus m_{k-1} \bullet \mathbf{g}_{k-1} \tag{2.4}$$

Με όρους Γραμμικής Άλγεβρας, τα k διανύσματα $\mathbf{g_0}$, ..., $\mathbf{g}_{k-1}$ αποτελούν μία βάση του υποχώρου και όσον αφορά την κωδικοποίηση κάθε ένα από αυτά είναι κωδική λέξη. Τα γραμμικώς ανεξάρτητα διανύσματα μπορούν να παρασταθούν σε μορφή πίνακα ως εξής:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

Ο πίνακας **G** ονομάζεται γεννήτορας πίνακας καθώς αποτελεί γεννήτορα μηχανισμό για την κατασκευή κάθε κωδικής λέξης. Όπως προκύπτει από τα παραπάνω, ο **G** είναι ο πίνακας γραμμικός συνδυασμός των γραμμών του οποίου δημιουργεί την κάθε λέξη που ανήκει στον κώδικα. Ο πίνακας αυτός έχει k γραμμές και n στήλες.
Για δοσμένο διάνυσμα πληροφορίας $m=(m_0, \ldots, m_{k-1})$ η αντίστοιχη κωδική λέξη λαμβάνεται μέσω πολλαπλασιασμού πινάκων όπως φαίνεται στη συνέχεια:

$$c = m \circ G = (m_0, m_1, \ldots, m_{k-1}) \circ \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

$$= (m_0, m_1, \ldots, m_{k-1}) \circ \begin{bmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{bmatrix} = m_0 \bullet \mathbf{g_0} \oplus m_1 \bullet \mathbf{g_1} \oplus \cdots \oplus m_{k-1} \bullet \mathbf{g_{k-1}}$$

Στην παραπάνω διαδικασία, το σύμβολο '$\circ$' δηλώνει εσωτερικό γινόμενο μεταξύ διανυσμάτων ενώ το σύμβολο '$\bullet$' δηλώνει πολλαπλασιασμό διανύσματος με βαθμωτό (μονόμετρο) μέγεθος.

Παράδειγμα 2.1: Έστω ο ακόλουθος 3 x 5 γεννήτορας πίνακας

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

Τότε, αν στην είσοδο του κωδικοποιητή έχουμε το μήνυμα **m**=(1 1 1) στην έξοδο θα πάρουμε την αντίστοιχη κωδική λέξη **c** ως εξής:

$$c = [1 \quad 1 \quad 1] \circ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} =$$

$$[1\cdot 1 \oplus 1\cdot 0 \oplus 1\cdot 0 \quad 1\cdot 0 \oplus 1\cdot 1 \oplus 1\cdot 0 \quad 1\cdot 0 \oplus 1\cdot 0 \oplus 1\cdot 1 \quad 1\cdot 1 \oplus 1\cdot 1 \oplus 1\cdot 0 \quad 1\cdot 1 \oplus 1\cdot 0 \oplus 1\cdot 1]$$

$$= [1 \quad 1 \quad 1 \quad 0 \quad 0]$$

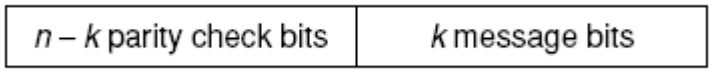
Με τον ίδιο τρόπο προκύπτει ο παρακάτω πίνακας

Πίνακας 2.1 Κωδικές λέξεις του (5, 3) κώδικα του παραδείγματος 2.1

Μήνυμα πληροφορίας	Κωδική λέξη
000	00000
001	00101
010	01010
011	01111
100	10011
101	10110
110	11001
111	11100

2.4.2 Συστηματικοί Συμπαγείς Κώδικες

Συστηματικός συμπαγής κώδικας είναι ο γραμμικός συμπαγής κώδικας του οποίου τα πρώτα (n-k) ψηφία κάθε λέξης προστίθενται από τον κωδικοποιητή, ενώ τα επόμενα k ταυτίζονται με την ακωδικοποίητη πληροφορία. Η δομή αυτή απεικονίζεται στο επόμενο Σχήμα 2.7



Σχήμα 2.7 Συστηματική μορφή κωδικής λέξης συμπαγούς κώδικα

Όπως φαίνεται, τα ψηφία (bits) που εισάγονται από τον κωδικοποιητή ονομάζονται ψηφία ελέγχου ισοτιμίας.

Στη βιβλιογραφία συναντάται και ο 'συμμετρικός', σε σχέση με τον παραπάνω, ορισμός συστηματικού κώδικα. Δηλαδή, ορίζεται ως συστηματικός ο κώδικας του οποίου τα πρώτα k ψηφία ταυτίζονται με την ακωδικοποίητη πληροφορία, ενώ τα επόμενα (n-k) προστίθενται από τον ακωδικοποιητή. Τέτοια περίπτωση αποτελεί ο κώδικας του παραδείγματος 2.1 όπως φαίνεται από τον πίνακα 2.1. Για παράδειγμα, το μήνυμα 111 μετατρέπεται σε 11100, ενώ με βάση το δικό μας ορισμό για να ήταν συστηματικός ο κώδικας θα έπρεπε το μήνυμα 111 να μετατρεπόταν σε κωδική λέξη της μορφής xx111. Όπως γίνεται αντιληπτό, οι όποιες ιδιότητες συστηματικών κωδίκων είναι ουσιαστικά οι ίδιες και στις δύο περιπτώσεις. Στη συνέχεια πάντως της εργασίας όπου απαντάται ο όρος συστηματικός κώδικας η αναφορά θα γίνεται στη δομή του σχήματος 2.7.

Ένας συστηματικός γραμμικός συμπαγής κώδικας προσδιορίζεται μονοσήμαντα από γεννήτορα πίνακα της μορφής

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \underbrace{\begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}}_{\text{submatrix } \boldsymbol{P} \text{ k} \times (\text{n} - \text{k})} \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\text{submatrix } \boldsymbol{I} \text{ k} \times \text{k}}$$

που μπορεί να γραφεί συνοπτικά και ως

$$G = \begin{bmatrix} P & I_k \end{bmatrix} \tag{2.5}$$

όπου με I_k συμβολίζεται ο k x k μοναδιαίος πίνακας.

Η δομή των συστηματικών κωδίκων μας επιτρέπει να εξάγουμε αναλυτικές εκφράσεις που συνδέουν τα bits ελέγχου ισοτιμίας (parity check bits) με τα bits ακωδικοποίητης πληροφορίας (message bits). Αν λοιπόν $m=(m_0, m_1, \ldots, m_{k-1})$ είναι το ακωδικοποίητο διάνυσμα πληροφορίας και $c=(c_0, c_1, \ldots, c_{n-1})$ το αντίστοιχο κωδικοποιημένο διάνυσμα ισχύει:

$$
\begin{aligned}
&c_{n-k+i} = m_i,\ 0 \le i \le k-1 \\
&c_j = m_0 \bullet p_{0j} + m_1 \bullet p_{1j} + \cdots + m_{k-1} \bullet p_{k-1,j},\ 0 \le j < n-k
\end{aligned}
\tag{2.6}
$$

Οι (n-k) εξισώσεις του δεύτερου σκέλους της (2.6) είναι αυτές που καλούνται εξισώσεις ελέγχου ισοτιμίας.

Παράδειγμα 2.2 Έστω ο ακόλουθος γεννήτορας πίνακας γραμμικού συστηματικού συμπαγούς κώδικα (7, 4).

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

Ο πίνακας **G** είναι της μορφής $G = \begin{bmatrix} P & I_k \end{bmatrix}$ με k = 4. Η κάθε κωδική λέξη προκύπτει από τη βασική σχέση

$$
c = m \circ G \tag{2.7}
$$

Στο παράδειγμα μας,

$$
\mathbf{c = m \circ G} = (m_0, m_1, m_2, m_3) \circ \begin{bmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

Τότε οι εξισώσεις ελέγχου ισοτιμίας είναι:

$$
\begin{aligned}
c_0 &= m_0 \oplus m_2 \oplus m_3 \\
c_1 &= m_0 \oplus m_1 \oplus m_2 \\
c_2 &= m_1 \oplus m_2 \oplus m_3
\end{aligned}
$$

ενώ για τα υπόλοιπα bits της κωδικής λέξης .ισχύει:

$$
\begin{aligned}
c_3 &= m_0 \\
c_4 &= m_1 \\
c_5 &= m_2 \\
c_6 &= m_3
\end{aligned}
$$

2.4.3 Περιγραφή μέσω Πίνακα Ελέγχου Ισοτιμίας **H**

Μια εναλλακτική περιγραφή των γραμμικών συμπαγών κωδίκων πραγματοποιείται μέσω του πίνακα ελέγχου ισοτιμίας **H**. Σε έναν κώδικα (n, k) ο πίνακας **H** είναι διαστάσεων (n-k) x n. Η θεμελιώδης ιδιότητα του **H** είναι ότι προκύπτει από τον γεννήτορα πίνακα του κώδικα **G** με βάση την παρακάτω σχέση:

$$G \circ H^T = 0 \tag{2.8}$$

όπου με H^T συμβολίζουμε τον ανάστροφο του **H**.
Από την (2.8) προκύπτει η συστηματική μορφή του **H** δεδομένου συστηματικού πίνακα **G** (2.5).

$$\boldsymbol{H} = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{\text{submatrix } \boldsymbol{I}\,(\mathrm{n}-\mathrm{k}) \times (\mathrm{n}-\mathrm{k})} \underbrace{\begin{bmatrix} p_{00} & p_{10} & \cdots & p_{k-1,0} \\ p_{01} & p_{11} & \cdots & p_{k-1,1} \\ \vdots & \vdots & & \vdots \\ p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}}_{\text{submatrix } \boldsymbol{P^T}\,(\mathrm{n}-\mathrm{k}) \times \mathrm{k}} = \begin{bmatrix} \boldsymbol{I_{n-k}} & \boldsymbol{P^T} \end{bmatrix}$$

Σχήμα 2.8 Γενική συστηματική μορφή του πίνακα ελέγχου ισοτιμίας

Επίσης, μπορούμε να εξάγουμε την ακόλουθη σχέση που ισχύει για κάθε κωδική λέξη **c.**

$$c \circ H^T = m \circ G \circ H^T = 0 \tag{2.9}$$

Η (2.9) αποτελεί σχέση ελέγχου εγκυρότητας μίας κωδικής λέξης. Έτσι, αν στον αποκωδικοποιητή καταφθάσει κωδική λέξη **r** γίνεται έλεγχος αν

$$r \circ H^T = 0$$

Αν ισχύει η παραπάνω ισότητα, το σύστημα δέχεται το διάνυσμα **r** ως έγκυρη κωδική λέξη, ενώ σε αντίθετη περίπτωση το διάνυσμα **r** θεωρείται μη-έγκυρο(πιθανότατα υπέστη αλλοίωση κατά τη μετάδοση του).

Παράδειγμα 2.3 Καθορισμός πίνακα ελέγχου ισοτιμίας δεδομένου του γεννήτορα πίνακα. Έστω λοιπόν συστηματικός γεννήτορας πίνακας 4 x 7, ίδιος με το προηγούμενο παράδειγμα, ως ακολούθως:

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

Ανατρέχοντας στην μορφή (2.5) ο G συντίθεται από τον (υπο)-πίνακα

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

και από το μοναδιαίο (υπο)-πίνακα

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

Από τους παραπάνω υποπίνακες και από το σχήμα 2.8 προκύπτει ο **H**

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

Επιβεβαιώνεται ότι οι διαστάσεις του πίνακα ελέγχου ισοτιμίας είναι (n-k) x n= (7-4) x 7 = 3 x 7.

2.4.4 Έλεγχος Συνδρόμου

Θεωρούμε κώδικα (n, k). Τότε, η προς μετάδοση κωδική λέξη είναι της μορφής $c=(c_0, c_1, \ldots, c_{n-1})$ (μία από τις 2^k διαφορετικές εκδοχές). Έστω **r** το διάνυσμα που αναπαριστά τη ληφθείσα λέξη με $r=(r_0, r_1, \ldots, r_{n-1})$ (μία από τις 2^n διαφορετικές εκδοχές). Ενδεχόμενα σφάλματα που προήλθαν από τη διαδικασία της μετάδοσης της κωδικής λέξης μπορούν να μοντελοποιηθούν μέσω του διανύσματος σφάλματος (ή διάνυσμα λαθών) **e** ως:

$$e = r \oplus c \tag{2.10}$$

Το διάνυσμα **e** έχει μη μηδενικά στοιχεία μόνο στις θέσεις i του ληφθέντος διανύσματος όπου έχει συμβεί κάποιο σφάλμα ($r_i \neq c_i$).
Από την (2.10) παίρνουμε ισοδύναμα

$$c = r \oplus e \text{ και } r = c \oplus e \tag{2.11}$$

σε όρους modulo-2 άθροισης πάντα.
Η σχέση $c = r \oplus e$ μπορεί να χρησιμοποιηθεί από τον αποκωδικοποιητή για διόρθωση του διανύσματος **r** που περιέχει σφάλματα που αντιστοιχούν σε διάνυσμα σφάλματος **e**. Όπως γίνεται εύκολα κατανοητό, το δύσκολο σε πρώτη φάση είναι ο προσδιορισμός του **e**.
Προς αυτή την κατεύθυνση, ορίζουμε το σύνδρομο του **r** ως

$$S = r \circ H^T \tag{2.12}$$

Αν το διάνυσμα **r** αντιστοιχεί σε κωδική λέξη, τότε το σύνδρομο **S** είναι το μηδενικό διάνυσμα όπως προκύπτει και από την (2.9). Αντίθετα, αν το ληφθέν διάνυσμα περιλαμβάνει σφάλματα, τότε το σύνδρομό του θα περιλαμβάνει ορισμένα μη μηδενικά στοιχεία. Σε τέτοια περίπτωση, ο αποκωδικοποιητής ή θα προχωρήσει σε αυτόματη διόρθωση λαθών (σύστημα FEC) ή θα ζητήσει επανεκπομπή του μηνύματος (σύστημα ARQ).
Αφού τώρα $r = c \oplus e$, η (2.12) μπορεί να πάρει την ακόλουθη μορφή:

$$S = r \circ H^T = (c \oplus e) \circ H^T = c \circ H^T \oplus e \circ H^T \tag{2.13}$$

Λόγω της (2.9) η (2.13) γίνεται:

$$S = e \circ H^T \tag{2.14}$$

Η παραπάνω ανάλυση κατέδειξε ότι το σύνδρομο ενός εσφαλμένου ληφθέντος διανύσματος είναι το ίδιο με το σύνδρομο του διανύσματος σφάλματος του. Μία σημαντική ιδιότητα των γραμμικών συμπαγών κωδίκων, θεμελιώδης στη διαδικασία αποκωδικοποίησης, είναι ότι η αντιστοίχιση μεταξύ διορθώσιμων διανυσμάτων σφάλματος και συνδρόμων είναι ένα προς ένα.

Σε αυτό το σημείο αναφέρονται δύο βασικές ιδιότητες που πρέπει να έχει ο πίνακας ελέγχου ισοτιμίας **H**.

- Ο πίνακας ελέγχου ισοτιμίας δεν πρέπει να περιέχει καμμιά στήλη που περιλαμβάνει μόνο μηδενικά στοιχεία γιατί σε τέτοια περίπτωση ενδεχόμενο λάθος στην αντίστοιχη θέση της λαμβανόμενης λέξης δεν θα είχε επίπτωση στο σύνδρομο. Η ιδιότητα αυτή πηγάζει από τη σχέση (2.14).
- Όλες οι στήλες του πίνακα ελέγχου ισοτιμίας πρέπει να είναι μοναδικές. Αν δύο στήλες ήταν ίδιες, λάθη στις δύο αντίστοιχες θέσεις της κωδικής λέξης θα ήταν μη διαχωρίσιμα.

Παράδειγμα 2.4 Έστω γραμμικός συμπαγής κώδικας (6, 3) με πίνακα ελέγχου ισοτιμίας

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

Αν μεταδόθηκε η κωδικοποιημένη λέξη **c** = (1 0 1 1 1 0) και ελήφθη η λέξη **r** = (0 0 1 1 1 0), τότε το σύνδρομο της ληφθείσας λέξης προκύπτει:

$$S = r \circ H^T = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

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

Επιπλέον,

$$e = r \oplus c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Τέλος, επιβεβαιώνουμε ότι

$$S = e \circ H^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

δηλαδή ότι το σύνδρομο του r ταυτίζεται με το σύνδρομο του e.

2.4.5 Διόρθωση Λαθών

Όπως προαναφέρθηκε, μία σημαντική ιδιότητα των γραμμικών συμπαγών κωδίκων, θεμελιώδης στη διαδικασία αποκωδικοποίησης, είναι ότι η αντιστοίχιση μεταξύ διορθώσιμων διανυσμάτων λαθών και συνδρόμων είναι ένα προς ένα. Πρώτο μέλημα λοιπόν για τη διόρθωση λαθών είναι η κατασκευή πίνακα αντιστοίχισης των πιθανών

συνδρόμων με τα αντίστοιχα διανύσματα λαθών, τα οποία ο κώδικας μπορεί να διορθώσει.

Όταν το πλήθος των σφαλμάτων βρίσκεται εντός των δυνατοτήτων διόρθωσης του κώδικα, κάτι για το οποίο θα πούμε περισσότερα όταν παρουσιαστεί στη συνέχεια η έννοια της διορθωτικής ικανότητας ενός κώδικα, ακολουθείται η ακόλουθη αλγοριθμική διαδικασία για τη διόρθωση τους

1. Υπολογίζεται το σύνδρομο της ληφθείσας κωδικοποιημένης λέξης ως $S = r \circ H^T$
2. Από τον πίνακα αντιστοίχισης συνδρόμου και διανύσματος λαθών προκύπτει η εκτίμηση του συστήματος για το διάνυσμα λαθών $\hat{e}$.
3. Από τη σχέση $\hat{c} = r \oplus \hat{e}$, ο αποκωδικοποιητής εξάγει την εκτίμηση του συστήματος για τη μεταδοθείσα κωδική λέξη.

Παράδειγμα 2.5 Έστω κώδικας ίδιος με το προηγούμενο παράδειγμα 2.4. Τότε, από τη σχέση $S = e \circ H^T$ κατασκευάζεται ο πίνακας 2.2 που αντιστοιχεί τα πιθανά σύνδρομα με τα αντίστοιχα διανύσματα λαθών.

Πίνακας 2.2 Αντιστοίχιση διανύσματος λαθών και συνδρόμου για το παράδειγμα 2.5

Διάνυσμα λαθών	Σύνδρομο
000000	000
000001	101
000010	011
000100	110
001000	001
010000	010
100000	100
010001	111

Αν μεταδόθηκε η κωδικοποιημένη λέξη **c** = (1 0 1 1 1 0) και ελήφθη η λέξη **r** = (0 0 1 1 1 0), τότε το σύνδρομο της ληφθείσας λέξης προκύπτει από το προηγούμενο παράδειγμα $S = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$. Χρησιμοποιώντας τον πίνακα 2.2, προκύπτει η εκτίμηση για το αντίστοιχο διάνυσμα λαθών $\hat{e} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$. Το διορθωμένο διάνυσμα βρίσκεται ως

$$\hat{c} = r \oplus \hat{e} = 001110 \oplus 100000 = 101110$$

Καθώς $\hat{c} = c$, ο αποκωδικοποιητής έκανε σωστή διόρθωση λαθών.

2.5 Ικανότητα Εντοπισμού και Διόρθωσης Λαθών

2.5.1 Βάρος και Απόσταση Δυαδικών Διανυσμάτων

Πρέπει να γίνει σαφές ότι δεν μπορούν να διορθωθούν όλα τα λάθη. Για παράδειγμα, σε συνέχεια των όσων αναφέρθηκαν για τους γραμμικούς συμπαγείς κώδικες, αν μεταδοθεί μία κωδική λέξη και φθάσει στον αποκωδικοποιητή μία αλλοιωμένη ως προς την αρχική λέξη που όμως αντιστοιχεί σε μία άλλη κωδική λέξη, τότε το σύνδρομο θα είναι μηδενικό και η ληφθείσα λέξη θα θεωρηθεί σωστή. Αυτό το λάθος όχι μόνο δεν μπορεί να διορθωθεί αλλά ούτε καν να εντοπιστεί. Για να εξετάσουμε σε περισσότερο βάθος την διορθωτική ικανότητα ενός κώδικα θα μας φανούν χρήσιμες οι ένοιες του βάρους και της απόστασης.

Το βάρος w μιας κωδικής λέξης **U** ορίζεται ως ο αριθμός των μη μηδενικών ψηφίων που περιλαμβάνονται στη λέξη. Για ένα δυαδικό διάνυσμα όπως αυτά που μας απασχολούν το βάρος ισούται με τον αριθμό των άσσων που περιέχονται συνολικά. Απόσταση d τώρα μεταξύ δύο κωδικοποιημένων λέξεων ονομάζεται ο αριθμός των θέσεων στις οποίες οι λέξεις διαφέρουν. Οι δύο παραπάνω έννοιες συχνά συναντώνται ως βάρος Hamming και απόσταση Hamming. Παραδείγματος χάριν, έστω δύο κωδικές λέξεις **U**, **V** με

$$U = 1 \quad 0 \quad 0 \quad 1$$
$$V = 0 \quad 1 \quad 1 \quad 1$$

Τότε, $w(U)=2$, $w(V)=3$ και $d(U, V)=3$.

Μέσω των ιδιοτήτων της modulo-2 άθροισης προκύπτει ότι το άθροισμα δύο δυαδικών διανυσμάτων είναι ένα νέο δυαδικό διάνυσμα το οποίο έχει άσσους στις θέσεις που τα διανύσματα διαφέρουν και μηδενικά αλλού. Για παράδειγμα,

$$U \oplus V = 1\ 1\ 1\ 0$$

Έτσι, βλέπουμε ότι η απόσταση Hamming δύο κωδικών λέξεων ισούται με το βάρος Hamming του αθροίσματος τους, δηλαδή $d(\mathbf{U}, \mathbf{V})=w(\mathbf{U} \oplus \mathbf{V})$. Ένα ακόμα συμπέρασμα είναι ότι το βάρος Hamming μίας λέξης ισούται με την απόσταση της λέξης από το μηδενικό διάνυσμα.

2.5.2 Ελάχιστη Απόσταση ενός Γραμμικού Κώδικα

Ελάχιστη απόσταση d_{min} ενός κώδικα ονομάζεται η ελάχιστη απόσταση μεταξύ των κωδικών λέξεων του κώδικα λαμβανομένων ανά δύο με όλους τους δυνατούς τρόπους.

Όπως αναφέρθηκε προηγουμένως, το άθροισμα δύο κωδικών λέξεων είναι επίσης κωδική λέξη. Δηλαδή αν **U**, **V** δύο κωδικές λέξεις τότε και το άθροισμα $\mathbf{U} \oplus \mathbf{V}=\mathbf{W}$ πρέπει να αποτελεί κωδική λέξη. Από την προηγούμενη παράγραφο γνωρίζουμε ότι η απόσταση δύο κωδικών λέξεων ισούται με το βάρος του αθροίσματος τους: $d(\mathbf{U}, \mathbf{V})=w(\mathbf{U} \oplus \mathbf{V})=d(\mathbf{W})$. Έτσι λοιπόν για την εύρεση της ελάχιστης απόστασης δεν είναι απαραίτητο να εξετάσουμε όλα τα ζεύγη κωδικών λέξεων. Αρκεί να εξετάσουμε το

βάρος όλων των μη μηδενικών λέξεων του κώδικα και να υπολογίσουμε την ελάχιστη απόσταση ως το ελάχιστο βάρος αυτών των λέξεων. Το μηδενικό διάνυσμα εξαιρέθηκε από τη διαδικασία καθώς δεν μπορεί να προκύψει ως άθροισμα διαφορετικών διανυσμάτων. Αντίθετα, προκύπτει ως άθροισμα μιας κωδικής λέξης με τον εαυτό της ως $\mathbf{U} \oplus \mathbf{U}=\mathbf{0}$. Όμως η ελάχιστη απόσταση αναφέρεται σε απόσταση μεταξύ διαφορετικών λέξεων και όχι στην απόσταση μιας λέξης από τον εαυτό της που προφανώς και είναι μηδενική.

Η έννοια της ελάχιστης απόστασης εισάγεται για να μας δώσει ένα μέτρο για την ελάχιστη ικανότητα ενός κώδικα και κατ' επέκταση για τη δύναμη του κώδικα. Η ελάχιστη απόσταση παρομοιάζεται με τον αδύναμο κρίκο μιας αλυσίδας[2]. Όπως στην αλυσίδα ο αδύναμος κρίκος και όχι ο πιο δυνατός χαρακτηρίζει τη δύναμη της έτσι και σε ένα κώδικα η ελάχιστη και όχι η μέγιστη απόσταση ποσοτικοποιεί τη δύναμη του.

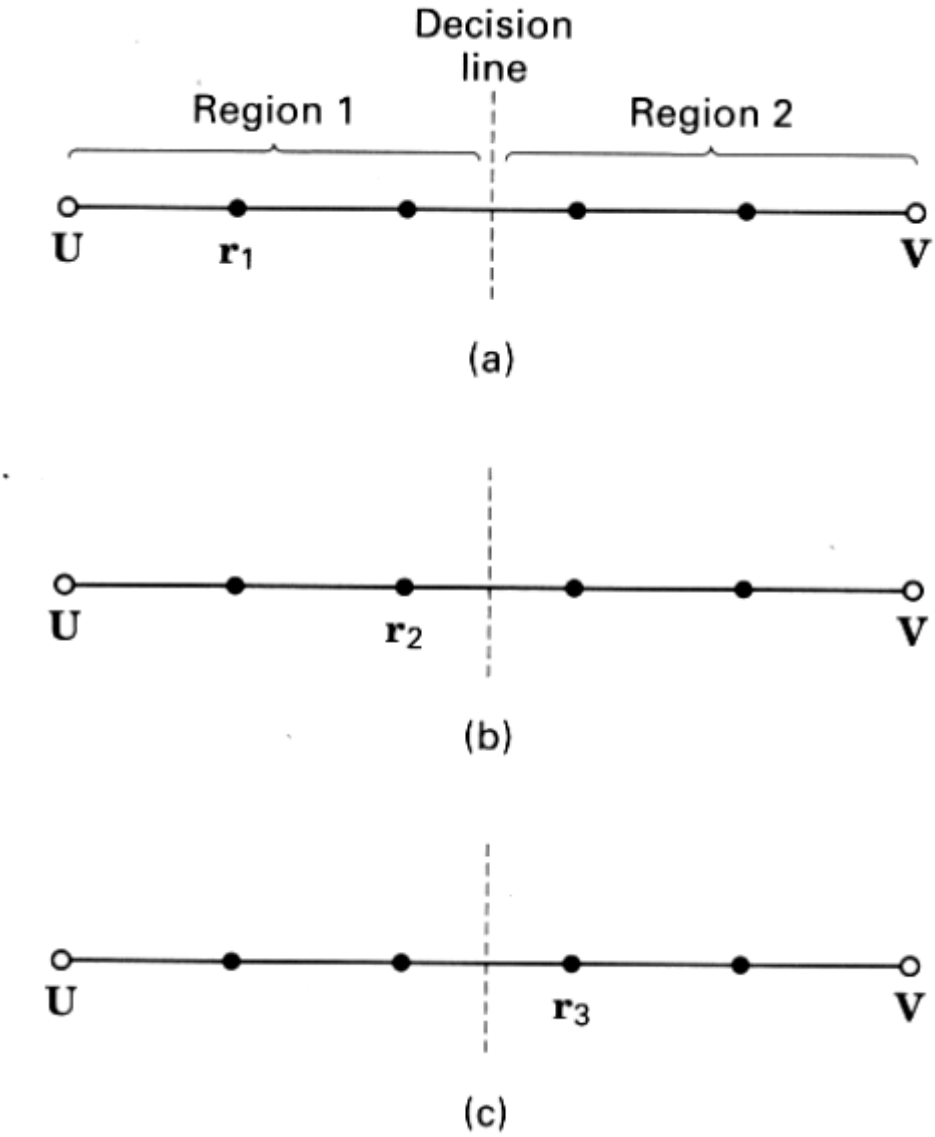
2.5.3 Εντοπισμός και Διόρθωση Λαθών

Ο ρόλος του αποκωδικοποιητή είναι αφού λάβει το κωδικό διάνυσμα **r** να κάνει τη βέλτιστη εκτίμηση για τη μεταδοθείσα κωδική λέξη. Μιλώντας για διακριτό μοντέλου διαύλου, η πιθανότητα εκπομπής μίας κωδικής λέξης **U** δεδομένης της λήψης του διανύσματος **r** είναι αντιστρόφως ανάλογη με την απόσταση ανάμεσα στα διανύσματα **r**, **U.** Συνεπώς, αυτό που ονομάσαμε βέλτιστη εκτίμηση έγκειται στην εύρεση από τον αποκωδικοποιητή της κωδικής λέξης με την ελάχιστη απόσταση από το διάνυσμα **r**.

Ας εξετάσουμε τώρα το σχήμα 2.9 όπου η απόσταση μεταξύ των κωδικών λέξεων **U**, **V** δείχνεται ως απόσταση Hamming. Κάθε μαύρη τελεία αναπαριστά μία αλλοιωμένη κωδική λέξη. Το σχήμα 2.9a αναπαριστά τη λήψη του διανύσματος $\mathbf{r}_1$ που απέχει 1 κατά Hamming από την κωδική λέξη U και 4 από την κωδική λέξη V. Όπως εξηγήσαμε προηγουμένως, ένας αποκωδικοποιητής βέλτιστης πιθανοτικής εκτίμησης θα διαλέξει το διάνυσμα **U** σε σύγκριση με το **V** καθώς απέχει λιγότερο από το. Αν τώρα το $\mathbf{r}_1$ προέκυψε ως παραμόρφωση του **U** κατά ένα bit, τότε ο αποκωδικοποιητής διόρθωσε επιτυχώς το σφάλμα. Όμως εάν το $\mathbf{r}_1$ είναι αποτέλεσμα παραμόρφωσης του **V** κατά τέσσερα bits τότε έχουμε σφάλμα στη διαδικασία αποκωδικοποίησης. Ομοίως, διπλό λάθος στη μετάδοση του U οδηγεί στο $\mathbf{r}_2$ που απέχει δύο από το U και 3 από το V, όπως φαίνεται στο σχήμα 2.9b. Και σε αυτή την περίπτωση, η επιλογή του αποκωδικοποιητή θα είναι το διάνυσμα U. Συνεχίζοντας, δείχνουμε στο σχήμα 2.9c την περίπτωση τριπλού λάθους κατά τη μετάδοση του U που οδηγεί στη λήψη του διανύσματος $\mathbf{r}_3$. Σε αυτή την περίπτωση και με την ίδια στρατηγική, ο αποκωδικοποιητής θα διαλέξει την κωδική λέξη V πραγματοποιώντας μια λανθασμένη εκτίμηση. Από το σχήμα καταλαβαίνουμε ότι αν στόχος είνα μόνο ο εντοπισμός και όχι η διόρθωση λαθών (ARQ), τότε λάθη σε 1, 2, 3 ή 4 bits της μεταδοθείσας κωδικής λέξης είναι εντοπίσιμα. Αντίθετα, 5 εσφαλμένα bits μπορεί να οδηγήσουν στη λήψη της κωδικής λέξης V ενώ είχε αποσταλεί η κωδική λέξη V. Τέτοιας μορφής σφάλμα είναι μη εντοπίσιμο.

Από το σχήμα 2.9 εξάγεται το συμπέρασμα ότι η ικανότητα εντοπισμού λαθών και η ικανότητα διόρθωσης λαθών ενός κώδικα εξαρτώνται από την ελάχιστη απόσταση του κώδικα. Στο παράδειγμα μας, όπου το κριτήριο απόφασης για την επιλογή της κωδικής λέξης U είναι αν το r βρίσκεται στην περιοχή 1 (Region 1) (ομοίως για την V αν το r βρίσκεται στην περιοχή 2), βλέπουμε ότι ένας τέτοιος κώδικας με ελάχιστη

απόσταση 5 μπορεί να διορθώσει μέχρι 2 λάθη. Το παραπάνω συμπέρασμα μπορεί αν γενικευτεί.



Σχήμα 2.9 Σχετικά με τη διορθωτική ικανότητα (a) Λήψη $\mathbf{r}_1$. (b) Λήψη $\mathbf{r}_2$. (c) Λήψη $\mathbf{r}_3$

Έτσι, ένας κώδικας με ελάχιστη απόσταση d_{min} μπορεί εγγυημένα να διορθώνει μέχρι και t λάθη ανά κωδική λέξη όπου

$$t = \left[\frac{d_{\min} - 1}{2} \right] \tag{2.15}$$

όπου με [x] συμβολίζουμε το ακέραιο μέρος του x.

Όσον αφορά τώρα τον εντοπισμό λαθών, από την ανάλυση που έγινε προκύπτει ότι ένας κώδικας μπορεί εγγυημένα να ανιχνεύει το σφάλμα αν τα λάθη ανά κωδική λέξη είναι μικρότερα ή ίσα από την τιμή

$$e = d_{\min} - 1 \tag{2.16}$$

2.6 Αξιολόγηση Συστημάτων Κωδικοποίησης

2.6.1 Χωρητικότητα Διαύλου - Θεώρημα Shannon

Το 1948 ο Shannon απέδειξε ότι σε διαύλους που χαρακτηρίζονται από λευκό προσθετικό θόρυβο, τα λάθη που εισάγονται κατά τη μετάδοση μπορεί να μειωθούν με κωδικοποίηση της ψηφιακής πληροφορίας χωρίς την ανάγκη αύξησης του ρυθμού μετάδοσης πληροφορίας R_i. Η προϋπόθεση για την ισχύ της παραπάνω πρότασης είναι ότι ο ρυθμός R_i δεν πρέπει να υπερβαίνει τη χωρητικότητα C του διαύλου η οποία βρίσκεται ως

$$C = B\log_2\left(1+\frac{S}{N}\right) \tag{2.17}$$

όπου B το εύρος ζώνης συχνοτήτων του διαύλου και S/N ο σηματοθορυβικός λόγος (Signal-to-noise ratio, SNR) στην είσοδο του δέκτη. Για το σηματοθορυβικό λόγο ισχύει

$$\frac{S}{N} = \frac{E_b}{n_0}\cdot\frac{R_i}{B} \tag{2.18}$$

όπου E_b η ενέργεια ανά bit και n_0 η μονόπλευρη φασματική πυκνότητα θορύβου. Λόγω της (2.18) τώρα, η σχέση (2.17) γράφεται ως

$$C = B\log_2\left(1+\frac{R_i\cdot E_b}{n_0\cdot B}\right) \tag{2.19}$$

Ισοδύναμα,

$$C = R_i\frac{E_b}{n_0}\left(z\log_2\left(1+\frac{1}{z}\right)\right) \tag{2.20}$$

με

$$z = \left(\frac{E_b}{n_0}\right)^{-1}\cdot\left(\frac{B}{R_i}\right) \tag{2.21}$$

Καθώς

$$\lim_{z\to\infty} z\log_2\left(1+\frac{1}{z}\right)=\log_2 e\cong 1.443 \tag{2.22}$$

λόγω της (2.20) παίρνουμε

$$\lim_{B\to\infty} C = 1.443R_i(E_b/n_0) \tag{2.23}$$

Επειδή ο ρυθμός μετάδοσης R_i δεν πρέπει να υπερβαίνει τη χωρητικότητα C

$$R_i \le \lim_{B\to\infty} C = 1.443R_i(E_b/n_0) \tag{2.24}$$

από όπου προκύπτει

$$\left(E_b/n_0\right) \ge 0.693 = -1.6dB \tag{2.25}$$

Η παραπάνω σχέση εκφράζει ένα σημαντικό θεωρητικό όριο για τις τηλεπικοινωνίες. Θέτει ένα ελάχιστο όριο για την τιμή του λόγου $\left(E_b/n_0\right)$, πάνω από το οποίο μπορούμε να επιτύχουμε θεωρητικά όσο μικρή πιθανότητα λάθους θέλουμε μέσω κατάλληλου σχήματος κωδικοποίησης.

2.6.2 Κέρδος Κωδικοποίησης

Το κέρδος κωδικοποίησης ορίζεται ως

$$G_c = \frac{\left(E_b/n_o\right)_u}{\left(E_b/n_o\right)_c} \tag{2.26}$$

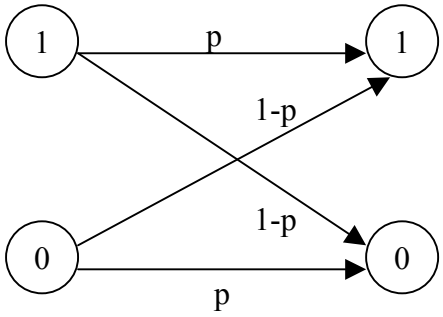
όπου $\left(E_b/n_o\right)_u$ και $\left(E_b/n_o\right)_c$ είναι οι λόγοι ενέργειας bit προς πυκνότητα θορύβου χωρίς ή με κωδικοποίηση αντίστοιχα, έτσι ώστε το ποσοστό λαθών να είναι το ίδιο και στις δύο περιπτώσεις. Η παράμετρος που ορίστηκε μέσω της σχέσης (2.26) και που ονομάστηκε κέρδος κωδικοποίησης ουσιαστικά ποσοτικοποιεί το πλεονέκτημα της χρήσης κωδικοποίησης σε όρους επιτρεπτής μείωσης του σηματοθορυβικού λόγου.
Γνωρίζουμε ότι για διαύλους που χαρακτηρίζονται από λευκό προσθετικό θόρυβο, το κέρδος κωδικοποίησης δίνεται προσεγγιστικά ως

$$G_c = 10\log_2\left(\frac{k}{n}d_{\min}\right) \tag{2.27}$$

Βλέπουμε ότι το κέρδος κωδικοποίησης αυξάνεται με την αύξηση της ελάχιστης απόστασης $d_{\min}$, κάτι που έρχεται ως συνέχεια της ανάλυσης που έγινε στην ενότητα 2.5.

2.6.3 Διακριτό Μοντέλο Διαύλου

Στην περίπτωση δυαδικού αλφάβητου που μας απασχολεί παρουσιάζεται το διακριτό μοντέλο διαύλου που φαίνεται στο σχήμα 2.10. Λόγω συμμετρίας, η πιθανότητα μετάβασης, ύστερα από τη μετάδοση, από 1 σε 0 είναι ίδια με την πιθανότητα μετάβασης από 0 σε 1 και συμβολίζεται (1-p). Με p συμβολίζεται η πιθανότητα μη αλλοίωσης της κατάστασης του μεταδιδόμενου ψηφίου. Σε συστήματα που δεν χρησιμοποίείται κωδικοποίηση η πιθανότητα (1-p) εκφράζει τη μέση πιθανότητα λάθους ανά μεταδιδόμενο ψηφίο που θα συμβολίζεται P_b.



Σχήμα 2.10 Διακριτό μοντέλου διαύλου

2.6.4 Πιθανότητα λάθους στους γραμμικούς συμπαγείς κώδικες

Θεωρούμε γραμμικό συμπαγή κώδικα (n, k) κατά τα γνωστά με ικανότητα διόρθωσης t λαθών ανά κωδικοποιημένη λέξη. Στόχος μας είναι η θεωρητική σύγκριση της πιθανότητας λάθους χωρίς και με κωδικοποίηση για διόρθωση λαθών. Για την εγκυρότητα της σύγκρισης υποθέτουμε ότι ο ρυθμός μετάδοσης πληροφορίας R_i και η ισχύς εκπομπής C μένουν αμετάβλητες. Συμβολίζοντας με E_b την ενέργεια ψηφίου στην περίπτωση μη χρήσης κωδικοποίησης και με E_c την ενέργεια ψηφίου σε περίπτωση χρήσης κωδικοποίησης παίρνουμε

$$E_b = \frac{C}{R_i} \tag{2.28}$$

και

$$E_c = \frac{C}{R_c} \tag{2.29}$$

όπου R_c είναι ο ρυθμός μετάδοσης ψηφίων σε περίπτωση χρήσης κωδικοποίησης. Από τα εγγενή χαρακτηριστικά του κώδικα, προκύπτει ότι σε n εκπεμπόμενα ψηφία τα k αντιστοιχούν σε μετάδοση ψηφίων πληροφορίας. Από τη στιγμή που υποθέσαμε ότι ο ρυθμός μετάδοσης πληροφορίας μένει αμετάβλητος πρέπει να ισχύει

$$R_c = \frac{n}{k} R_i \tag{2.30}$$

Τότε,

$$E_c = \frac{C}{R_c} = \frac{C}{nR_i / k} = \frac{k}{n} E_b < E_b \tag{2.31}$$

Όταν δεν χρησιμοποιείται κωδικοποίηση, λάθος ανά λέξη έχουμε όταν τουλάχιστον ένα ψηφίο της λέξης ληφθεί λανθασμένα. Έτσι, η πιθανότητα λάθους ανά λέξη προκύπτει ως συμπληρωματική πιθανότητα του ενδεχόμενου όλα τα μεταδιδόμενα ψηφία να ληφθούν δίχως σφάλμα. Χρησιμοποιώντας τον ίδιο συμβολισμό που εισήχθη στο διακριτό μοντέλο διαύλου παίρνουμε για την πιθανότητα λάθους ανά μη κωδικοποιημένη λέξη:

$$P_e = 1 - p^k \tag{2.32}$$

Με χρήση κωδικοποίησης τώρα, η πιθανότητα λάθους ανά κωδικοποιημένη λέξη P_{ec} φράσσεται από το όριο

$$P_{ec} \le \sum_{i=t+1}^{n} \binom{n}{i} p_c^{\,n-i} (1-p_c)^i \tag{2.33}$$

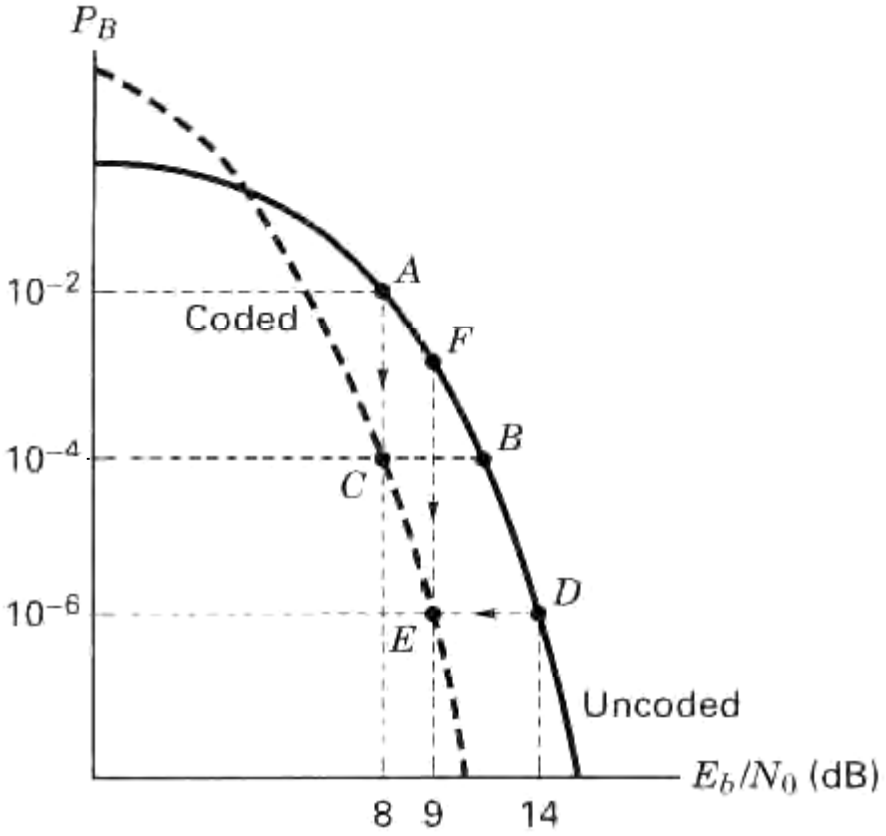
που αντιστοιχεί στην πιθανότητα να γίνουν περισσότερα από t λάθη στα n ψηφία της λέξης. Με $(1-p_c)$ συμβολίζουμε την πιθανότητα εσφαλμένης μετάβασης ψηφίου όταν χρησιμοποιείται κωδικοποίηση.
Αν $n(1-p_c) \ll 1$ προσεγγίζουμε το παραπάνω άθροισμα κρατώντας μόνο τον πρώτο όρο που είναι σημαντικά μεγαλύτερος και παίρνουμε

$$P_{ec} \le \binom{n}{t+1} p_c^{\,n-t-1} \cdot (1-p_c)^{t+1} \approx \binom{n}{t+1} (1-p_c)^{t+1} \tag{2.34}$$

2.6.5 Πλεονεκτήματα και μειονεκτήματα από τη χρήση Κωδικοποίησης για Διόρθωση Λαθών

Όπως έχει ήδη επισημανθεί, η χρήση κωδικοποίησης για διόρθωση λαθών (FEC) είναι αναπόσπαστο κομμάτι για τα σύγχρονα τηλεπικοινωνιακά συστήματα. Η ανάγκη χρήσης FEC είναι ακόμα πιο επιτακτική στο χώρο των ασυρμάτων επικοινωνιών. Εδώ, ξεχωρίζουμε τις κινητές επικοινωνίες που επιβαρύνονται από φαινόμενα πολυδιαδρομικής διάδοσης του σήματος και τις δορυφορικές επικοινωνίες που επιβαρύνονται από τις ιδιαιτερότητες που εμφανίζει η διάδοση ηλεκτρομαγνητικών κυμάτων στην ατμόσφαιρα.

Στόχος αυτής της ενότητας είναι ένας γενικός σχολιασμός πάνω στο ισοζύγιο κερδών-απωλειών από τη χρήση κωδικοποίησης για διόρθωση λαθών. Η σύγκριση βέβαια θα γίνει με την περίπτωση μη χρησιμοποίησης κάποιου σχήματος κωδικοποίησης. Η τυπική καμπύλη που συγκρίνει την επίδοση συστήματος με ή χωρίς κωδικοποίηση (Coded vs Uncoded) δίνεται στο σχήμα 2.11. Η επίδοση του συστήματος εκφράζεται σε όρους πιθανότητας λάθους ανά μεταδιδόμενο ψηφίο (P_b).



Σχήμα 2.11 Τυπική σύγκριση επίδοσης χωρίς και με κωδικοποίηση

Με βάση το σχήμα 2.11 μπορούμε να εξάγουμε κάποια πλεονεκτήματα από τη χρήση κωδικοποίησης (διακεκκομένη γραμμή στο σχήμα) τα οποία όμως έχουν και το κόστος τους (trade-off).

2.6.5.1 Ισοζύγιο Επίδοσης-Εύρους Ζώνης

Έστω τηλεπικοινωνιακό σύστημα που δεν χρησιμοποιεί κωδικοποίηση και το σημείο λειτουργίας του είναι το σημείο A (E_b/ N_0 =8dB, P_b =10^{-2}) της καμπύλης 2.11. Έστω τώρα ότι απαιτείται βελτίωση της επίδοσης του συστήματος σε όρους πιθανότητας λάθους στην τιμή 10^{-4}. Για να γίνει κάτι τέτοιο στο υπάρχον σύστημα, θα έπρεπε το σημείο λειτουργίας να μετατοπιστεί στο σημείο B της καμπύλης. Όπως φαίνεται, αυτή η μετατόπιση προϋποθέτει αύξηση της τιμής του λόγου E_b/ N_0 κάτι που δεν είναι πάντα εφικτό κυρίως λόγω της περιορισμένης ισχύος των πομποδεκτών. Σε τέτοια περίπτωση η κωδικοποίηση μπορεί να δώσει λύση αφού από την αντίστοιχη καμπύλη (Coded) φαίνεται ότι το σημείο C καλύπτει τις προδιαγραφές επίδοσης χωρίς να απαιτεί αύξηση του σηματοθορυβικού λόγου. Το κόστος αυτής της ποιοτικής βελτίωσης είναι η αύξηση του χρησιμοποιούμενου εύρους ζώνης συχνοτήτων. Αυτό γιατί η κωδικοποίηση χρησιμοποιεί πλεονάζοντα δεδομένα και για να διατηρείται σταθερός ο ρυθμός μετάδοσης πληροφορίας θα πρέπει ο ολικός ρυθμός μετάδοσης R_c να αυξηθεί όπως υπαγορεύεται από τη σχέση 2.30.

$$R_c = \frac{n}{k} R_i$$

Αυτή η αύξηση του ολικού ρυθμού μετάδοσης απαιτεί δέσμευση μεγαλύτερου εύρους ζώνης.

2.6.5.2 Ισοζύγιο Ισχύος-Εύρους Ζώνης

Έστω τώρα το σημείο λειτουργίας D (E_b/ N_0 =14dB, P_b =10^{-6}) της καμπύλης 2.11. Η επίδοση είναι ικανοποιητική όμως ο σηματοθορυβικός λόγος θα θέλαμε να μειωθεί. Με χρήση κωδικοποίησης υπάρχει η δυνατότητα μετατόπισης του σημείου λειτουργίας στο σημείο E όπου διατηρώντας τα ίδια επίπεδα ποιότητας-επίδοσης μειώνουμε τον απαιτούμενο λόγο E_b/ N_0 δημιουργώντας έτσι κέρδος κωδικοποίησης όπως αυτό ορίστηκε μέσω της σχέσης (2.26) . Το κόστος σε αυτή την περίπτωση είναι το ίδιο με προηγουμένως : κατανάλωση περισσότερου εύρους ζώνης.

Σημειώνουμε ότι η απαίτηση για σταθερό ρυθμό μετάδοσης πληροφορίας που προκαλεί και την αύξηση του εύρους ζώνης αφορά τηλεπικοινωνιακά συστήματα πραγματικού χρόνου (real-time) όπως είναι η τηλεφωνία. Σε συστήματα μη πραγματικού χρόνου το κόστος θα ήταν η χρονοκαθυστέρηση αντί του εύρος ζώνης.

2.6.5.3 Ισοζύγιο Ρυθμού Μετάδοσης-Εύρους Ζώνης

Η τιμή του σηματοθορυβικού λόγου στο δέκτη (SNR) δίνεται από τη σχέση

$$\frac{S}{N} = \frac{E_b \cdot R}{N_0 W} \tag{2.35}$$

με S τη λαμβανόμενη ισχύ, R το ρυθμό μετάδοσης (bits/sec) και W το εύρος ζώνης του συστήματος.

Ισοδύναμα,

$$\frac{E_b}{N_0} = \frac{S}{N_o}\left(\frac{1}{R}\right) \tag{2.36}$$

Έστω σημείο λειτουργίας D και έστω ότι επιθυμούμε αύξηση του ρυθμού μετάδοσης R. Κάτι τέτοιο όπως προκύπτει από την παραπάνω σχέση συνεπάγεται μείωση του λόγου E_b/ N_0 και μετατόπιση του σημείου λειτουργίας στα αριστερά της καμπύλης, έστω στο σημείο F όπου έχουμε μείωση της επιδοσης. Με χρήση κωδικοποίησης μπορούμε να μεταφέρουμε το σύστημα στο σημείο Ε όπου κρατώντας σταθερή την ποιότητα υπηρεσίας επιτυγχάνουμε και την επιθυμητή αύξηση του ρυθμού μετάδοσης. Και πάλι το κόστος είναι το αυξημένο εύρος ζώνης.

3 Προσομοίωση Τεχνικών Κωδικοποίησης για την Εκτίμηση της Επίδοσής τους

3.1 Γενικά περί Προσομοίωσης

Ο ρόλος της προσομοίωσης στα σύγχρονα τηλεπικοινωνιακά συστήματα γίνεται ολοένα και πιο σημαντικός. Η εξέλιξη αυτή αποδίδεται σε διάφορους παράγοντες. Πρώτον, η αυξανόμενη πολυπλοκότητα των συστημάτων επικοινωνιών σε επίπεδο αρχιτεκτονικής καθιστά πολύ δύσκολη την εύρεση αναλυτικών εκφράσεων για την επίδοση τους. Επίσης, αναλυτικές εκφράσεις είναι πολύ δύσκολο να έχουμε για τη μοντελοποίηση του τηλεπικοινωνιακού διαύλου. Για παράδειγμα, στο πεδίο των κινητών επικοινωνιών έχουμε περιβάλλοντα που οδηγούν σε έντονες διαλείψεις στο δέκτη κάτι που είναι δύσκολο να μοντελοποιηθεί μέσω αναλυτικών σχέσεων.

Λύση στα παραπάνω ζητήματα μπορεί να δώσει με αρκετή επιτυχία η προσομοίωση. Με κύριο αρωγό την ανάπτυξη ισχυρών υπολογιστών, μπορεί να γίνει μελέτη του συστήματος ύστερα από μια διαδικασία που περιλαμβάνει πολύ μεγάλο αριθμό επαναλήψεων που κάθε μία κάνει μία εκτίμηση για την επίδοση του συστήματος. Αυτός ο μεγάλος αριθμός επαναλήψεων, που βέβαια θα ήταν αδύνατος δίχως τη χρήση υπολογιστικών συστημάτων, μας επιτρέπει να εξάγουμε μια αξιόπιστη εικόνα για το σύστημα. Παράλληλα, η ανάγκη για αποτελεσματική προσομοίωση έχει οδηγήσει στην ανάπτυξη ισχυρών πακέτων λογισμικού ενώ υπάρχει και σημαντική κινητικότητα σε αυτό που ονομάζεται ‘Θεωρία Προσομοίωσης’. Ένα ακόμα θετικό στοιχείο είναι το ελάχιστο κόστος μιας τέτοιας διαδικασίας.

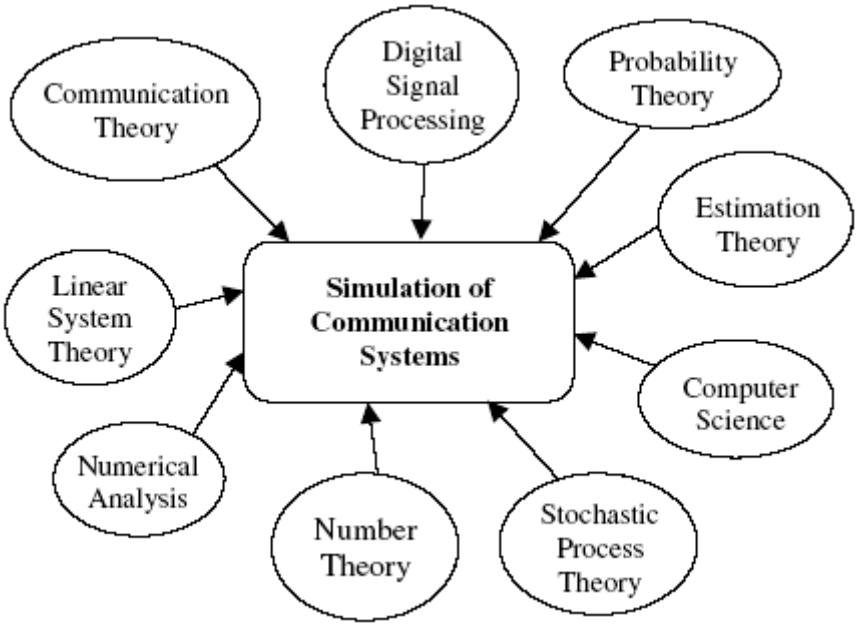
Βλέπουμε λοιπόν ότι η πρόοδος της προσομοίωσης οφείλεται κατά κύριο λόγο στην εξέλιξη δύο μεγάλων πεδίων: των τηλεπικοινωνιών και των υπολογιστών. Έτσι, περιοχές που ανήκουν σε αυτά τα μεγάλα πεδία, όπως θεωρία επικοινωνιών και αριθμητική ανάλυση, συμβάλλουν στην ανάπτυξη της προσομοίωσης. Στο σχήμα 3.1 φαίνονται οι κύριες επιδρώσες περιοχές στη μελέτη και την ανάπτυξη της προσομοίωσης τηλεπικοινωνιακών συστημάτων. Στη συνέχεια, θα γίνει μια σύντομη αναφορά στο ρόλο κάθε μίας από τις εννέα εικονιζόμενες περιοχές.

Η θεωρία γραμμικών συστημάτων (linear system theory) μας δίνει τις απαραίτητες γνώσεις για την περιγραφή της σχέσης εισόδου-εξόδου γραμμικών συστημάτων. Έτσι, κατά την προσομοίωση μπορούμε μέσω αυτών των σχέσεων να μοντελοποιήσουμε είτε στο πεδίο του χρόνου είτε στο πεδίο της συχνότητας τη λειτουργία ενός συστατικού ή ενός υποσυστήματος του όλου συστήματος.

Η θεωρία επικοινωνιών (communication theory) σαφώς και αποτελεί κομμάτι της όλης διαδικασίας. Η αρχιτεκτονική του συστήματος επικοινωνιών, τα λειτουργικά χαρακτηριστικά υποσυστημάτων όπως κωδικοποιητές και ο τηλεπικοινωνιακός δίαυλος είναι στοιχεία που οπωσδήποτε πρέπει να έχουμε γνώση τους ούτως ώστε σωστή μοντελοποίηση τους να οδηγήσει και σε αξιόπιστα συμπεράσματα μετά το πέρας της προσομοίωσης. Αξίζει να σημειώσουμε ότι αυτό που θεωρείται γενικά πιο δύσκολο να μοντελοποιηθεί είναι τα χαρακτηριστικά του τηλεπικοινωνιακού διαύλου ειδικά μάλιστα σε περίπτωση που η ζεύξη υποφέρει από διαλείψεις, φαινόμενο συχνό όπως είπαμε σε περιβάλλοντα κινητών επικοινωνιών.

Η θεωρία επεξεργασίας ψηφιακών σημάτων (digital signal processing) συνεισφέρει κυρίως στην ανάπτυξη προσεγγίσεων διακριτού χρόνου για συστήματα συνεχούς

χρόνου, όπως φίλτρα. Αυτό πρέπει να γίνει λόγω της εκ φύσεως διακριτού χρόνου λειτουργίας της προσομοίωσης.



Σχήμα 3.1 Επιστημονικές Περιοχές σχετικές με την Προσομοίωση

Η αριθμητική ανάλυση (numerical analysis) τώρα είναι συνυφασμένη με την ανάπτυξη αποδοτικών υπολογιστικών τεχνικών. Πολύπλοκα ολοκληρώματα, πολύπλοκες λύσεις διαφορικών εξισώσεων μπορούν να προσεγγιστούν με αποδοτικό τρόπο και συνήθως μέσω επαναληπτικών διαδικασιών χρησιμοποιώντας τεχνικές αριθμητικής ανάλυσης.

Η θεωρία πιθανοτήτων (probability theory) έχει και αυτή τον ιδιαίτερο ρόλο της στην προσομοίωση τηλεπικοινωνιακών συστημάτων. Πρώτο απ' όλα, πολύ συχνά η επίδοση τηλεπικοινωνιακών συστημάτων εκφράζεται με πιθανοτικούς όρους, π.χ. απαιτείται η πιθανότητα λάθους να είναι μικρότερη από 0.001 για 98% του χρόνου λειτουργίας. Επίσης, το αποτέλεσμα μιας προσομοίωσης είναι συνήθως μια τυχαία μεταβλητή και πρέπει να μελετηθεί η τυπική απόκλιση της προκειμένου να χαρακτηριστεί η επιτυχία ή όχι της προσομοίωσης. Μεγάλη τυπική απόκλιση σημαίνει αστάθεια στα αποτελέσματα της προσομοίωσης κάτι που δεν θεωρείται θετικό. Στο ίδιο πεδίο, βρίσκει εφαρμογή και η θεωρία στοχαστικών διαδικασιών (stochastic process theory). Για παράδειγμα, ο λευκός προσθετικός θόρυβος Gauss είναι μία στοχαστική διαδικασία και εμείς στην προσομοίωση παράγουμε κάθε φορά ένα δείγμα αυτής της στοχαστικής διαδικασίας που εκφράζεται εν προκειμένω μέσω συνάρτησης πυκνότητας πιθανότητας.

Η θεωρία αριθμών (number theory) μπορεί να βοηθήσει στην δημιουργία τυχαίων ακολουθιών. Για παράδειγμα, στην προσομοίωση μας επιθυμούμε η είσοδος στον κωδικοποιητή να είναι μια τυχαία ακολουθία αποτελούμενη από 0 ή 1.

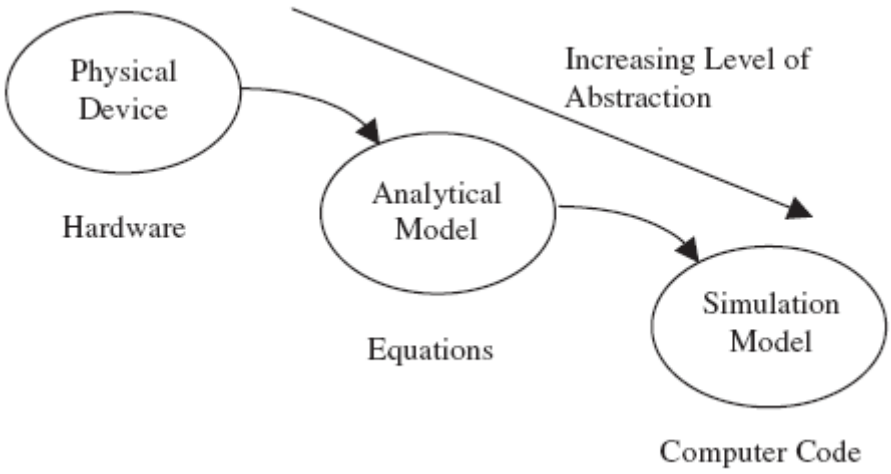
Κάποια βασικά στοιχεία της επιστήμης υπολογιστών (computer science) βρίσκουν εφαρμογή επίσης. Η επιλογή της κατάλληλης γλώσσας προγραμματισμού για κάθε προσομοίωση καθώς και τεχνικές για τη δημιουργία αποδοτικών αλγορίθμων που

μπορούν να ελαττώσουν το συνολικό χρόνο προσομοίωσης αλλά και την απαιτούμενη υπολογιστική μνήμη είναι μερικά από αυτά.

Τέλος, η θεωρία εκτίμησης (estimation theory) μας δίνει εργαλεία για την εκτίμηση ενός προσομοιωτικού αποτελέσματος. Κάτι τέτοιο είναι απαραίτητο γιατί όπως προαναφέρθηκε το αποτέλεσμα μιας προσομοίωσης αντιμετωπίζεται ως τυχαία μεταβλητή. Με σχετικούς όρους, η εκτίμηση για να θεωρείται επιτυχής πρέπει να είναι συνεπής (consistent) και αμερόληπτη (unbiased).

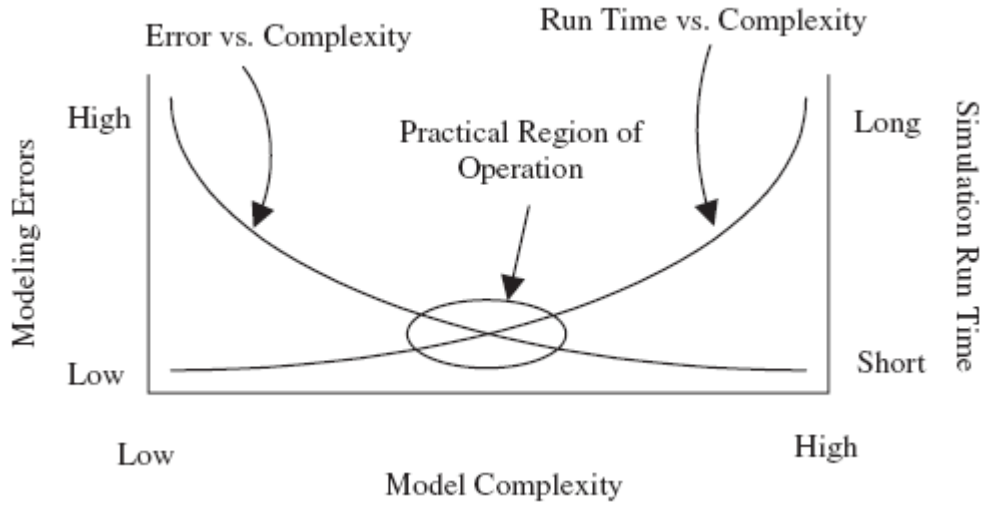
3.2 Μοντελοποίηση

Το πρώτο βήμα για τη δημιουργία μιας προσομοίωσης ενός συστήματος είναι η μοντελοποίηση του. Η μοντελοποίηση περιλαμβάνει κατά χρονική σειρά ανάπτυξης δύο ειδών μοντέλα: τα αναλυτικά μοντέλα και τα προσομοιωτικά μοντέλα. Η διαδικασία ανάπτυξης τους φαίνεται καλύτερα στο σχήμα 3.2



Σχήμα 3.2 Διαδικασία ανάπτυξης μοντέλων

Όπως φαίνεται, ξεκινάμε πάντα από τη μελέτη ενός πραγματικού στοιχείου. Αυτό μπορεί να είναι απλά ένα φίλτρο π.χ., μπορεί όμως να είναι και ένα πλήρες τηλεπικοινωνιακό σύστημα. Η μελέτη του μας οδηγεί στην ανάπτυξη ενός αναλυτικού μοντέλου που το περιγράφει και που εκφράζεται συνήθως μέσω εξισώσεων. Αυτές οι παραγόμενες εξισώσεις είναι μια αφαιρετική περιγραφή του προς μελέτη στοιχείου και εμπεριέχουν αρκετές υποθέσεις και προσεγγίσεις. Στη συνέχεια, ακόμα ένα αφαιρετικό επίπεδο πιο πάνω, το αναλυτικό μοντέλο μετατρέπεται σε προσομοιωτικό μοντέλο που θα εκφραστεί τελικά ως κώδικας μιας γλώσσας προγγραματισμού. Και σε αυτή τη μετάβαση γίνονται αρκετές υποθέσεις και προσεγγίσεις. Ένας πιθανός λόγος είναι ότι ένα ακριβές αναλυτικό μοντέλο ενδέχεται να οδηγήσει σε πολύ μεγάλο χρόνο εκτέλεσης της προσομοίωσης λόγω της πολυπλοκότητας του, κάτι που επιτάσσει ελάττωση της πολυπλοκότητας μέσω προσεγγίσεων. Τέτοιου είδους ισοζύγια (trade-offs) απαντώνται στις περισσότερες προσομοιώσεις και φαίνονται συνοπτικά στο σχήμα 3.3. Σε κάθε περίπτωση πάντως , πρέπει παρά τις όποιες απλοποιήσεις να εξασφαλίζεται ότι το μοντέλο προσομοίωσης είναι έγκυρο και τα αποτελέσματά του αντικατοπτρίζουν την πραγματικότητα



Σχήμα 3.3 Ακρίβεια αποτελεσμάτων και χρόνος εκτέλεσης συναρτήσει της πολυπλοκότητας του μοντέλου

3.3 Ντετερμινιστικές και Στοχαστικές Προσομοιώσεις

Υπάρχουν βασικά δύο είδη προσομοιώσεων: οι ντετερμινιστικές και οι στοχαστικές. Με τον όρο ντετερμινιστικές αναφερόμαστε στις προσομοιώσεις (και γενικότερα στις διαδικασίες) που έχουν προδιαγεγραμμένο αποτέλεσμα. Ένα παράδειγμα είναι η προσομοίωση ηλεκτρικού κυκλώματος αποτελούμενο από γνωστές πηγές τάσης σταθερής τιμής και γνωστές αντιστάσεις επίσης σταθερής τιμής. Το αποτέλεσμα θα είναι και πρέπει να είναι πάντα το ίδιο. Το ίδιο αποτέλεσμα θα προέκυπτε και αν επιλύαμε το κύκλωμα με κλασσικές μεθόδους (χαρτί και μολύβι). Ο λόγος που μπορεί να προτιμήσουμε την προσομοίωση σε υπολογιστή είναι η αποφυγή χρονοβόρων υπολογιστών πράξεων.

Από την άλλη πλευρά υπάρχουν και διαδικασίες που δεν έχουν προκαθορισμένα αποτελέσματα καθώς εμπεριέχουν και τον παράγοντα της τυχαιότητας. Τα αποτέλεσματα αυτών των διαδικασιών αντιμετωπίζονται ως τυχαίες μεταβλητές. Τέτοιες διαδικασίες και τέτοιες προσομοιώσεις ονομάζονται στοχαστικές. Ένα παράδειγμα, άμεσα σχετιζόμενο με την προσομοίωση που θα εκτελέσουμε στη συνέχεια της εργασίας, είναι ένα ψηφιακό τηλεπικοινωνιακό σύστημα στο οποίο το λαμβανόμενο σήμα προκύπτει ως άθροισμα του μεταδιδόμενου σήματος και τυχαίου θορύβου. Έστω ότι θέλουμε να υπολογίσουμε την πιθανότητα λάθους ανά σύμβολο. Γνωρίζουμε ότι αν χρησιμοποιείται BPSK διαμόρφωση και ο δίαυλος είναι τύπου AWGN αυτή η πιθανότητα εκφράζεται ως

$$P_E = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{3.1}$$

όπου Q(x) η συνάρτηση Gauss που ορίζεται ως

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(\frac{-y^2}{2} \right) dy \tag{3.2}$$

Το ζήτημα που προκύπτει σε σχέση με την προσομοίωση είναι ότι η παραπάνω πιθανότητα έχει προκύψει θεωρητικά μελετώντας άπειρο αριθμό συμβόλων. Είναι προφανές όμως ότι σε μια προσομοίωση μπορεί να προσομοιωθεί ένας πεπερασμένος αριθμός συμβόλων (η έννοια σύμβολο είναι γενική, στην περίπτωση μας αναφερόμαστε σε bits).
Η κοινή πρακτική σε τέτοιες περιπτώσεις είναι να προσομοιώσουμε ένα μεγάλο αριθμό bits και να μετρήσουμε τα λάθη στην έξοδο του δέκτη. Εάν μεταδώσουμε N ψηφία και παρατηρηθούν N_e λάθη, η εκτίμηση για την πιθανότητα λάθους είναι

$$\hat{P}_E = \frac{N_e}{N} \tag{3.3}$$

Αυτός ο ρυθμός λαθών συναντάται ως BER (Bit error rate). Μέσω του ορισμού τώρα της πιθανότητας χρησιμοποιώντας σχετική συχνότητα προκύπτει

$$P_E = \lim_{N \to \infty} \frac{N_e}{N} \tag{3.4}$$

Όπως προαναφέρθηκε, μπορούμε να προσομοιώσουμε πεπερασμένο αριθμό ψηφίων, άρα η ζητούμενη πιθανότητα μπορεί μόνο να προσεγγιστεί από την προσομοίωση. Η επιτυχία λοιπόν της προσομοίωσης έγκειται στο κατά πόσο ακριβής είναι η προσέγγιση-εκτίμηση της.

Αφού κάθε προσομοιωτικό αποτέλεσμα ξεχωριστά, σε μια τέτοια στοχαστική προσομοίωση αντιμετωπίζεται ως τυχαία μεταβλητή, για να κρίνουμε την επιτυχία ολόκληρης της προσομοίωσης πρέπει να μελετήσουμε τα στατιστικά χαρακτηριστικά της. Κυρίως ενδιαφερόμαστε για δύο στατιστικά χαρακτηριστικά της εκτίμησης μιας προσομοίωσης. Πρώτον, επιθυμούμε η τυπική απόκλιση της εκτίμησης να τείνει στο μηδέν όταν το N τείνει στο άπειρο κάτι που καθιστά την εκτίμηση συνεπή (consistent). Δεύτερον, επιθυμούμε η μέση τιμή της εκτίμησης όταν το N τείνει στο άπειρο να ισούται με το πραγματικό μέγεθος, εν προκειμένω $E\left\{\hat{P}_E\right\} = P_E$. Αυτή η ιδιότητα καθιστά την εκτίμηση αμερόληπτη (unbiased).

Σε αυτό το σημείο, να σημειώσουμε ότι οι στοχαστικές προσομοιώσεις που κάνουν εκτίμηση μιας παραμέτρου μέσω διαδοχικών εκτιμήσεων που κάθε μία εμπεριέχει έναν παράγοντα τυχαιότητας, όπως συμβαίνει και στο προηγούμενο παράδειγμα, είναι ευρέως γνωστές ως ‘προσομοιώσεις Monte Carlo’. Η ονομασία αυτή παραπέμπει στα τυχερά παιχνίδια για τα οποία φημίζεται το Monte Carlo.

3.4 Επιλογή της Πλατφόρμας για την Προσομοίωση μας

Ένας από τους βασικούς στόχους αυτής της εργασίας είναι η ανάπτυξη προσομοιωτή για την εκτίμηση της επίδοσης LDPC κωδίκων, που θα παρουσιαστούν

στο κεφάλαιο 4, υπό διάφορα σχήματα διαμόρφωσης (BPSK, QAM, κλπ) και υπό διάφορους τύπους διαύλου (AWGN, Rayleigh, Rice). Ένα από τα πρώτα ζητήματα που έπρεπε να απαντηθούν ήταν η επιλογή της πλατφόρμας προσομοίωσης, του προγραμματιστικού περιβάλλοντος δηλαδή πάνω στο οποίο θα γίνει η ανάπτυξη της προσομοίωσης.

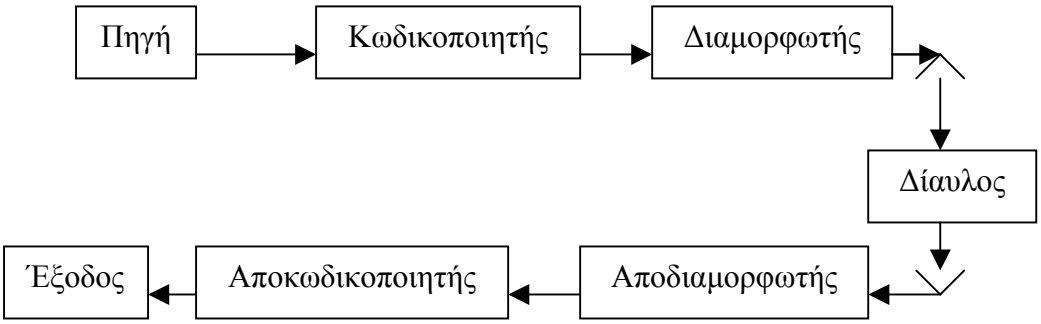
Βασικοί υποψήφιοι ήταν η γλώσσα προγραμματισμού C και το περιβάλλον Matlab. Ο αρχικός λόγος είναι οι καλές επιδόσεις τους σε απαιτητικές υπολογιστικές διεργασίες, όπως η προσομοίωση μας. Όπως έχει ήδη γίνει αντιληπτό η εκτίμηση της επίδοσης ενός κώδικα αποτελεί στοχαστική διαδικασία αφού σε κάθε βήμα της προσομοίωσης η είσοδος και στον κωδικοποιητή και στον αποκωδικοποιητή είναι τυχαία μεταβλητή. Στον κωδικοποιητή καταφθάνουν κατά τυχαίο τρόπο ψηφία από τη ψηφιακή πηγή πληροφορίας και στην είσοδο του αποκωδικοποιητή εμπεριέχεται και η τυχαιότητα που έχει εισαχθεί από τη στιγμιαία τιμή των χαρακτηριστικών του διαύλου. Ως στοχαστική προσομοίωση λοιπόν απαιτεί μεγάλο αριθμό επαναλήψεων άρα και πολλούς υπολογισμούς.

Σε πρώτη φάση, η γλώσσα C είναι αποδοτικότερη όσον αφορά το χρόνο εκτέλεσης από το Matlab, ως γλώσσα χαμηλότερου επιπέδου. Όμως τα πλεονεκτήματα της χρήσης του Matlab κρίθηκαν σημαντικότερα. Πρώτον, είναι ευρύτατα αποδεκτό από την κοινότητα των μηχανικών. Επίσης, από τα εγγενή χαρακτηριστικά του Matlab (MATrix LABoratory) υπάρχει μεγάλη ευχέρεια και αποτελεσματικότητα στη χρήση πινάκων κάτι που θα μας φανεί πολύ χρήσιμο (γεννήτορας πίνακας, πίνακας ελέγχου ισοτιμίας, εύρεση αντιστρόφου κ.ά.). Γενικότερα από τη χρήση πινάκων, σαφώς και έχει πιο εμπλουτισμένη βιβλιοθήκη με έτοιμες ρουτίνες από τη γλώσσα C παράλληλα με πολύ βοηθητικά πρόσθετα όπως το Simulink. Μάλιστα, το Matlab είναι από τα πλέον εξελισσόμενα περιβάλλοντα εμπλουτιζόμενο συνεχώς κάτι που διαφαίνεται και στη συνεχή ανανέωση του μέσω νέων εκδόσεων. Χαρακτηριστικό παράδειγμα είναι η τελευταία έκδοση μέχρι ώρας του Matlab (R2007) που περιλαμβάνει πλήρες προσομοιωτικό μοντέλο για το νέο πρότυπο DVB-S2 που χρησιμοποιεί LDPC κωδικοποίηση. Σαφώς και τέτοιες δυνατότητες δεν μπορούν να προσφερθούν από μια γλώσσα προγραμματισμού χαμηλού επιπέδου όπως η C. Αξίζει να σημειώσουμε ότι και αυτό που προσφέρει καλύτερα η C, δηλαδή ταχύτητα εκτέλεσης, μπορεί εν μέρει να χρησιμοποιηθεί καθώς υπάρχει η δυνατότητα ενσωμάτωσης στο περιβάλλον Matlab αρχείων γραμμένα σε C. Τέλος, σημαντικό ρόλο στην προσομοίωση μας, όπως και γενικότερα, παίζει η οπτική παρουσίαση των αποτελεσμάτων. Είναι αδιαμφισβήτητο ότι το γραφικό περιβάλλον του Matlab είναι από τα πλέον εξελιγμένα.

3.5 Μοντελοποίηση ενός απλού πομποδέκτη για την εκτίμηση της επίδοσης του

Σε αυτή την ενότητα θα κάνουμε τη γενική περιγραφή του τηλεπικοινωνιακού συστήματος που θα προσομοιώσουμε με στόχο να μελετήσουμε την επίδοση του σε όρους ρυθμού λαθών ανά ψηφίο (BER). Αυτή η μελέτη θα γίνει κάτω από διάφορα σχήματα διαμόρφωσης, σε διάφορους τύπους διαύλου και σαν συνάρτηση του

σηματοθορυβικού λόγου (SNR ή E_b/ N_0). Το σταθερό στοιχείο είναι η LDPC κωδικοποίηση αφού κάτι τέτοιο αποτελεί και τον κύριο στόχο της εργασίας. Βέβαια, και στην LDPC κωδικοποίηση θα μελετηθούν διάφοροι παράγοντες όπως λογαριθμικός ή μη αποκωδικοποιητής και διάφορες τιμές για τον μέγιστο αριθμό επαναλήψεων του επαναληπτικού αλγορίθμου αποκωδικοποίησης όπως αυτός θα παρουσιαστεί στο επόμενο κεφάλαιο. Ο κώδικας της προσομοίωσης είναι γραμμένος στο περιβάλλον Matlab και τα αποτελέσματα θα παρουσιαστούν και θα σχολιαστούν στο κεφάλαιο 5. Στη συνέχεια θα αναλύσουμε τη μοντελοποίηση καθενός υποσυστήματος που εμπλέκεται στην προσομοίωση μας. Η ανάλυση θα γίνει με βάση το μπλοκ διάγραμμα του τηλεπικοινωνιακού συστήματος που θα προσομοιώσουμε και το οποίο φαίνεται στο σχήμα 3.4.



Σχήμα 3.4 Μπλοκ διάγραμμα προσομοιούμενου συστήματος

3.5.1 Πηγή Ψηφιακής Πληροφορίας

Ως πηγή του συστήματος θεωρούμε την ακολουθία δυαδικών ψηφίων που καταφθάνει στην είσοδο του κωδικοποιητή κατά τυχαίο τρόπο. Δεν μας ενδιαφέρει τι είδους είναι η πρωτογενής πηγή πληροφορίας (ήχος, εικόνα κλπ) ούτε και τα ενδιάμεσα στάδια πριν τον κωδικοποιητή (βλ. σχήμα 1.2) καθώς δεν επηρρεάζουν το μέγεθος που μας ενδιαφέρει: το ρυθμό λαθών ανά ψηφίο (BER) στην έξοδο του αποκωδικοποιητή στο δέκτη.

Η ακολουθία που περιγράφηκε μπορεί να υλοποιηθεί στο Matlab μέσω της συνάρτησης randint που επιστρέφει 0 ή 1 με ίση πιθανότητα.

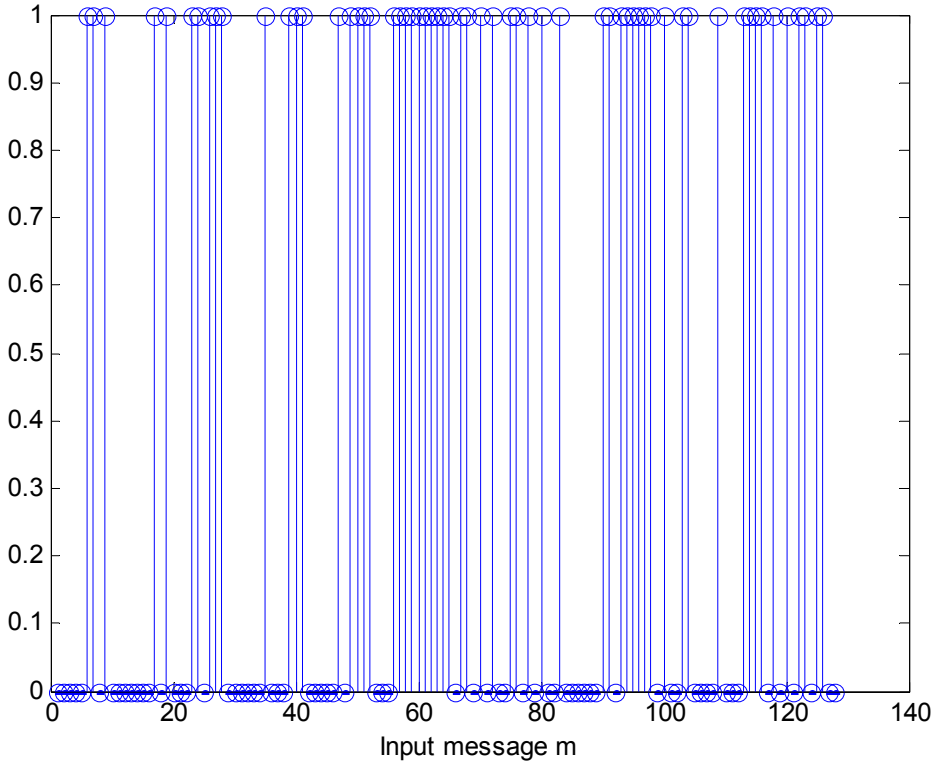
3.5.2 Κωδικοποιητής

Ο κώδικας που κατασκευάσαμε είναι LDPC (256, 128). Αυτό σημαίνει ότι ο κωδικοποιητής λαμβάνει την είσοδο από την πηγή σε μορφή πακέτων των 128 bits και στην έξοδο του δίνει πακέτα των 256 bits. Αν η ακολουθία δυαδικών ψηφίων μήκους 128 bits συμβολιστεί με **m,** τότε στην έξοδο του αποκωδικοποιητή θα εμφανιστεί το κωδικοποιημένο διάνυσμα **c** μήκους 256 bits που προκύπτει μέσω της γνωστής σχέσης

$$c = m \circ G$$

όπου **G** ο γεννήτορας πίνακας του κώδικα.

Μέσω του Matlab (συνάρτηση stem) δίνουμε στο σχήμα 3.5 μια πιθανή εκδοχή του **m**

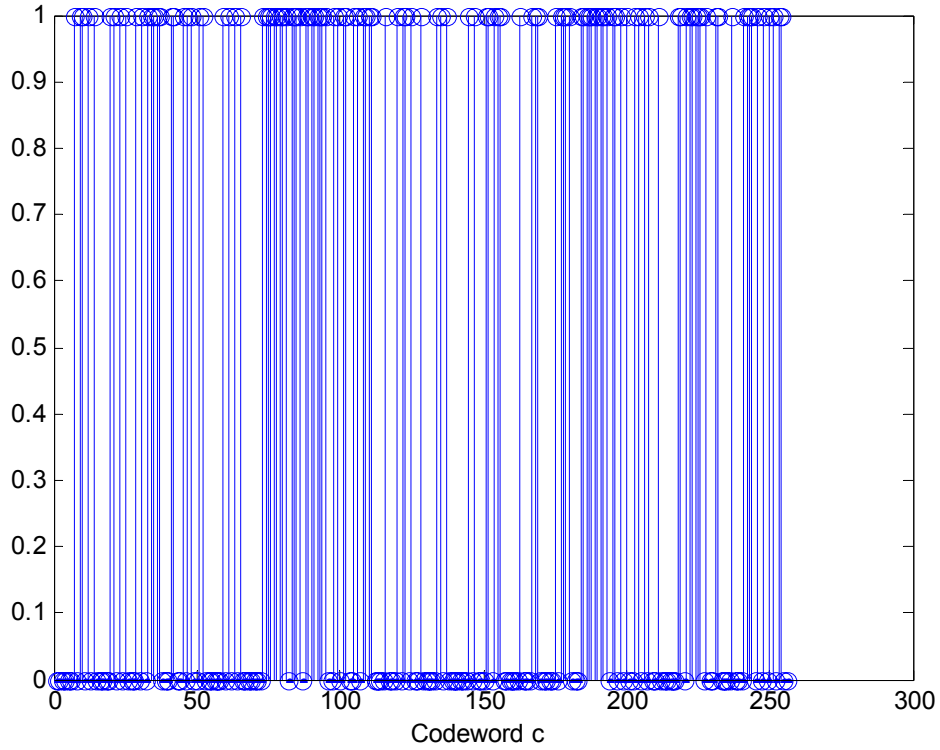


Σχήμα 3.5 Μία τυχαία είσοδος στον κωδικοποιητή

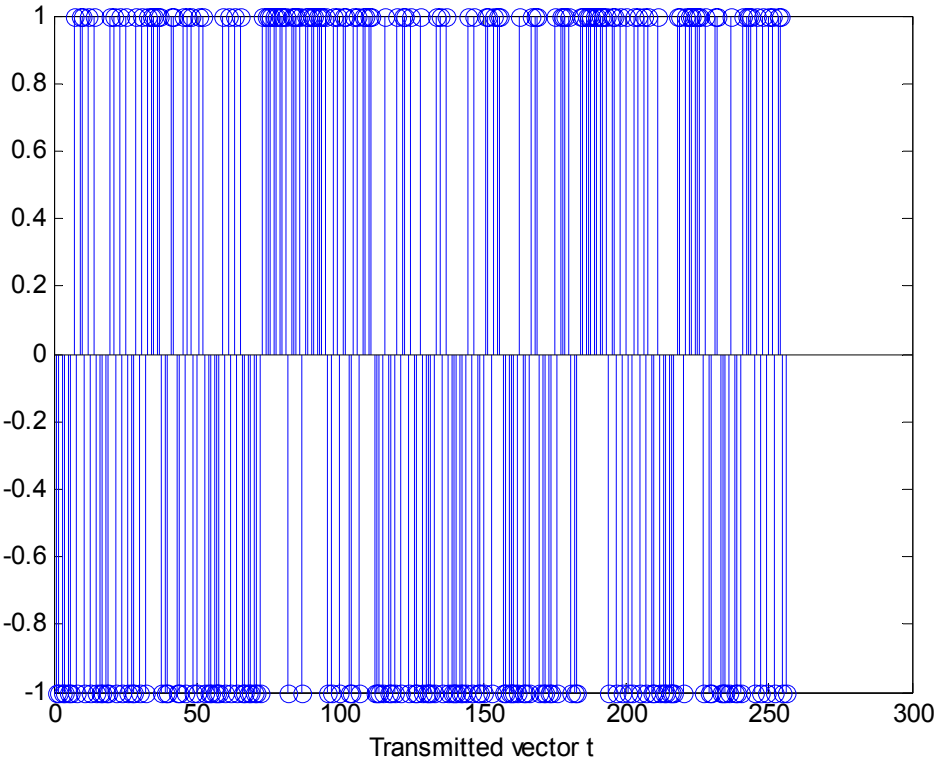
Το κωδικοποιημένο μήνυμα που αντιστοιχεί στο παραπάνω διάνυσμα **m** φαίνεται στο σχήμα 3.6.

3.5.3 Διαμορφωτής

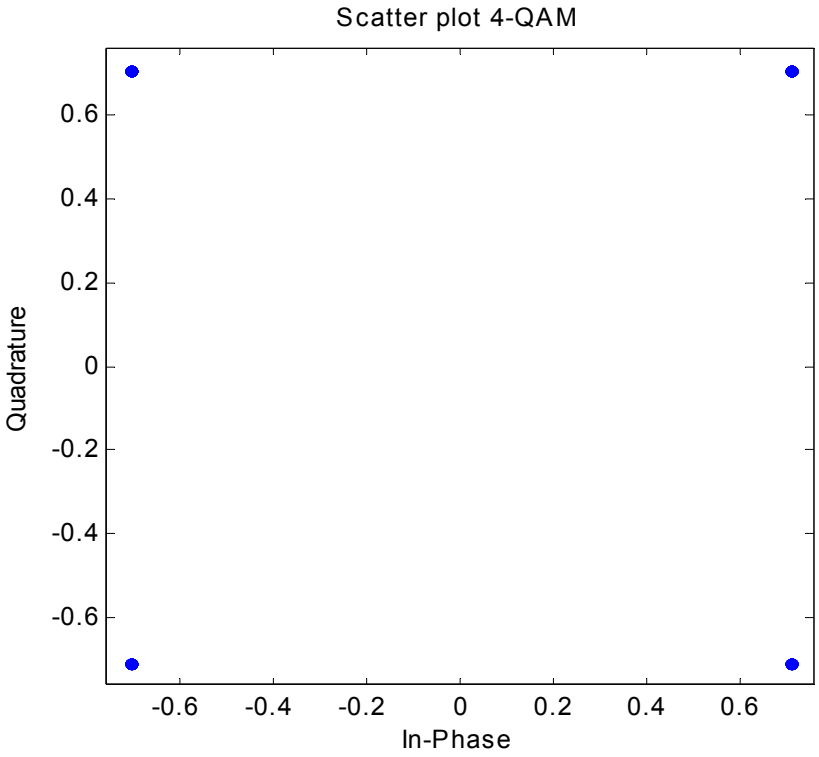
Στην είσοδο του διαμορφωτή καταφθάνουν κατά σειρά οι κωδικές λέξεις από την έξοδο του κωδικοποιητή. Κάθε σχήμα διαμόρφωσης αντιστοιχεί τα εισερχόμενα δυαδικά ψηφία σε σύμβολα – παλμούς προς μετάδοση. Το απλούστερο σχήμα διαμόρφωσης που θα χρησιμοποιήσουμε είναι η διαμόρφωση BPSK, όπου το ψηφίο 1 αντιστοιχίζεται σε παλμό πλάτους +A και το ψηφίο 0 αντιστοιχίζεται σε παλμό πλάτους –A. Εμείς θα χρησιμοποιήσουμε την κανονικοποιημένη τιμή +1 και –1 (βλ. σχήμα 3.7). Άλλο σχήμα διαμόρφωσης είναι η διαμόρφωση M-QAM όπου αντιστοιχίζεται ένα σύμβολο ανά $\log_2 M$ δυαδικά ψηφία. Για παράδειγμα, στη διαμόρφωση 4-QAM ο διαμορφωτής αντιστοιχεί ανά δύο ψηφία (00, 01, 10, ή 11) και ένα σύμβολο οπότε προκύπτουν συνολικά 4 διαφορετικά σύμβολα. Ομοίως και για QAM διαμορφώσεις ανώτερης τάξης. Τα προς μετάδοση σύμβολα ανήκουν στο μιγαδικό επίπεδο και αποδίδονται σχηματικά στη συνέχεια (Σχ. 3.8, 3.9).



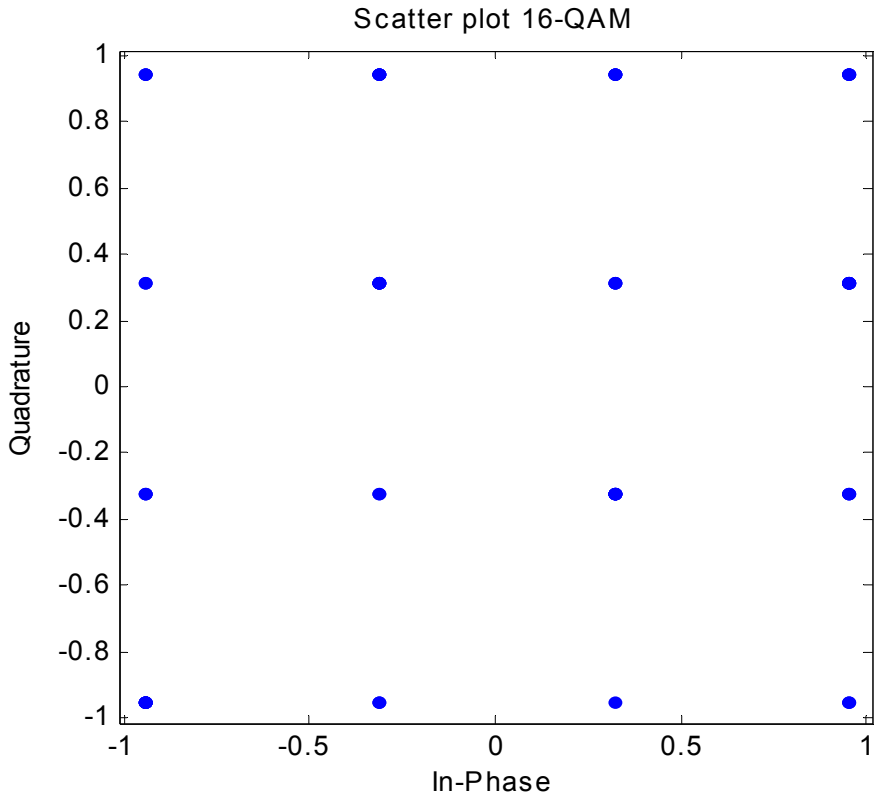
Σχήμα 3.6 Κωδική λέξη που αντιστοιχεί στο μήνυμα πληροφορίας του σχήματος 3.5



Σχήμα 3.7 BPSK διαμόρφωση στην κωδική λέξη του σχήματος 3.6



Σχήμα 3.8 Αστερισμός 4-QAM όπου φαίνονται τα 4 διαφορετικά μεταδιδόμενα σύμβολα



Σχήμα 3.9 Αστερισμός 16-QAM όπου φαίνονται τα 16 διαφορετικά μεταδιδόμενα σύμβολα

3.5.4 Δίαυλος

Στην περιγραφή των διαφορετικών τύπων διαύλων θα εστιάσουμε την προσοχή μας σε διαύλους που απαντώνται κυρίως στις ασύρματες επικοινωνίες. Ως πρώτη κατηγοριοποίηση κατανέμουμε τους διαύλους σε τύπου προσθετικού λευκού θορύβου Gauss (AWGN) και σε διαύλους που προκαλούν διαλείψεις, δηλαδή απότομες μεταβολές στην ένταση του λαμβανόμενου σήματος. Προχωρώντας ένα βήμα στη δεύτερη κατηγορία θα αναλύσουμε ξεχωριστά διαύλους τύπου Rayleigh και τύπου Rice ενώ θα μελετηθεί και η χρονική μεταβλητότητα του διαύλου λόγω κίνησης.

3.5.4.1 Προσθετικός Λευκός Θόρυβος Gauss (AWGN)

Το μοντέλο AWGN είναι ένα θεωρητικό κατασκεύασμα για την περιγραφή του θορύβου που εισάγει ο δίαυλος, που βρίσκει σημαντική εφαρμογή στην ανάλυση τηλεπικοινωνιακών συστημάτων.

Πρώτον, το ότι ο θόρυβος χαρακτηρίζεται Gauss σημαίνει ότι ακολουθεί την κατανομή Gauss. Γενικά, λέμε ότι η τυχαία μεταβλητή Y έχει κατανομή Gauss αν η συνάρτηση πυκνότητας πιθανότητας αυτής έχει τη μορφή:

$$p_Y(y) = \frac{1}{\sqrt{2\pi\sigma_Y}} \exp\left[-\frac{(y - m_Y)^2}{2\sigma_Y^2} \right] \tag{3.5}$$

όπου m_Y είναι η μέση τιμή και σ_Y^2 η μεταβλητότητα της τυχαίας μεταβλητής Y. Το γιατί τώρα βρίσκει πρακτική εφαρμογή η μοντελοποίηση του θορύβου με βάση την γκαουσιανή κατανομή έχει τις ρίζες του στο θεώρημα του κεντρικού ορίου (central limit theorem). Το εν λόγω θεώρημα λοιπόν, δηλώνει ότι αν μια τυχαία μεταβλητή, εν προκειμένω ο θόρυβος, προκύπτει ως το άθροισμα μεγάλου αριθμού τυχαίων μεταβλητών ανεξάρτητων μεταξύ τους, τότε η κατανομή πιθανότητας της τυχαίας μεταβλητής προσεγγίζεται από τη γκαουσιανή. Για παράδειγμα, έστω ο θερμικός θόρυβος, ο ηλεκτρικός θόρυβος δηλαδή που οφείλεται στην τυχαία κίνηση των ηλεκτρονίων σε έναν αγωγό. Επειδή ο αριθμός των ηλεκτρονίων σε μία αντίσταση είναι πολύ μεγάλος και οι τυχαίες κινήσεις τους είναι στατιστικά ανεξάρτητες, το θεώρημα κεντρικού ορίου υπαγορεύει ότι ο θερμικός θόρυβος ακολουθεί κατανομή Gauss με μηδενική μάλιστα μέση τιμή.

Έπειτα, πρέπει να σχολιάσουμε το χαρακτηρισμό του θορύβου ως λευκός. Ο λευκός θόρυβος λοιπόν αποτελεί μία ιδανική μορφή θορύβου, η πυκνότητα φάσματος ισχύος του οποίου είναι ανεξάρτητη της συχνότητας λειτουργίας. Ο όρος λευκός αποδόθηκε με την έννοια ότι το λευκό φως περιέχει ίσες ποσότητες ακτινοβολίας από όλες τις ορατές συχνότητες. Από τον ορισμό του λευκού θορύβου προκύπτει ύστερα από ανάλυση ότι οποιαδήποτε δύο διαφορετικά δείγματα λευκού θορύβου, ανεξάρτητα πό το πόσο κοντά στο χρόνο λαμβάνονται, είναι ασυσχέτιστα. Δηλαδή, σε περίπτωση λευκού θορύβου Gauss τα δύο δείγματα είναι ασυσχέτιστα και ανεξάρτητα. Για αυτό το λόγο έχει γραφτεί χαρακηριστικά ότι ο λευκός θόρυβος Gauss αποτελεί το ζενίθ της "τυχαιότητας".

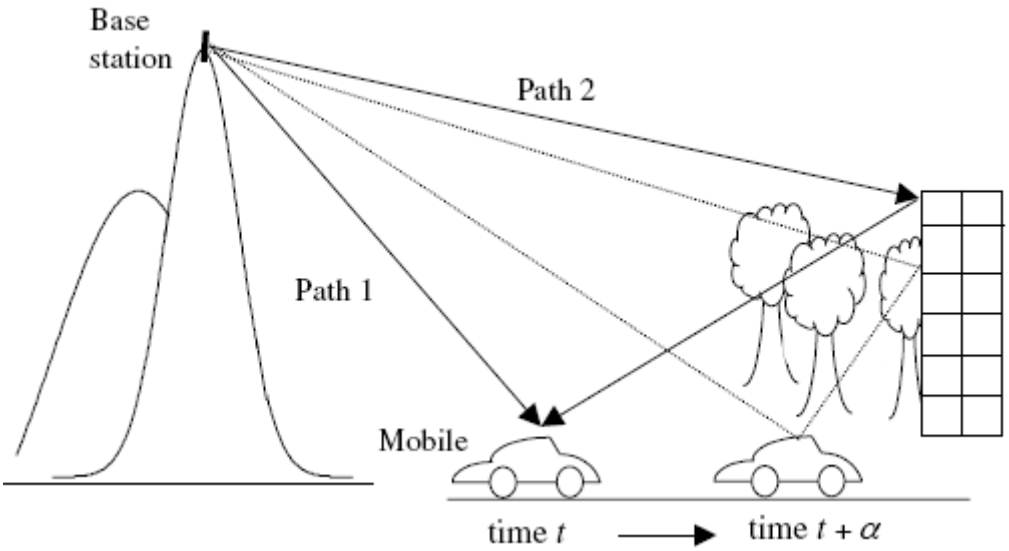
Τέλος, με τον όρο προσθετικός εννοούμε ότι το λαμβανόμενο σήμα y(t) προκύπτει ως το άθροισμα του μεταδιδόμενου σήματος x(t) με το θόρυβο n(t), για κάθε χρονική στιγμή t.

$$y(t) = x(t) + n(t) \tag{3.6}$$

Να σημειώσουμε ότι το Matlab παρέχει τη συνάρτηση awgn, η οποία προσθέτει λευκό θόρυβο Gauss στο σήμα μας.

3.5.4.2 Δίαυλος τύπου Rayleigh

Στη συνέχεια θα επικεντρώσουμε σε διαύλους που προκαλούν διαλείψεις στο λαμβανόμενο σήμα. Τέτοια φαινόμενα παρατηρούνται συχνά στις κινητές τηλεπικοινωνίες και οφείλονται κατά βάση στην πολυδιαδρομική διάδοση του σήματος και στην κίνηση του δέκτη (σχήμα 3.10). Επίσης, παρατηρούνται και στις δορυφορικές επικοινωνίες όπου εδώ οι διαλείψεις οφείλονται κυρίως στις ιδιαιτερότητες που εμφανίζει η διάδοση ηλεκτρομαγνητικών κυμάτων στην ατμόσφαιρα (π.χ. ατμοσφαιρικές κατακρημνίσεις).



Σχήμα 3.10 Παράδειγμα πολυδιαδρομικής διάδοσης και κινητικότητας δέκτη που οδηγούν σε διαλείψεις

Έχει αποδειχθεί ότι η κατανομή της περιβάλλουσας των σημάτων που υποφέρουν από διαλείψεις λόγω πολυδιαδρομικής διάδοσης όταν δεν υπάρχει οπτική επαφή πομπού-δέκτη (NLOS) περιγράφεται από την κατανομή Rayleigh. Η πυκνότητα πιθανότητας της κατανομής Rayleigh δίνεται από τη σχέση

$$\begin{aligned} p_X(x) &= \frac{2x}{b}\exp\left(-\frac{x^2}{b}\right),\ x>0 \\ p_X(x) &= 0,\ x \le 0 \end{aligned} \tag{3.7}$$

3.5.4.3 Δίαυλος τύπου Rice

Όταν το σήμα μας υποφέρει από διαλείψεις και υπάρχει οπτική επαφή πομπού-δέκτη (LOS), τότε η περιβάλλουσα του σήματος ακολουθεί την κατανομή Rice. Η πυκνότητα πιθανότητας της κατανομής Rice δίνεται από τη σχέση (a, b>0)

$$p_X(x) = \frac{x}{b}\exp\left[\left(-x^2 + a^2\right)/2b\right] I_0\left(\frac{ax}{b}\right),\ x>0$$
$$p_X(x) = 0,\ x \le 0 \tag{3.8}$$

όπου

- **x**, η λαμβανόμενη τιμή της περιβάλλουσας του σήματος (σε Volts)
- **a**, το πλάτος του κυρίαρχου σήματος (LOS)
- $I_0(x)$ η τροποποιημένη συνάρτηση Bessel πρώτου είδους.

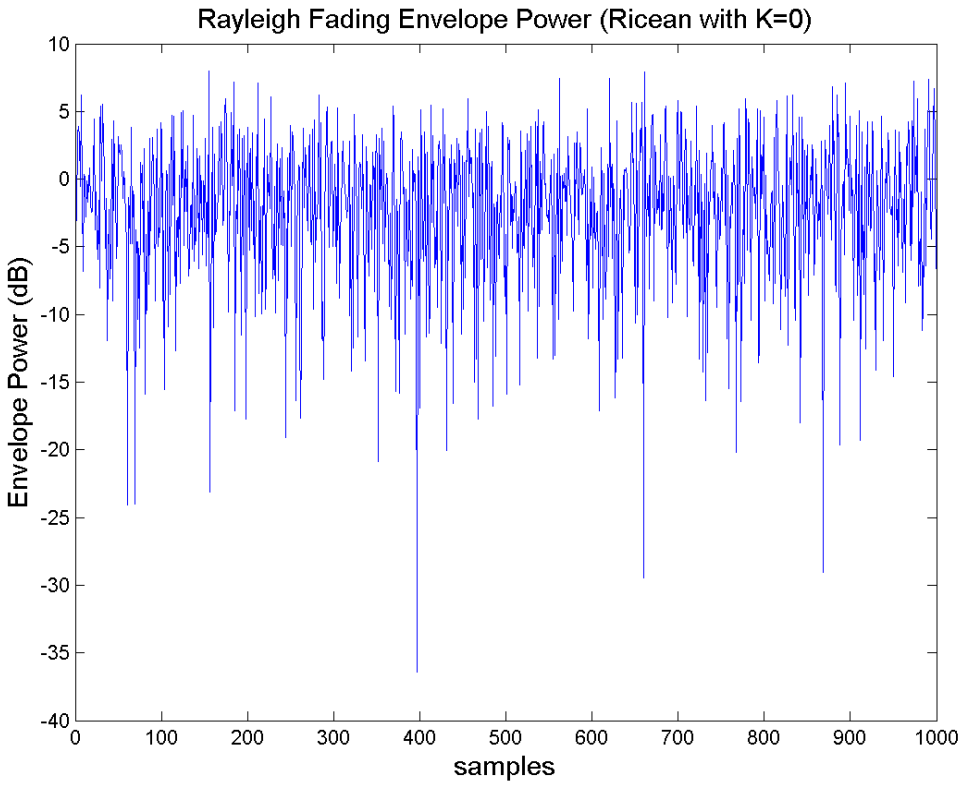
Θεωρητικά, αναμένουμε μικρότερες επιβαρύνσεις σε διαύλους τύπου Rice σε σχέση με διαύλους Rayleigh ακριβώς λόγω της οπτικής επαφής πομπού – δέκτη. Για τον ακριβέστερο χαρακτηρισμό ενός διαύλου Rice χρησιμοποιούμε τον παράγοντα K, όπου

$$K = \frac{\text{ΙΣΧΥΣ ΑΠΕΥΘΕΙΑΣ ΚΥΜΑΤΟΣ}}{\text{ΙΣΧΥΣ ΣΚΕΔΑΣΜΕΝΩΝ ΚΥΜΑΤΩΝ}}$$

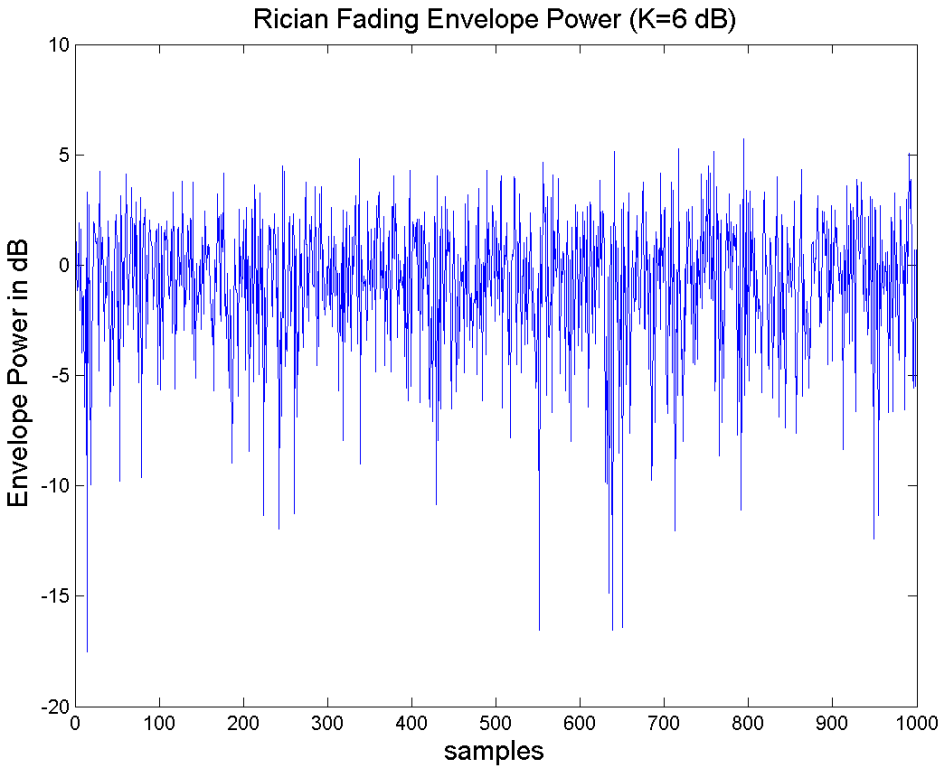
ή

$$K = \frac{a^2}{2b}$$

Από τα παραπάνω, προκύπτει ότι αν K=0 τότε ο δίαυλος Rice ισοδυναμεί με δίαυλο Rayleigh. Στα επόμενα σχήματα (**Σχ. 3.11**, **Σχ. 3.12**), δίνουμε τα αποτελέσματα της προσομοίωσης στο Matlab για διαύλους Rayleigh, Rice αντίστοιχα όπου οι τιμές ισχύος της περιβάλλουσας του σήματος (Envelope Power) έχουν παραχθεί βάσει των αντίστοιχων πυκνοτήτων πιθανότητας όπως αυτές δίνονται από τις εξισώσεις **3.7, 3.8**.



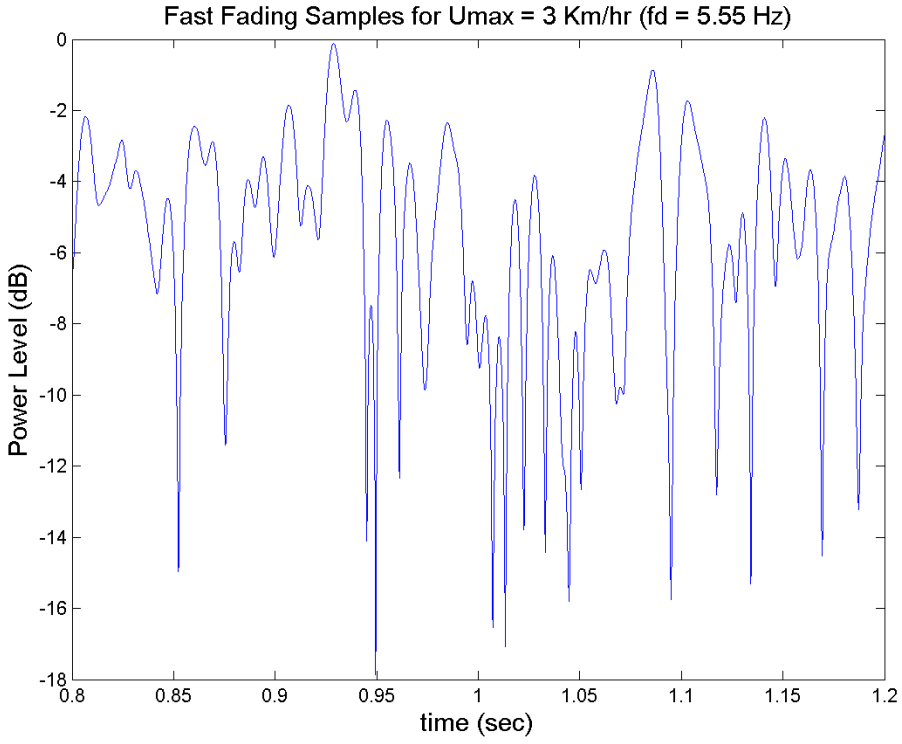
Σχήμα 3.11 Περιβάλλουσα Rayleigh με διαλείψεις



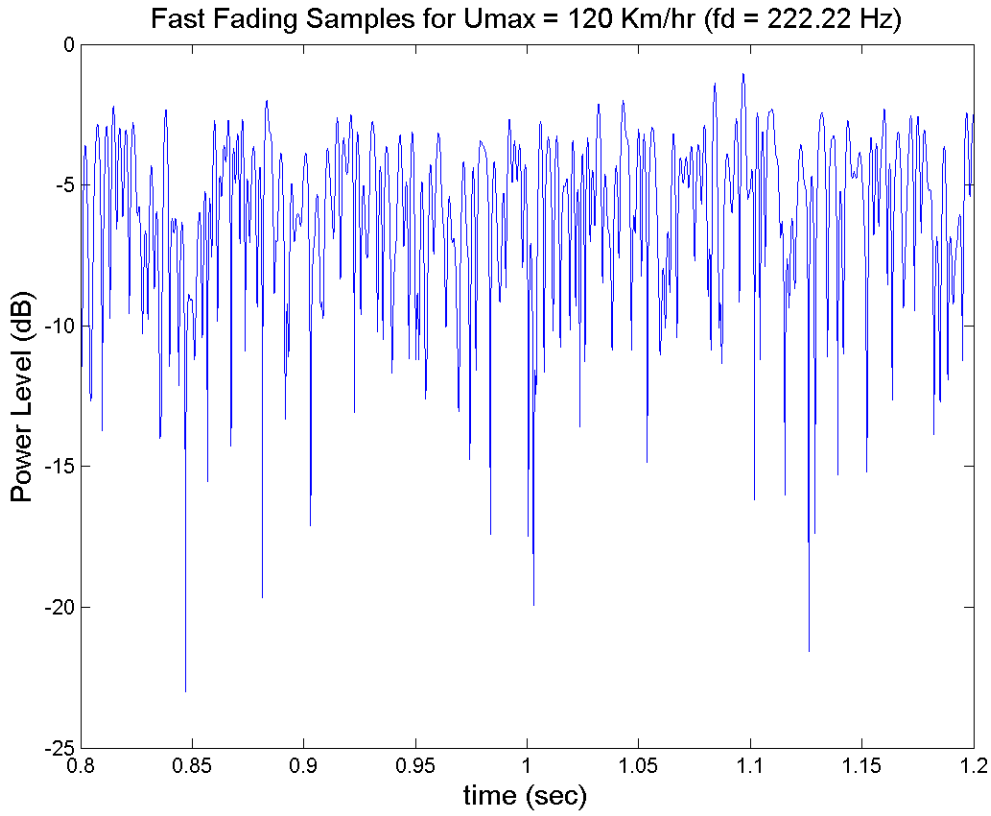
Σχήμα 3.12 Περιβάλλουσα Rice (K=6 **dB**) με διαλείψεις

3.5.4.4 Διαλείψεις λόγω κίνησης του δέκτη

Μέχρι τώρα, περιγράψαμε τις διαλείψεις που οφείλονται στην πολυδιαδρομική διάδοση του σήματος και περιγράψαμε τη βασική κατηγοριοποίηση σε δίαυλο Rayleigh και Rice. Όπως προαναφέρθηκε όμως, στις κινητές επικοινωνίες διαλείψεις προκαλούνται και λόγω της κίνησης του δέκτη. Η συνεχής μεταβολή της θέσης του δέκτη συνεπάγεται και συνεχή μεταβολή στα μονοπάτια διάδοσης, φαινόμενο που οδηγεί σε διαλείψεις. Για παράδειγμα, στο σχήμα 3.10 παρατηρούμε ότι άλλα είναι τα μονοπάτια διάδοσης τη χρονική στιγμή t και άλλα τη χρονική στιγμή t + a, ακριβώς λόγω της κίνησης του δέκτη. Αυτή η χρονική μεταβλητότητα περιγράφεται μέσω της φασματικής πυκνότητας ισχύος Doppler και υπάρχουν διάφορα μοντέλα για την περιγραφή της. Εμείς, ακολουθώντας την αναφορά [4] υιοθετούμε στην προσομοίωση μας το μοντέλο Jakes. Προφανώς, βασική παράμετρος για την εξέταση διαλείψεων λόγω κίνησης είναι η ταχύτητα του κινητού δέκτη. Στη συνέχεια, παραθέτουμε τα αποτελέσματα της προσομοίωσης για μέγιστη ταχύτητα δέκτη V_{max}= 3 km/hr και για V_{max}= 120 km/hr, για δίαυλο Rayleigh και τις δύο φορές. **Συμπεραίνουμε ότι όσο αυξάνεται η μέγιστη ταχύτητα δέκτη τόσο αυξάνεται ο ρυθμός διαλείψεων λόγω κίνησης.** Να σημειώσουμε ότι στην προσομοίωση για τη μελέτη επίδοσης LDPC κωδίκων υπό περιβάλλον διαλείψεων, όπως αυτή θα παρουσιαστεί στο κεφάλαιο 5, είτε αναλύουμε δίαυλο Rayleigh είτε Rice η μέγιστη ταχύτητα δέκτη είναι σταθερή παράμετρος ορισμένη ως $\mathbf{V_{max}}$**= 10 km/hr**.



Σχήμα 3.13 Διαλείψεις για V_{max}= 3 km/hr



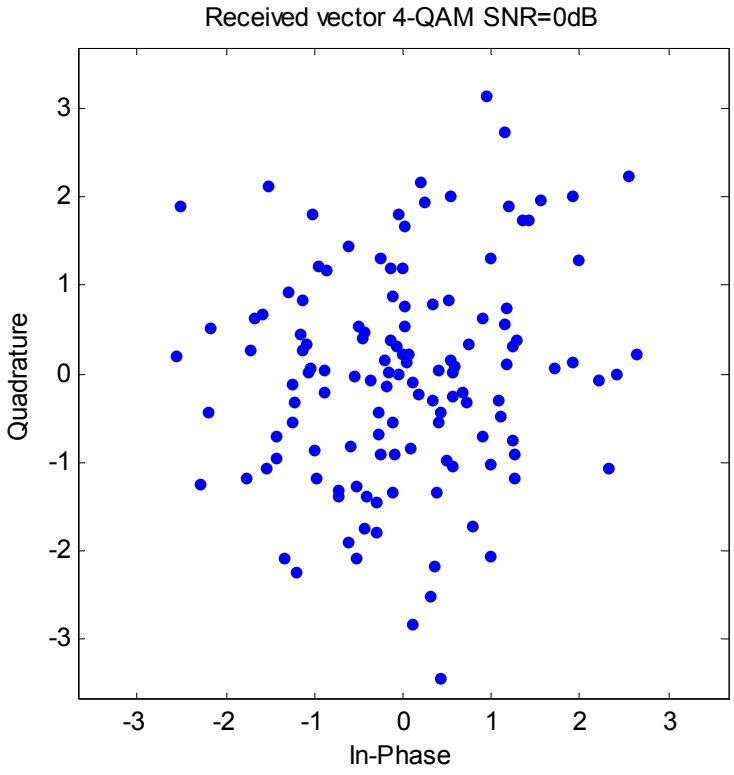
Σχήμα 3.14 Διαλείψεις για V_{max}= 120 km/hr

3.5.5 Αποδιαμορφωτής

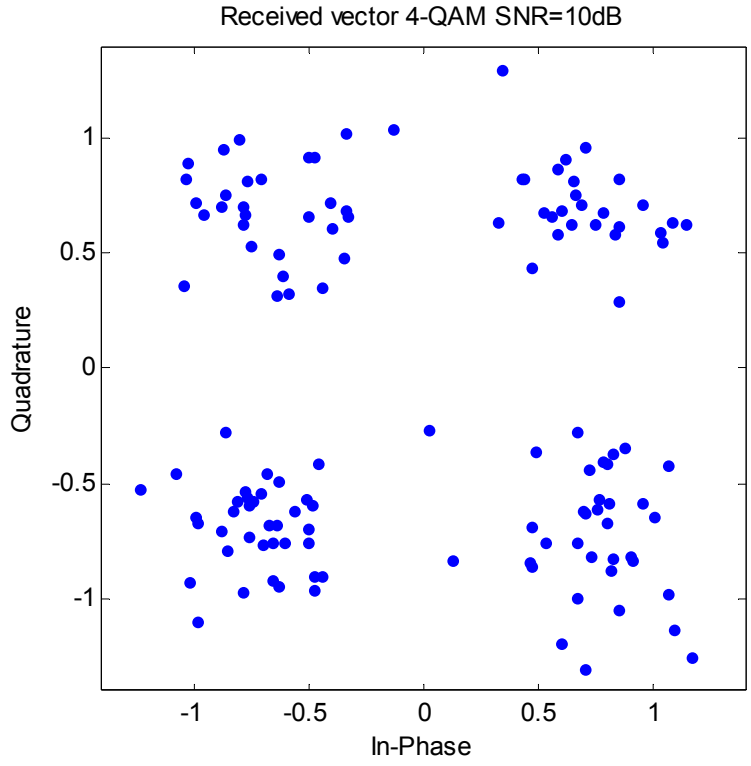
Ο αποδιαμορφωτής κάνει την αντίστροφη διαδικασία από το διαμορφωτή. Σαν είσοδο δηλαδή παίρνει σύμβολα και σαν έξοδο παράγει την ανάλογη ακολουθία bits. Προφανώς, πρέπει να υπάρχει αντιστοιχία με το σχήμα διαμόρφωσης, δηλαδή διαμόρφωση PSK συνεπάγεται και αποδιαμόρφωση PSK. Όπως είδαμε στην ενότητα 3.5.3 εκπέμπονται σύμβολα, ύστερα από τη διαδικασία της διαμόρφωσης, που το καθένα αντιστοιχεί σε μια από τις επιτρεπόμενες τιμές κάθε σχήματος διαμόρφωσης, π.χ. για 4-QAM είδαμε ότι υπάρχουν δυνητικά 4 διαφορετικά σύμβολα. Μετά από τη μετάδοση όμως και λόγω των επιβαρύνσεων που έχει εισάγει ο δίαυλος τα σύμβολα καταφθάνουν αλλοιωμένα. Δουλειά του αποδιαμορφωτή είναι για κάθε ληφθέν σύμβολο να βρει ποιο από τα αρχικά σύμβολα απέχει λιγότερο και να κάνει έτσι την αντιστοίχιση. Η έννοια της απόστασης εδώ ορίζεται ως η ευκλίδεια απόσταση στο μιγαδικό επίπεδο των συμβόλων. Από τα παραπάνω προκύπτει ότι αν η αλλοίωση είναι μεγάλη ενδέχεται ο αποδιαμορφωτής να κάνει λανθασμένη εκτίμηση για το σύμβολο που εκπέμφθηκε. Ύστερα τώρα από την εκτίμηση για το σύμβολο παράγεται η ακολουθία ψηφίων που αντιστοιχεί στο εν λόγω σύμβολο και που οδηγείται στη συνέχεια στον αποκωδικοποιητή.

Το όλο ζήτημα περί της εκτίμησης συμβόλων ενδείκνυται να παρουσιαστεί γραφικά. Για αυτό το σκοπό θα χρησιμοποιήσουμε σχήματα που προέκυψαν από την προσομοίωση μας. Στα σχήματα 3.15, 3.16 δίνουμε ληφθέντα σύμβολα που αντιστοιχούν σε διαμόρφωση 4-QAM και σηματορυβικό λόγο 0 και 10 dB αντίστοιχα. Χωρίς να μας ενδιαφέρει ιδιαίτερα σε αυτό το σημείο σημειώνουμε ότι το σύνολο των ληφθέντων συμβόλων αντιστοιχεί σε μια κωδική λέξη (256 bits) του κώδικα μας, οπότε το πλήθος τους είναι (256 / $\log_2 4$)=128. Παρατηρούμε τώρα, και σε αντιπαραβολή με το σχήμα 3.8, ότι για SNR 0 dB (πολύ χαμηλή τιμή) έχουμε σημαντική αλλοίωση στις θέσεις των συμβόλων στο μιγαδικό επίπεδο κάτι που αναπόφευκτα θα οδηγήσει τον αποδιαμορφωτή σε σωρεία σφαλμάτων δυσχεραίνοντας έτσι και τη μετέπειτα εργασία του αποκωδικοποιητή. Αντίθετα, με την αύξηση του SNR σε 10 dB (αρκετά καλή τιμή) τα σύμβολα βρίσκονται σε περιοχές σαφώς πιο εντοπισμένες και γύρω από τα αρχικά σύμβολα.

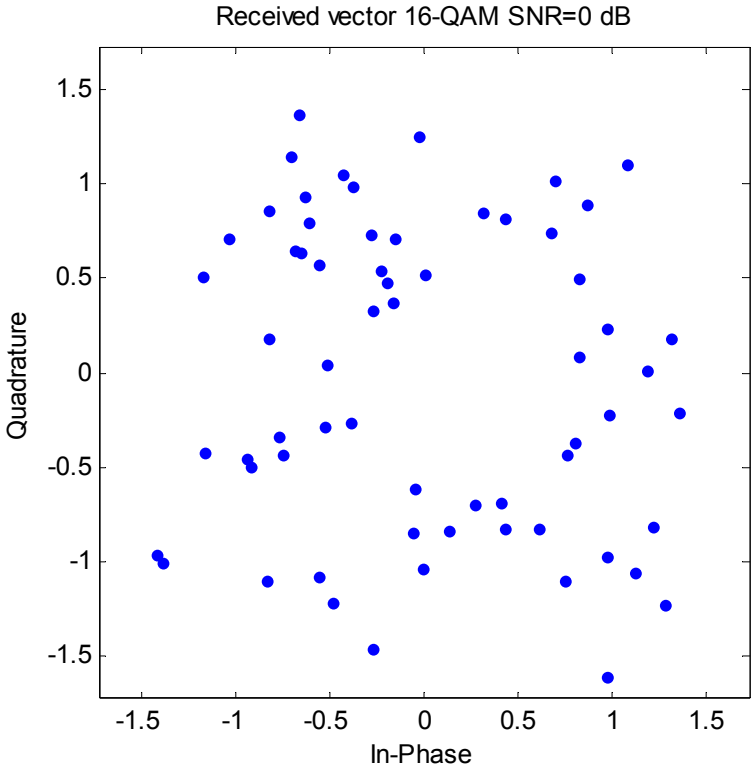
Έτσι, σε αυτή την περίπτωση ακόμα και αν υπάρχει κάποια αλλοίωση του συμβόλου ύστερα από τη μετάδοση του, ο αποδιαμορφωτής θα κάνει τις περισσότερες φορές σωστή εκτίμηση διότι το ληφθέν σύμβολο βρίσκεται πιο κοντά στο αρχικό σύμβολο παρά στα υπόλοιπα.



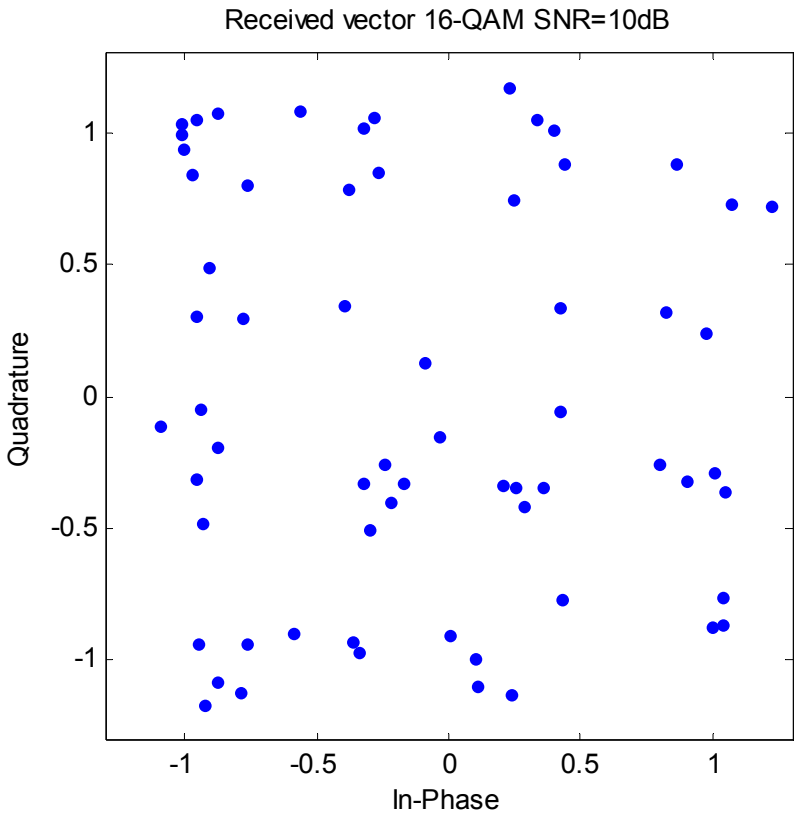
Σχήμα 3.15 Ληφθέντα σύμβολα για 4-QAM και σηματοθορυβικό λόγο 0dB



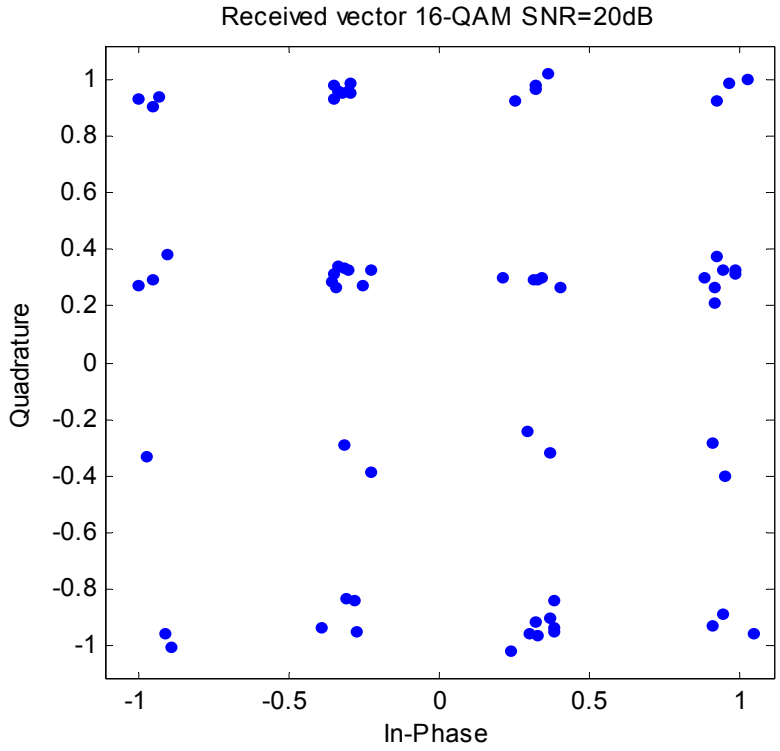
Σχήμα 3.16 Ληφθέντα σύμβολα για 4-QAM και σηματοθορυβικό λόγο 10dB



Σχήμα 3.17 Ληφθέντα σύμβολα για 16-QAM και σηματοθορυβικό λόγο 0dB



Σχήμα 3.18 Ληφθέντα σύμβολα για 16-QAM και σηματοθορυβικό λόγο 10dB



Σχήμα 3.19 Ληφθέντα σύμβολα για 16-QAM και σηματοθορυβικό λόγο 20dB

Τα ίδια συμπεράσματα προκύπτουν και από την ανάλυση μας για διαμόρφωση 16-QAM που φαίνονται στα σχήματα 3.17-3.19. Είναι φανερό και εδώ ότι με την καλυτέρευση του σηματοθορυβικού λόγου, οι περιοχές που καλύπτουν τα ληφθέντα σύμβολα γίνονται ολοένα και πιο εντοπισμένες γύρω από τα αρχικά σύμβολα οδηγώντας έτσι τον αποδιαμορφωτή σε χαμηλοτερα ποσοστά λάθους εκτίμησης.

Συγκρίνοντας τώρα τη διαμόρφωση 16-QAM με τη διαμόρφωση 4-QAM, παρατηρούμε ότι η 4-QAM έχει καλύτερες ιδιότητες απόστασης. Με άλλα λόγια, για ίδιο SNR, ένα σύμβολο 4-QAM είναι λιγότερο πιθανό να μεταβεί σε περιοχή άλλου συμβόλου από ότι ένα σύμβολο 16-QAM. Συνεπώς, η διαμόρφωση 4-QAM έχει μικρότερες τιμές πιθανότητας λάθους σε σχέση με τη 16-QAM. Το κέρδος τώρα από τη χρήση 16-QAM είναι η καλύτερη φασματική απόδοση κάτι που διαφαίνεται και από το παράδειγμα μας. Και στις δύο περιπτώσεις τα ληφθέντα σύμβολα που παριστάνονται στα σχήματα αντιστοιχούν σε μία κωδική λέξη. Στην περίπτωση 4-QAM ο αριθμός των συμβόλων είναι $(256 / \log_2 4)=128$ ενώ στην περίπτωση 16-QAM ο αριθμός των συμβόλων είναι $(256 / \log_2 16)=64$, υποδιπλασιάζεται λοιπόν το καταναλισκόμενο εύρος ζώνης. Η σύγκριση που έγινε σε αυτή την παράγραφο μπορεί να γενικευτεί για διαμορφώσεις χαμηλής και υψηλής φασματικής απόδοσης.

Αξίζει να σημειώσουμε ότι στα σύγχρονα τηλεπικοινωνιακά συστήματα χρησιμοποιείται μια τεχνική, γνωστή ως **προσαρμοστική διαμόρφωση**, που προσπαθεί να εκμεταλευτεί τα θετικά και των δύο κατηγοριών διαμόρφωσης, χαμηλής και υψηλής φασματικής απόδοσης. Η τεχνική αυτή συνίσταται στη χρήση διαμορφώσεων υψηλής φασματικής απόδοσης κατά τη διάρκεια ευνοϊκών συνθηκών στο δίαυλο και μετάβαση σε περισσότερο ανθεκτικά / εύρωστα σχήματα μικρότερης φασματικής απόδοσης κατά τη διάρκεια δυσμενών συνθηκών στο δίαυλο.

3.5.6 Αποκωδικοποιητής

Τα bits που εξέρχονται από τον αποδιαμορφωτή οδηγούνται στην είσοδο του LDPC(256, 128) αποκωδικοποιητή. Ο αποκωδικοποιητής τα λαμβάνει σε πακέτα των 256 bits όσο είναι το μήκος δηλαδή της κωδικής λέξης του κώδικα που έχουμε ορίσει. Η διαδικασία αποκωδικοποίησης εφαρμόζεται σε κάθε κωδική λέξη ξεχωριστά. Στο τέλος αυτής της διαδικασίας απομονώνονται τα bits της κωδικής λέξης που αντιστοιχούν στο μήνυμα πληροφορίας και που εξ ορισμού του κώδικα μας είναι 128.

Όσον αφορά τώρα την ουσία της διαδικασίας αποκωδικοποίησης αυτή θα παρουσιαστεί αναλυτικά στο κεφάλαιο 4 όπου θα γίνει εκτενής αναφορά στους LDPC κώδικες. Αυτό που μπορούμε να πούμε από τώρα είναι ότι ο αλγόριθμος αποκωδικοποίησης LDPC κωδίκων βρίσκει την βέλτιστη πιθανοτικά εκτίμηση για κάθε bit της κωδικής λέξης, όχι απαραίτητα όμως και την βέλτιστη εκτίμηση για την κωδική λέξη ως σύνολο.

3.5.7 Εκτιμητής Λαθών

Μετά το πέρας της αποκωδικοποίησης του εκάστοτε ληφθέντος κωδικοποιημένου μηνύματος, το σύστημα απομονώνει το μήνυμα πληροφορίας. Παρ' ότι ο στόχος και

του αποδιαμορφωτή και του αποκωδικοποιητή είναι προς την κατεύθυνση αυτό το μήνυμα να ταυτίζεται με το αρχικό μήνυμα, κάτι τέτοιο βεβαίως δεν συμβαίνει πάντα. Ήδη έχουμε αναφερθεί για παράδειγμα στους περιορισμούς πάνω στη διορθωτική ικανότητα των κωδίκων. Το στοιχείο της προσομοίωσης που ονομάζουμε εκτιμητής λαθών συγκρίνει το αρχικό μήνυμα πληροφορίας στον πομπό με το τελικό στο δέκτη και καταγράφει τα λάθη. Για να πετύχουμε το στόχο μας που είναι η εκτίμηση της πιθανότητας λάθους πρέπει όπως εξηγήσαμε να προσομοιώσουμε πολλά πακέτα και να λάβουμε τη μέση τιμή των σφαλμάτων που καταγράφονται. Προς αποφυγή παρεξηγήσεων να τονίσουμε ότι εκτιμητής λαθών στο δέκτη δεν μπορεί να υπάρξει σε πραγματικό σύστημα καθώς ο δέκτης δεν μπορεί να γνωρίζει με σιγουριά το αρχικό μήνυμα πληροφορίας...

4 Τεχνικές Κωδικοποίησης LDPC (Low Density Parity Check)

4.1 Γενικά

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τους **LDPC** (Low density parity-check) κώδικες, οι οποίοι είναι από τους πλέον αποτελεσματικούς κώδικες διόρθωσης λαθών και για τους οποίους σημαντική έρευνα εξελίσσεται στις μέρες μας.
Το βασικό χαρακτηριστικό των εν λόγω κωδίκων είναι ότι πρόκειται για γραμμικούς συμπαγείς κώδικες που κατασκευάζονται βάσει ενός αραιού πίνακα ελέγχου ισοτιμίας **H**, κάτι στο οποίο οφείλουν και την ονομασία τους. Όσον αφορά τη δυαδική περίπτωση (binary case) που θα μας απασχολήσει, αραιός πίνακας σημαίνει ότι ο **H** αποτελείται από σχετικά λίγους άσσους διάσπαρτους ανάμεσα σε πολλά μηδενικά ή με άλλα λόγια έχει χαμηλή πυκνότητα (low-density) άσσων. Η επίδοση των LDPC κωδίκων εξαρτάται σημαντικά από τη σχεδίαση του πίνακα **H** και από τον αλγόριθμο αποκωδικοποίησης που χρησιμοποιείται. Σε γενικές γραμμές μπορούμε να πούμε ότι η αποκωδικοποίηση των LDPC γίνεται χρησιμοποιώντας belief-propagation αλγόριθμο ο οποίος λειτουργεί επαναληπτικά, ανανεώνοντας σε κάθε βήμα την πιθανότητα κάθε bit να είναι 0 ή 1 λαμβάνοντας υπόψη την πληροφορία που απορρέει από την αντίστοιχη πιθανότητα των υπόλοιπων bits.

Ιστορικά τώρα, αυτή η κλάση κωδίκων με τα χαρακτηριστικά που προαναφέρθηκαν προτάθηκε για πρώτη φορά από τον Gallager το 1962. Η πολυπλοκότητα όμως του επαναληπτικού αλγορίθμου ξεπερνούσε τις υπάρχουσες τότε δυνατότητες των ηλεκτρονικών επεξεργαστών. Έτσι, αυτοί οι κώδικες ήταν ξεχασμένοι μέχρι το 1996 παρά την προσπάθεια του Tanner το 1981 να ξαναζωντανέψει το ενδιαφέρον για αυτούς. Ήταν το 1996 λοιπόν, όταν οι MacKay και Neal επανέφεραν τους LDPC κώδικες στο προσκήνιο και τώρα πια μιλάμε για κώδικες που αγγίζουν το θεωρητικό όριο του Shannon περί της επίδοσης των σχημάτων κωδικοποίησης καναλιού για διόρθωση λαθών, κάτι που ισχύει μόνο για τους LDPC και τους turbo κώδικες. Μάλιστα, έχει δειχθεί [8] ότι κατάλληλα σχεδιασμένοι LDPC κώδικες ξεπερνούν την απόδοση των turbo κωδίκων που χρησιμοποιούνται στα συστήματα κινητών επικοινωνιών $3^{ης}$ γενιάς (3G).

Η εξαιρετική επίδοση των LDPC κωδίκων έχει οδηγήσει στην υιοθέτηση τους από διάφορα πρότυπα επικοινωνιών, με πιο χαρακτηριστικά τα DVB-S2, IEEE80216.e, IEEE802.3an(10BASE-T) κ.ά.. Επιπλέον, αναμένεται να χρησιμοποιηθούν και σε διάφορα σύγχρονα συστήματα επικοινωνιών, όπως τα συστήματα πολλαπλών εισόδων - πολλαπλών εξόδων (MIMO) που μπορούν να επιτύχουν σημαντική αύξηση της χωρητικότητας. Σε κάθε περίπτωση βέβαια και παρά τη δεδομένη αποδοτικότητα των LDPC κωδίκων πρέπει να γίνεται ειδικός έλεγχος για διάφορους περιοριστικούς παράγοντες, όπως πολυπλοκότητα, απαιτήσεις μνήμης, μέγεθος κώδικα, συμβατότητα ρυθμού. Κατά συνέπεια, έχουν προταθεί διάφορες μέθοδοι κατασκευής και διάφορες τροποποιήσεις του βασικού αλγορίθμου αποκωδικοποίησης. Στη συνέχεια, θα δείξουμε τα βασικά χαρακτηριστικά των LDPC κωδίκων και στο επόμενο κεφάλαιο θα παρουσιάσουμε τα αποτελέσματα από την προσομοίωση μας που στόχο έχει να μελετήσει την επίδοση τους.

4.2 Κωδικοποίηση LDPC

4.2.1 Περιγραφή LDPC κωδίκων

Οι LDPC κώδικες σχεδιάζονται συνήθως ως γραμμικοί και δυαδικοί συμπαγείς κώδικες. Σε τέτοια περίπτωση, κατά τα γνωστά ένας γεννήτορας πίνακας **G** μετατρέπει το μήνυμα **m** μήκους k στο κωδικοποιημένο διάνυσμα **c** μήκους n μέσω πολλαπλασιασμού πινάκων στο δυαδικό πεδίο. Ισχύει δηλαδή $c = m \circ G$, έτσι ώστε το κωδικοποιημένο διάνυσμα **c** να ικανοποιεί την εξίσωση συνδρόμου $c \circ H^T = 0$. Εντελώς ισοδύναμη είναι η περίπτωση κωδικοποίησης κατά την οποία $c = G^T \circ m$ και $H \circ c = 0$.

Όπως προαναφέρφηκε, οι LDPC κώδικες σχεδιάζονται μέσω κατάλληλης κατασκευής του πίνακα **H**. Διακρίνουμε δύο μεγάλες κατηγορίες με βάση αυτόν τον πίνακα: τους κανονικούς (regular) κώδικες που χαρακτηρίζονται από σταθερό αριθμό άσσων ανά γραμμή και ανά στήλη και τους μη- κανονικούς (irregular) που δεν πληρούν την παραπάνω συνθήκη. Η πρώτη κατηγορία ήταν αυτή που πρωτοπροτάθηκε από τον Gallager ο οποίος ανέδειξε και τις βασικές ιδιότητες αυτών των κωδίκων: η πιθανότητα λάθους μειώνεται εκθετικά για αυξανόμενο μέγεθος κώδικα n και η ελάχιστη απόσταση αυξάνει με την αύξηση του μεγέθους του κώδικα.

4.2.2 Κανονικοί (regular) LDPC κώδικες

Κανονικοί ονομάζονται οι κώδικες που έχουν σταθερό αριθμό άσσων ανά γραμμή u και σταθερό αριθμό άσσων ανά στήλη s, όσον αφορά τον πίνακα **H**. Ένας τέτοιος LDPC κώδικας συμβολίζεται $C_{LDPC}(n, s, u)$. Λόγω της αραιότητας του **H**, οι παράμετροι s, u είναι μικροί αριθμοί συγκρινόμενοι με το μήκος κώδικα n.

Σαν παράδειγμα για καλύτερη κατανόηση των παραπάνω παραθέτουμε παρακάτω τον πίνακα Η ενός κώδικα $C_{LDPC}(12, 3, 6)$

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

Δεδομένου ότι ο **H** είναι διαστάσεων (n-k) x n και σε περίπτωση που όλες οι γραμμές του είναι γραμμικώς ανεξάρτητες μεταξύ τους, ισχύει η ακόλουθη σχέση για τον ρυθμό κώδικα k/n που προκύπτει από τις δύο διαφορετικές εκφράσεις για το συνολικό αριθμό άσσων στον πίνακα:

$$ns = (n-k)u \Rightarrow \frac{k}{n} = \frac{u-s}{u} \tag{4.1}$$

Σε περίπτωση που υπάρχουν γραμμικώς εξαρτημένες γραμμές ο ρυθμός κώδικα δίνεται από τη σχέση:

$$code\_rate = \frac{n-s'}{n} \tag{4.2}$$

όπου s΄ η πραγματική διάσταση του υποχώρου γραμμών του πίνακα **H**.

Οι μέχρι τώρα περιορισμοί που θέσαμε για τους κανονικούς LDPC κώδικες δεν οδηγούν συνήθως στη σχεδίαση ενός πίνακα **H** σε συστηματική μορφή. Έτσι, συνήθως απαιτείται η χρήση της μεθόδου απαλοιφής Gauss ώστε να γίνει μετατροπή σε συστηματικό πίνακα ελέγχου ισοτιμίας, πίνακα δηλαδή της μορφής:
$H' = [I_{n-k} \; P^T]$, όπου $\mathbf{I}_{\text{n-k}}$ ο μοναδιαίος πίνακας διαστάσεων (n-k) x (n-k). Ο πίνακας ελέγχου ισοτιμίας του κώδικα είναι ο αρχικά σχεδιασμένος **H** και ο γεννήτορας πίνακας **G** έχει την ακόλουθη μορφή που προκύπτει από τον **H':**
G=[P $\mathbf{I}_{\text{k}}$]
Αναλύοντας τώρα τη σχεδιαστική πορεία ενός LDPC κώδικα, αρχικά κατασκευάζεται ένας αραιός πίνακας ελέγχου ισοτιμίας **H= [A B]** που υπακούει στους σχεδιαστικούς περιορισμούς. Στη γενική περίπτωση, αυτός ο πίνακας δεν είναι σε συστηματική μορφή. Οι υποπίνακες **A** και **B** είναι και οι δύο αραιοί. Ο **A** είναι διαστάσεων (n-k) x (n-k) και πρέπει να είναι αντιστρέψιμος, με αντίστροφο έστω $\mathbf{A}^{-1}$. Ο **B** κατά συνέπεια είναι διαστάσεων (n-k) x k.
Η μέθοδος απαλοιφής Gauss, πάνω στο δυαδικό πεδίο (binary field), μετατρέπει τον πίνακα **H= [A B]** στη μορφή **H' = [** $\mathbf{I}_{\text{n-k}}$ $\mathbf{A}^{-1}$ **B] = [**$\mathbf{I}_{\text{n-k}}$ $\mathbf{P}^{\mathbf{T}}$**].** Από τη στιγμή που σχηματιστεί ο ισοδύναμος πίνακας ελέγχου ισοτιμίας **H',** ο αντίστοιχος γεννήτορας πίνακας **G** μπορεί να σχηματιστεί με τη βοήθεια των υποπινάκων του **H'** και να δώσει **G=[P $\mathbf{I}_{\text{k}}$].** Το τελευταίο βήμα προέκυψε από τη βασική σχέση που συνδέει τον γεννήτορα πίνακα με τον πίνακα ελέγχου ισοτιμίας στη θεωρία κωδικοποίησης:

$$G \circ H^T = 0$$

Έτσι, μετά από τα παραπάνω έχουμε ορίσει και τον πίνακα ελέγχου ισοτιμίας και τον γεννήτορα πίνακα˙ ο LDPC κώδικας λοιπόν έχει σχεδιασθεί. Σε αυτό το σημείο να σημειώσουμε ότι οι πίνακες ενδιαφέροντος είναι οι **H, G** με τον **H'** να έχει καθαρά βοηθητικό ρόλο.
Ανάλογα τώρα με τη μέθοδο που χρησιμοποείται για την κατασκευή του **H** οι LDPC κώδικες χωρίζονται σε τυχαίους και δομημένους. Γενικά, οι τυχαίοι κώδικες εμφανίζουν καλύτερες επιδόσεις όσον αφορά το ποσοστό λαθών BER, έχουν όμως πολυπλοκότερη κωδικοποίηση.

4.2.3 Μη-Κανονικοί (Irregular) LDPC κώδικες

Μη-κανονικοί ονομάζονται οι οι LDPC κώδικες με μεταβλητό αριθμό μονάδων ανά γραμμή ή ανά στήλη του **H**. Κατάλληλα σχεδιασμένοι μη-κανονικοί

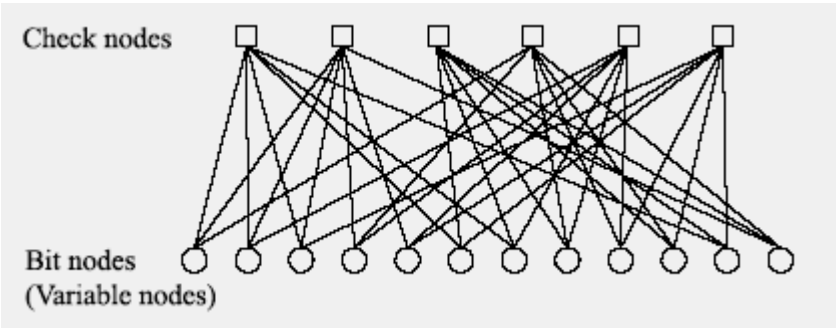
κώδικες εμφανίζουν γενικά μικρότερες τιμές BER σε σχέση με τους κανονικούς. Υπάρχουν διάφορες μέθοδοι σχεδιασμού αυτής της κατηγορίας κωδίκων [10].

4.3 Αποκωδικοποίηση LDPC

4.3.1 Ο γράφος Tanner

Ένα χρήσιμο εργαλείο αναπαράστασης ενός LDPC κώδικα είναι ο γράφος Tanner, ένας διμερής γράφος που αναπαριστά τις εξισώσεις ισοτιμίας (parity equations) που προκύπτουν από τον πίνακα **H.** Ο πίνακας ελέγχου ισοτιμίας **H** οφείλει την ονομασία του στο γεγονός ότι εκτελεί m=n-k ξεχωριστούς ελέγχους ισοτιμίας ανά λαμβανόμενη κωδικοποιημένη λέξη.

Ο γράφος Tanner που αντιστοιχεί σε έναν (n, k) κώδικα αποτελείται από n κόμβους bit (bit nodes), από m=n-k κόμβους ελέγχου (check nodes) και από έναν αριθμό ακμών ανάμεσα στις δύο αυτές κατηγορίες κόμβων. Κάθε κόμβος bit αναπαριστά ένα bit της κωδικοποιημένης λέξης. Κάθε κόμβος ελέγχου αναπαριστά έναν έλεγχο ισοτιμίας. Ακμή μεταξύ κόμβου bit και κόμβου ελέγχου υπάρχει αν και μόνο αν υπάρχει "1" στην αντίστοιχη θέση του **H.** Πιο φορμαλιστικά, εάν $H_{ij}=1$ τότε και μόνο τότε υπάρχει σύνδεση ανάμεσα στον κόμβο bit j και στον κόμβο ελέγχου i. Σε αυτό το σημείο παραθέτουμε τον αντίστοιχο γράφο Tanner για τον πίνακα **H** της ενότητας **4.2.2**

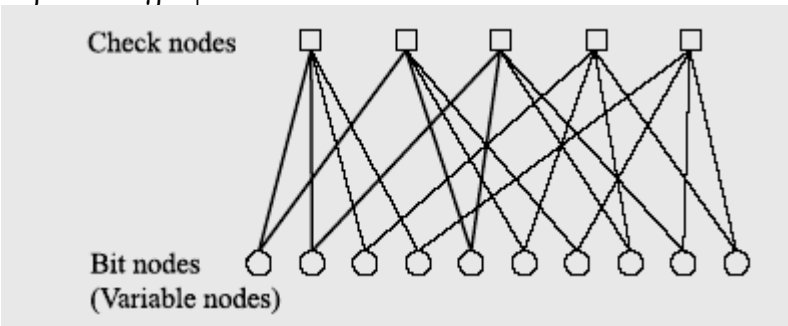


Σχήμα 4.1 Γράφος Tanner για τον πίνακα H της ενότητας 4.2.2

Ουσιαστικά, όπως εξηγήσαμε πρόκειται για γραφική αναπαράσταση των εξισώσεων ισοτιμίας του προς ανάλυση πίνακα, που είναι οι ακόλουθες:

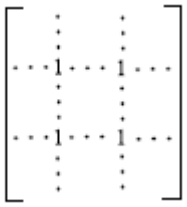
$$c_1 \oplus c_2 \oplus c_3 \oplus c_6 \oplus c_7 \oplus c_{11} = 0$$
$$c_1 \oplus c_2 \oplus c_3 \oplus c_4 \oplus c_5 \oplus c_{12} = 0$$
$$c_6 \oplus c_7 \oplus c_8 \oplus c_{10} \oplus c_{11} \oplus c_{12} = 0$$
$$c_1 \oplus c_4 \oplus c_8 \oplus c_9 \oplus c_{10} \oplus c_{12} = 0$$
$$c_2 \oplus c_4 \oplus c_5 \oplus c_7 \oplus c_8 \oplus c_9 = 0$$
$$c_3 \oplus c_5 \oplus c_6 \oplus c_9 \oplus c_{10} \oplus c_{11} = 0$$

Σε ένα γράφο Tanner, κύκλος μήκους ν ονομάζεται ένα μονοπάτι που περιλαμβάνει ν διαφορετικές ακμές και αρχίζει και τελειώνει στον ίδιο κόμβο. Το ελάχιστο μήκος κύκλου αναφέρεται και ως *girth. Για παράδειγμα το girth* του παρακάτω γράφου είναι 6.



Σχήμα 4.2 Γράφος Tanner ενός (10, 5) LDPC κώδικα με girth ίσο με 6

Γενικά, είναι φανερό ότι η ελάχιστη τιμή που μπορεί να πάρει το girth σε γράφο τύπου Tanner είναι 4. Σε τέτοια περίπτωση, τέσσερις άσσοι βρίσκονται στις γωνίες ενός υποπίνακα του **H** όπως δείχνουμε και στο επόμενο σχήμα



Σχήμα 4.3 Παράδειγμα πίνακα ελέγχου ισοτιμίας με κύκλο μήκους 4

Οι καλύτερες επιδόσεις των LDPC κωδίκων παρατηρούνται όταν συνδυάζονται με belief-propagation αλγόριθμο αποκωδικοποίησης, όπου η πιθανότητα κάθε bit διαδίδεται μέσω των ακμών και χρησιμοποιείται ως εξωτερική πληροφορία για την ανανέωση της πιθανότητας άλλων bits. Έχει δειχθεί ότι όσο πιο 'εξωτερική' είναι αυτή η πληροφορία τόσο αυξάνει η επίδοση του κώδικα. Σε περίπτωση κύκλου μικρού μήκους, η πληροφορία επιστρέφει σχετικά γρήγορα στον αρχικό κόμβο περιέχοντας μικρό ποσό εξωτερικής πληροφορίας κάτι που δεν βοηθά στην αποδοτικότητα του κώδικα. Κατά συνέπεια, το ελάχιστο μήκος κύκλου αποτελεί σημαντικό σχεδιαστικό παράγοντα για τους LDPC κώδικες και αρκετή ερευνητική προσπάθεια καταβάλλεται για την κατασκευή κωδίκων με μεγάλο girth.

Από τα παραπάνω κάποιος θα μπορούσε να οδηγηθεί στο συμπέρασμα ότι η ιδανική περίπτωση θα ήταν γράφος δενδρικής δομής, χωρίς κύκλους δηλαδή. Είναι αλήθεια ότι σε τέτοια περίπτωση ο belief-propagation (BP) επαναληπτικός αλγόριθμος αποκωδικοποίησης που θα εξηγηθεί στη συνέχεια τερματίζει πάντα μετά από πεπερασμένο αριθμό επαναλήψεων. Παραταύτα, κώδικες με ακυκλικούς γράφους εμφανίζουν χαμηλές τιμές BER εξαιτίας της μικρής ελάχιστης απόστασης τους: η ελάχιστη απόσταση τους είναι δύο για ρυθμούς κώδικα R > 1/2 (k/n > 1/2). Έτσι, η παρουσία κύκλων σε αποδοτικούς από όλες τις απόψεις LDPC κώδικες είναι επιτακτική. Αυτό που μπορεί και πρέπει να γίνεται είναι η αφαίρεση κύκλων μικρού μήκους (4, 6, 8 κλπ) ή τουλάχιστον η κατά το δυνατόν ελάττωση τους. Τελειώνοντας την ανάλυση μας για την σχέση του γράφου Tanner ενός LDPC κώδικα με τις ιδιότητες που αυτός εμφανίζει, αξίζει να αναφέρουμε ότι η αρνητική επιρροή κύκλων μικρού μήκους μειώνεται για αυξανόμενο μέγεθος κώδικα και μειώνεται δραστικά για μεγάλα μεγέθη κώδικα (> 1000 bits).

4.3.2 Ο αλγόριθμος BP

Στη συνέχεια, θα περιγράψουμε τον βασικό αλγόριθμο αποκωδικοποίησης LDPC κωδίκων: τον αλγόριθμο belief-propagation (BP), γνωστό και ως sum-product αλγόριθμο. Στην πλειονότητα τους οι χρησιμοποιούμενοι αλγόριθμοι αποκωδικοποίησης LDPC κωδίκων είναι παραλλαγές ή απλοποιήσεις του βασικού αλγορίθμου που θα παρουσιάσουμε παρακάτω.

Στόχος του αλγορίθμου είναι να προσδιορίσει κατά βέλτιστο τρόπο την εκ των υστέρων (a posteriori) πιθανότητα κάθε bit να είναι 0 ή 1, γνωρίζοντας το σήμα που έχει λάβει ο δέκτης, τα χαρακτηριστικά του κώδικα που σε αυτή την περίπτωση εκφράζονται ως εξισώσεις ισοτιμίας καθώς και τα χαρακτηριστικά του τηλεπικοινωνιακού διαύλου της διαδρομής από τον πομπό στο δέκτη. Σε αυτό το σημείο να τονίσουμε ότι ο αλγόριθμος BP βρίσκει την καλύτερη εκτίμηση για κάθε bit της κωδικής λέξης που καταφθάνει, αλλά όχι απαραίτητα την καλύτερη εκτίμηση για την κωδική λέξη ως σύνολο. Αυτό είναι συνέπεια του ότι πρόκειται για a posteriori αλγόριθμο σε αντίθεση με τους αλγόριθμους μεγίστης πιθανοφάνειας, όπως ο αλγόριθμος Viterbi, που βελτιστοποιούν την αποκωδικοποίηση ολόκληρου του λαμβανόμενου κωδικοποιημένου διανύσματος.

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

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

$$P(h_i/d_j = x) = \sum_{d:d_j=x} P(h_i/d)P(d/d_j = x) \qquad (4.3)$$

Αυτή η πιθανότητα υπολογίζεται για όλα τα πιθανά αποκωδικοποιημένα διανύσματα **d** για τα οποία ικανοποιείται η i-οστή εξίσωση ελέγχου ισοτιμίας, με την προϋπόθεση ότι ο κόμβος bit j βρίσκεται στην κατάσταση x.

Για τον κόμβο ελέγχου ισοτιμίας h_i , η πληροφορία που θα σταλεί στον συνδεδεμένο με αυτόν κόμβο bit j υπολογίζεται για κάθε τιμή του x και δίνεται από τη σχέση:

$$R_{ij}^{x} = \sum_{d:d_j=x} P(h_i/d) \prod_{k \in N(i)\setminus j} Q_{ik}^{d_k} \qquad (4.4)$$

Στην παραπάνω έκφραση ο συμβολισμός N(i) αναπαριστά το σύνολο των δεικτών των κόμβων bit που είναι συνδεδεμένοι με τον h_i , ενώ ο συμβολισμός N(i) \ j αναπαριστά το ίδιο σύνολο με την αφαίρεση όμως του κόμβου bit j. Βλέπουμε δηλαδή ότι η πληροφορία που καταφθάνει στον κόμβο bit j είναι αμιγώς εξωτερική, δεν είναι συνάρτηση δηλαδή της τρέχουσας κατάστασης του κόμβου. Η πιθανότητα τώρα $P(h_i / d)$ να ικανοποιείται η h_i είναι 0 ή 1 για δοθέν διάνυσμα **d.** Η παράμετρος Q_{ij}^{x} αναφέρεται στην πιθανότητα ο κόμβος d_j να είναι στην κατάσταση x και στέλνεται από τον κόμβο bit d_j σε όλους τους συνδεδεμένους με αυτόν κόμβους ελέγχου h_i. Η τιμή της παραμέτρου ανανεώνεται σε κάθε βήμα του αλγορίθμου και από την πληροφορία που έχει σταλεί στον d_j από τους συνδεδεμένους με αυτόν κόμβους ελέγχου h_i μέσω της πιθανότητας R_{ij}^{x}, όπως φαίνεται από την παρακάτω σχέση:

$$Q_{ij}^{x} = a_{ij} f_j^{x} \prod_{k \in M(j)\setminus i} R_{kj}^{x} \qquad (4.5)$$

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

$$\sum_{x} Q_{ij}^{x} = 1 \qquad (4.6)$$

Έχοντας υπολογίσει τις πιθανότητες Q_{ij}^{x}, R_{ij}^{x} απομένει η εκτίμηση της τιμής κάθε bit ξεχωριστά. Κάτι τέτοιο γίνεται με τη βοήθεια της ακόλουθης σχέσης:

$$\hat{d}_j = \arg\max_{x} f_j^{x} \prod_{k \in M(j)} R_{kj}^{x}$$

Η τιμή d_j υπολογίζεται στη δυαδική περίπτωση που μας απασχολεί για x=0 και x=1 και αναλόγως με το ποια από τις δύο τιμές είναι μεγαλύτερη, τίθεται και η αντίστοιχη τιμή (0 ή 1) στο j-οστό bit του διανύσματος προς αποκωδικοποίηση. Η διαδικασία αυτή εκτελείται για όλες τις τιμές του j, έτσι ώστε να προκύψει εκτίμηση για το διάνυσμα **d.** Εάν $c \circ H^T = 0$, τότε το διάνυσμα **d** θεωρείται έγκυρο αποκωδικοποιημένο διάνυσμα και ο επαναληπτικός αλγόριθμος τερματίζει δίνοντας ως έξοδο το διάνυσμα **d.** Σε άλλη περίπτωση, η διαδικασία επαναλαμβάνεται εξαιρουμένης της φάσης αρχικοποίησης, μέχρις ώτου βρεθεί έγκυρο διάνυσμα ή μέχρι να φτάσει ο αλγόριθμος ένα άνω όριο επαναλήψεων. Στην τελευταία περίπτωση, το διάνυσμα που θα οδηγηθεί στην έξοδο του αποκωδικοποιητή θα έχει λάθη αλλά ο αποκωδικοποιητής θα έχει κάνει βέλτιστη εκτίμηση της κατάστασης κάθε bit. Το πώς το σύστημα θα κάνει διαχείριση του λανθασμένου πακέτου (π.χ. αίτηση για επανεκπομπή) εξαρτάται από την εκάστοτε υλοποίηση.

Συνοψίζοντας, μπορούμε να πούμε ότι ο BP αλγόριθμος για αποκωδικοποίηση LDPC κωδίκων έχει την παρακάτω γενική περιγραφή:

- Αρχικοποίηση: Υπολογίζονται οι εκ των προτέρων πιθανότητες κατάστασης κάθε bit με βάση τη ληφθείσα τιμή του σήματος που αντιστοιχεί σε κάθε bit και τα χαρακτηριστικά του διαύλου, έστω f_j^x. Αυτές οι τιμές χρησιμοποιούνται για την αρχικοποίηση των συντελεστών Q_{ij}^x, $Q_{ij}^x = f_j^x$. Επιπλέον, καθορίζεται ο μέγιστος δυνατός αριθμός επαναλήψεων του αλγορίθμου αποκωδικοποίησης, έστω max_iter.
- Επαναληπτική Αποκωδικοποίηση:

  Βήμα 1: Ροή πληροφορίας από κόμβους bit προς κόμβους ελέγχου
  Αποστολή προς κόμβους ελέγχου των συντελεστών Q_{ij}^x. Ενημέρωση των συντελεστών R_{ij}^x.

  Βήμα 2: Ροή πληροφορίας από κόμβους ελέγχου προς κόμβους bit
  Αποστολή προς κόμβους ελέγχου των συντελεστών R_{ij}^x. Ενημέρωση των συντελεστών Q_{ij}^x. Ενημέρωση της πιθανότητας κατάστασης κάθε bit. Με βάση το ποια τιμή τιμή είναι πιο πιθανή, κάθε bit παίρνει την τιμή 0 ή 1, οπότε παίρνουμε μια εκτίμηση για το όλον διάνυσμα, έστω vector_est.

  Βήμα 3: Έλεγχος εγκυρότητας
  1. Εάν $vector\_est \circ H^T = 0$, ο αλγόριθμος σταματά και το vector_est θεωρείται έγκυρο διάνυσμα.
  2. Αλλιώς, ο αλγόριθμος αρχίζει μια νέα επανάληψη αρχίζοντας από το Βήμα 1.
  3. Σε περίπτωση που ο αριθμός επαναλήψεων φτάσει την τιμή max_iter, ο αλγόριθμος τερματίζεται.

Σε αυτό το σημείο, παραθέτουμε τον παρακάτω πίνακα που θα βοηθήσει στην εύκολη ανάγνωση των συμβολισμών και στη συνέχεια

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

4.3.3 Ένα παράδειγμα εκτέλεσης BP αλγορίθμου

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

Ο κώδικας μας χαρακτηρίζεται από τους παρακάτω πίνακες **H, G**

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

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

Όπως αναφέραμε, πρωταρχική σχεδίαση γίνεται μόνο για τον **H**, οπότε ο **G** είναι απόρροια του **H** με διαδικασία που αναφέρεται σε προηγούμενο εδάφιο. Όπως προκύπτει από τις διαστάσεις του H (8 x 12), κάθε ακωδικοποίητο μήνυμα έχει μήκος m=12-8=4 και κάθε κωδικοποιημένο διάνυσμα έχει μήκος n=12. Συνεπώς, ο ρυθμός κώδικα, ένα πολύ σημαντικό μέγεθος για κάθε σχήμα κωδικοποίησης, είναι 1/3. Παρατηρούμε επίσης ότι ο συγκεκριμένος LDPC κώδικας είναι μη-κανονικός. Αξίζει επίσης να σημειώσουμε ότι οι χρησιμοποιούμενοι στην πράξη πίνακες έχουν πολύ μεγαλύτερο μέγεθος, όμως σε αυτή την ενότητα μας ενδιαφέρει η κατανόηση της λειτουργίας της κωδικοποίησης και της αποκωδικοποίησης και όχι τόσο η αποδοτική σχεδίαση κώδικα που θα μας απασχολήσει στην προσομοίωση μας.

Για αυτό το παράδειγμα, θεωρούμε μήνυμα **m**=(1 0 0 0). Το **m** κωδικοποιείται ως **c**, με βάση τη γνωστή μας σχέση $c = m \circ G$.
Έτσι, προκύπτει **c**=(1 1 1 1 1 0 0 0 1 0 0 0). Το κωδικοποιημένο διάνυσμα ακολούθως διαμορφώνεται προς μετάδοση με BPSK διαμόρφωση, οπότε σε πολική μορφή μεταδίδεται το παρακάτω διάνυσμα:
t = (+1 +1 +1 +1 +1 –1 –1 –1 +1 –1 –1 –1).
Θεωρώντας ότι το σήμα διέρχεται από δίαυλο που εισάγει προσθετικό λευκό θόρυβο Gauss τυπικής απόκλισης σ =0.8 και ως αποτέλεσμα της μετάδοσης και της διαδικασίας δειγματοληψίας φτάνει στην είσοδο του αποκωδικοποιητή το ακόλουθο διάνυσμα **r:**

$$r = (+1.3129 +2.6584 +0.7413 +2.1745 +0.5981 -0.8323 -0.3962 -1.7586 \\ +1.4905 +0.4084 -0.9290 +1.0765)$$

Εάν χρησιμοποιούνταν αποκωδικοποιητής σκληρής απόφασης (hard decision), το αποκωδικοποιημένο διάνυσμα θα ήταν:
d = (1 1 1 1 1 0 0 0 1 1 0 1)
Ο δίαυλος επικοινωνίας λοιπόν θα είχε εισάγει δύο λάθη στις θέσεις 10, 12 όπως φαίνεται από τη σύγκριση των διανυσμάτων **c**, **d.**

Αφού ο δίαυλος μας είναι τύπου AWGN, η έκφραση για την εκ των προτέρων πιθανότητα κατάστασης κάθε bit του ληφθέντος διανύσματος θα δίνεται από την γκαουσιανή συνάρτηση πυκνότητας πιθανότητας. Έτσι, συμβολίζοντας όπως και πριν με f_j^0 την πιθανότητα το bit στη θέση j του διανύσματος να είναι 0 και με f_j^1 την πιθανότητα το bit στη θέση j του διανύσματος να είναι 1, έχουμε τις παρακάτω εκφράσεις:

$$f_j^{\,0} = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r_j+1)^2/2\sigma^2} \tag{4.7}$$

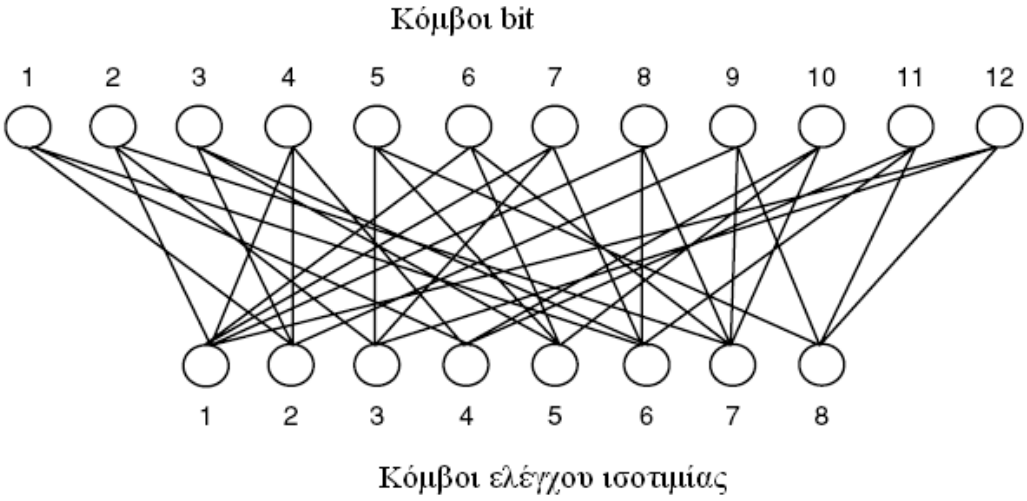
$$f_j^{\,1} = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r_j-1)^2/2\sigma^2} \tag{4.8}$$

Όπως φαίνεται από τις παραπάνω σχέσεις, πρέπει να γνωρίζουμε την τυπική απόκλιση σ του διαύλου. Υπάρχουν διάφορες τεχνικές με τις οποίες ο δέκτης μπορεί να κάνει εκτίμηση της τυπικής απόκλισης, όπως εύρεση της παραμέτρου σ μέσω της μεθόδου ελαχιστοποίησης του τετραγωνικού σφάλματος (MMSE). Στη συνέχεια, αλλά και στην προσομοίωση μας, θα θεωρούμε ότι το σύστημα του δέκτη γνωρίζει την τιμή της σταθεράς σ.
Με βάση τις παραπάνω σχέσεις τώρα για τα f_j^0 και f_j^1, παραθέτουμε τον παρακάτω πίνακα

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

Μέχρι τώρα, είμαστε στη φάση αρχικοποίησης του αλγορίθμου. Για την κατανόηση της επαναληπτικής διαδικασίας, παραθέτουμε τον γράφο Tanner του υπό εξέταση κώδικα και τις αντίστοιχες εξισώσεις ισοτιμίας, όπως προκύπτουν από τη μελέτη του πίνακα **H**.

Γράφος Tanner:



Σχήμα 4.4 Γράφος Tanner για το παράδειγμα μας

Εξισώσεις ισοτιμίας:

$$c_2 \oplus c_4 \oplus c_6 \oplus c_7 \oplus c_8 \oplus c_{12} = 0$$
$$c_1 \oplus c_3 \oplus c_4 \oplus c_9 = 0$$
$$c_2 \oplus c_5 \oplus c_7 \oplus c_{12} = 0$$
$$c_1 \oplus c_4 \oplus c_{10} \oplus c_{11} = 0$$
$$c_3 \oplus c_5 \oplus c_6 \oplus c_{10} = 0$$
$$c_1 \oplus c_3 \oplus c_7 \oplus c_8 \oplus c_{11} = 0$$
$$c_2 \oplus c_6 \oplus c_8 \oplus c_9 \oplus c_{10} = 0$$
$$c_5 \oplus c_9 \oplus c_{11} \oplus c_{12} = 0$$

Κάθε γραμμή του πίνακα **H** αντιστοιχεί και σε μία εξίσωση ισοτιμίας, συνεπώς και σε έναν κόμβο ελέγχου ισοτιμίας. Είναι φανερή η αντιστοιχία γράφου Tanner και εξισώσεων ισοτιμίας, π.χ. από την πρώτη εξίσωση ισοτιμίας:

$$c_2 \oplus c_4 \oplus c_6 \oplus c_7 \oplus c_8 \oplus c_{12} = 0$$

προκύπτει ότι ο κόμβος ελέγχου 1 είναι συνδεδεμένος με τους κόμβους bit 2, 4, 6, 7, 8, 12.

Όπως εξηγήσαμε και στην προηγούμενη ενότητα, έχουμε αρχικοποίηση των συντελεστών Q_{ij}^x ως $Q_{ij}^x = f_j^x$. Για το παράδειγμα μας λοιπόν, παίρνουμε:

$$
\begin{aligned}
Q^0_{12} &= 0.0000 & Q^1_{12} &= 0.0582 \\
Q^0_{14} &= 0.0002 & Q^1_{14} &= 0.1697 \\
Q^0_{16} &= 0.4878 & Q^1_{16} &= 0.0362 \\
Q^0_{17} &= 0.3751 & Q^1_{17} &= 0.1088 \\
Q^0_{18} &= 0.3181 & Q^1_{18} &= 0.0013 \\
Q^0_{1,12} &= 0.0172 & Q^1_{1,12} &= 0.4964
\end{aligned}
$$

$$
\begin{aligned}
Q^0_{21} &= 0.0076 & Q^1_{21} &= 0.4619 \\
Q^0_{23} &= 0.0467 & Q^1_{23} &= 0.4733 \\
Q^0_{24} &= 0.0002 & Q^1_{24} &= 0.1697 \\
Q^0_{29} &= 0.0039 & Q^1_{29} &= 0.4132
\end{aligned}
$$

$$
\begin{aligned}
Q^0_{32} &= 0.0000 & Q^1_{32} &= 0.0582 \\
Q^0_{35} &= 0.0678 & Q^1_{35} &= 0.4396 \\
Q^0_{37} &= 0.3751 & Q^1_{37} &= 0.1088 \\
Q^0_{3,12} &= 0.0172 & Q^1_{3,12} &= 0.4964
\end{aligned}
$$

$$
\begin{aligned}
Q^0_{41} &= 0.0076 & Q^1_{41} &= 0.4619 \\
Q^0_{44} &= 0.0002 & Q^1_{44} &= 0.1697 \\
Q^0_{4,10} &= 0.1059 & Q^1_{4,10} &= 0.3794 \\
Q^0_{4,11} &= 0.4967 & Q^1_{4,11} &= 0.0272
\end{aligned}
$$

$$
\begin{aligned}
Q^0_{53} &= 0.0467 & Q^1_{53} &= 0.4733\\
Q^0_{55} &= 0.0678 & Q^1_{55} &= 0.4396\\
Q^0_{56} &= 0.4878 & Q^1_{56} &= 0.0362\\
Q^0_{5,10} &= 0.1059 & Q^1_{5,10} &= 0.3794
\end{aligned}
$$

$$
\begin{aligned}
Q^0_{61} &= 0.0076 & Q^1_{61} &= 0.4619\\
Q^0_{63} &= 0.0467 & Q^1_{63} &= 0.4733\\
Q^0_{67} &= 0.3751 & Q^1_{67} &= 0.1088\\
Q^0_{68} &= 0.3181 & Q^1_{68} &= 0.0013\\
Q^0_{6,11} &= 0.4967 & Q^1_{6,11} &= 0.0272
\end{aligned}
$$

$$
\begin{aligned}
Q^0_{72} &= 0.0000 & Q^1_{72} &= 0.0582\\
Q^0_{76} &= 0.4878 & Q^1_{76} &= 0.0362\\
Q^0_{78} &= 0.3181 & Q^1_{78} &= 0.0013\\
Q^0_{79} &= 0.0039 & Q^1_{79} &= 0.4132\\
Q^0_{7,10} &= 0.1059 & Q^1_{7,10} &= 0.3794
\end{aligned}
$$

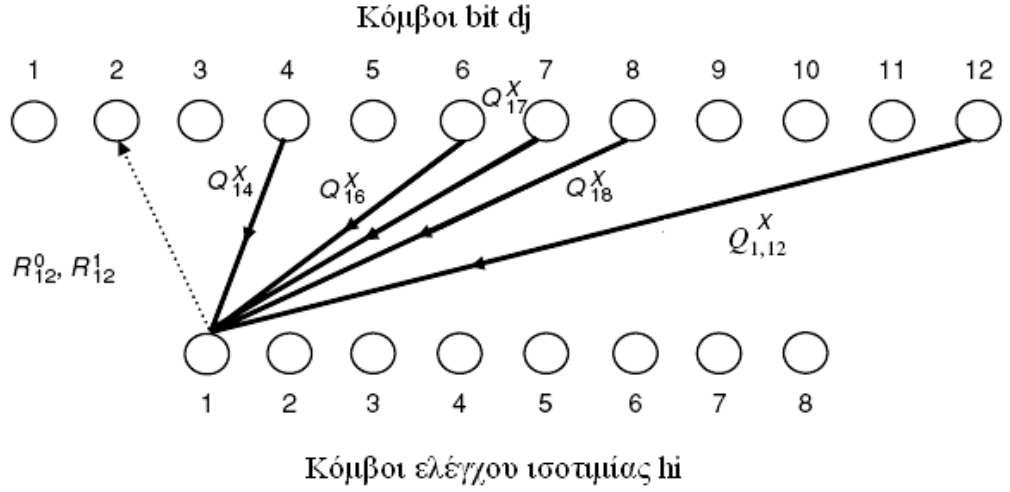
$$
\begin{aligned}
Q^0_{85} &= 0.0678 & Q^1_{85} &= 0.4396\\
Q^0_{89} &= 0.0039 & Q^1_{89} &= 0.4132\\
Q^0_{8,11} &= 0.4967 & Q^1_{8,11} &= 0.0272\\
Q^0_{8,12} &= 0.0172 & Q^1_{8,12} &= 0.4964
\end{aligned}
$$

Σε αυτό το σημείο έχει ολοκληρωθεί η φάση αρχικοποίησης του αλγορίθμου, η αρχικοποίηση δηλαδή των συντελεστών Q_{ij}^x. Όπως περιγράψαμε παραπάνω, επόμενο στάδιο είναι το Βήμα 1 του επαναληπτικού αλγορίθμου αποκωδικοποίησης, η ανάθεση τιμών δηλαδή στους συντελεστές R_{ij}^x. Επαναλαμβάνουμε ότι κάθε συντελεστής R_{ij}^x εκφράζει την πιθανότητα να ικανοποιείται η εξίσωση ισοτιμίας i δεδομένου ότι ο κόμβος bit j βρίσκεται στην κατάσταση x. Έτσι, στο παράδειγμα μας, η τιμή R_{12}^0 εκφράζει την πιθανοτική εκτίμηση να ικανοποιείται η εξίσωση ισοτιμίας 1, δηλαδή $c_2 \oplus c_4 \oplus c_6 \oplus c_7 \oplus c_8 \oplus c_{12} = 0$ όταν το bit στη θέση 2 του διανύσματος έχει την τιμή 0. Παρατηρούμε ότι η παραπάνω εξίσωση ικανοποιείται για οποιοδήποτε συνδυασμό των bits c_4, c_6, c_7, c_8, c_{12}, στον οποίο ο αριθμός των άσσων είναι άρτιος όπως προκύπτει από τους κανόνες της modulo-2 αριθμητικής. Θα υπολογίσουμε λοιπόν την πιθανότητα R_{12}^0 αθροίζοντας την πιθανότητα να ισχύει κάθε ένας από τους προαναφερθέντες συνδυασμούς. Για κάθε έναν από αυτούς τους συνδυασμούς, π.χ. τα bits c_4, c_6, c_7, c_8, c_{12} να είναι όλα 0, η πιθανότητα του συνδυασμού ισούται με το γινόμενο των πιθανοτήτων κάθε bit να είναι στην κατάσταση που εξετάζουμε λόγω ανεξαρτησίας. Όπως ήδη ξέρουμε, η πιθανότητα το

bit j να είναι στην κατάσταση x εκφράζεται μέσω του συντελεστή Q_{ij}^{x}. Είμαστε σε θέση τώρα να υπολογίσουμε την τιμή R_{12}^{0} ως εξής:

$$R_{12}^{0} = Q_{14}^{0}Q_{16}^{0}Q_{17}^{0}Q_{18}^{0}Q_{1,12}^{0} + Q_{14}^{0}Q_{16}^{0}Q_{17}^{0}Q_{18}^{1}Q_{1,12}^{1} + Q_{14}^{0}Q_{16}^{0}Q_{17}^{1}Q_{18}^{0}Q_{1,12}^{1} +$$
$$Q_{14}^{0}Q_{16}^{0}Q_{17}^{1}Q_{18}^{1}Q_{1,12}^{0} + Q_{14}^{0}Q_{16}^{1}Q_{17}^{0}Q_{18}^{0}Q_{1,12}^{1} + Q_{14}^{0}Q_{16}^{1}Q_{17}^{0}Q_{18}^{1}Q_{1,12}^{0} + Q_{14}^{0}Q_{16}^{1}Q_{17}^{1}Q_{18}^{0}Q_{1,12}^{0}$$
$$+Q_{14}^{0}Q_{16}^{1}Q_{17}^{1}Q_{18}^{1}Q_{1,12}^{1} + Q_{14}^{1}Q_{16}^{0}Q_{17}^{0}Q_{18}^{0}Q_{1,12}^{1} + Q_{14}^{1}Q_{16}^{0}Q_{17}^{0}Q_{18}^{1}Q_{1,12}^{0} + Q_{14}^{1}Q_{16}^{0}Q_{17}^{1}Q_{18}^{0}Q_{1,12}^{0}$$
$$+Q_{14}^{1}Q_{16}^{0}Q_{17}^{1}Q_{18}^{1}Q_{1,12}^{1} + Q_{14}^{1}Q_{16}^{1}Q_{17}^{0}Q_{18}^{0}Q_{1,12}^{0} + Q_{14}^{1}Q_{16}^{1}Q_{17}^{0}Q_{18}^{1}Q_{1,12}^{1} + Q_{14}^{1}Q_{16}^{1}Q_{17}^{1}Q_{18}^{0}Q_{1,12}^{1}$$
$$+Q_{14}^{1}Q_{16}^{1}Q_{17}^{1}Q_{18}^{1}Q_{1,12}^{0}$$
$$= 0.0051$$

Με παρόμοια ανάλυση προκύπτει ότι η τιμή R_{12}^{1} προκύπτει για όλους τους συνδυασμούς με περιττό αριθμό άσσων. Στη συνέχεια αποδίδουμε σχηματικά την ανταλλαγή πληροφορίας που συντελείται για τον υπολογισμό των R_{12}^{0}, R_{12}^{1} και κατ' επέκταση δίνουμε στον αναγνώστη να καταλάβει τη διαδικασία υπολογισμού των συντελεστών R_{ij}^{x} εν γένει.



Σχήμα 4.5 Σχηματική απεικόνιση της διαδικασίας υπολογισμού των συντελεστών R_{12}^{0}, R_{12}^{1}

Στους παρακάτω πίνακες δίνουμε τις τιμές των R_{ij}^{0} , R_{ij}^{1} αντίστοιχα κατά την πρώτη επανάληψη.

Πίνακας 4.1 Τιμές των συντελεστών R_{ij}^0 κατά την πρώτη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.0051		0.0017		0.0002	0.0001	0.0004				0.0006
2	0.0036		0.0009	0.0113					0.0043			
3		0.0868			0.0109		0.0024					0.0100
4	0.0325			0.0889						0.0390	0.0088	
5			0.0875		0.0925	0.0423				0.1049		
6	0.0126		0.0100				0.0348	0.0431			0.0270	
7		0.0250				0.0008		0.0016	0.0035	0.0037		
8					0.1022				0.1101		0.0179	0.0912

Πίνακας 4.2 Τιμές των συντελεστών R_{ij}^1 κατά την πρώτη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.0020		0.0007		0.0006	0.0008	0.0009				0.0002
2	0.0332		0.0324	0.0906					0.0372			
3		0.0393			0.0035		0.0128					0.0043
4	0.0107			0.0305						0.0028	0.0299	
5			0.0416		0.0398	0.0857				0.0333		
6	0.0295		0.0280				0.0060	0.0188			0.0107	
7		0.0089				0.0029		0.0046	0.0012	0.0003		
8					0.0101				0.0264		0.0908	0.0197

Με βάση τη γνωστή μας σχέση:

$$\hat{d}_j = \arg\max_x f_j^x \prod_{k \in M(j)} R_{kj}^x$$

μπορούμε να δώσουμε την εκτίμηση που προκύπτει για το διάνυσμα κατά την πρώτη επανάληψη.
Για παράδειγμα για το bit 1 παίρνουμε:

$$\hat{d}_1 = \left\{ \begin{array}{l} \hat{0} \rightarrow f_1^0 \times R_{21}^0 \times R_{41}^0 \times R_{61}^0 = 1.13 \times 10^{-8} \\ \hat{1} \rightarrow f_1^1 \times R_{21}^1 \times R_{41}^1 \times R_{61}^1 = 4.85 \times 10^{-6} \end{array} \right\} \Rightarrow '1'$$

καθώς η πιθανότητα να είναι 1 βρέθηκε μεγαλύτερη από την πιθανότητα να είναι 0.
Κάνοντας την ίδια διαδικασία για όλα τα bits παίρνουμε την πρώτη εκτίμηση για το αποκωδικοποιημένο διάνυσμα:

$$\mathbf{d} = (1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0)$$

Συγκρίνοντας την εκτίμηση με το αρχικό διάνυσμα **c**=(1 1 1 1 1 0 0 0 1 0 0 0) βλέπουμε ότι υπάρχουν τρία λάθη. Σε αυτή τη φάση ο αλγόριθμος κάνει έλεγχο εγκυρότητας, ή όπως περιγράψαμε παραπάνω ελέγχει εάν το σύνδρομο $\mathbf{d} \circ H^T = \mathbf{0}$. Από την στιγμή που το σύνδρομο δεν είναι το μηδενικό διάνυσμα, μια νέα επανάληψη αρχίζει.
Σε πρώτη φάση στέλνονται προς τους κόμβους ελέγχου οι ανανεωμένες τιμές των συντελεστών Q_{ij}^x. Οι τιμές αυτές έχουν υπολογιστεί από τη σχέση

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

Για παράδειγμα,

$$Q_{12}{}^{0} = a_{12} f_2{}^{0} R_{32}{}^{0} R_{72}{}^{0}$$

και

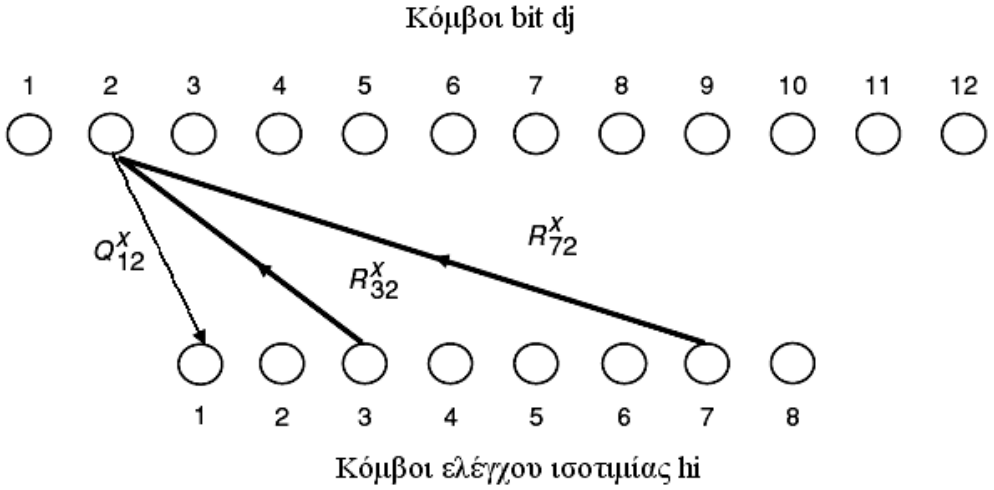
$$Q_{12}{}^{1} = a_{12} f_2{}^{1} R_{32}{}^{1} R_{72}{}^{1}$$

Υπενθυμίζουμε ότι στον παραπάνω συμβολισμό R_{kj}, το k αντιστοιχεί σε όλους τους κόμβους ελέγχου ισοτιμίας που είναι συνδεδεμένοι με τον κόμβο bit d_j, πλην του κόμβου i. Επιπλέον, η κανονικοποιητική σταθερά a_{12} εισάγεται με το σκοπό να ικανοποιείται η εξίσωση:

$$Q_{12}{}^{0} + Q_{12}{}^{1} = 1, \text{ οπότε προκύπτει}$$

$$a_{12} = \frac{1}{f_2{}^{0} R_{32}{}^{0} R_{72}{}^{0} + f_2{}^{1} R_{32}{}^{1} R_{72}{}^{1}}$$

Και σε αυτή την περίπτωση, δίνουμε τη σχηματική παράσταση, με τη βοήθεια του διμερούς γράφου, μέσω της οποίας γίνεται ο υπολογισμός των συντελεστών $Q_{12}{}^{0}$, $Q_{12}{}^{1}$ και κατ' επέκταση όλων των συντελεστών $Q_{ij}{}^{x}$.



Σχήμα 4.6 Σχηματική απεικόνιση της διαδικασίας υπολογισμού των συντελεστών $Q_{12}{}^{0}$, $Q_{12}{}^{1}$

Έτσι λοιπόν, στο Βήμα 1 του επαναληπτικού αλγορίθμου κατά τη δεύτερη επανάληψη 'στέλνονται' οι ακόλουθες τιμές των συντελεστών $Q_{ij}{}^{0}$, $Q_{ij}{}^{1}$ αντίστοιχα:

Πίνακας 4.3 Τιμές των συντελεστών Q_{ij}^{0} κατά τη δεύτερη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.0015		0.0004		0.6601	0.7898	0.9950				0.2731
2	0.0209		0.0691	0.0083					0.1014			
3		0.0018			0.7843		0.6946					0.3068
4	0.0008			0.0004						0.9091	0.9008	
5			0.0010		0.8294	0.5620				0.9777		
6	0.0054		0.0056				0.0688	0.9710			0.5147	
7		0.0014				0.6847		0.9954	0.0046	0.9239		
8					0.5273				0.0031		0.9316	0.1838

Πίνακας 4.4 Τιμές των συντελεστών Q_{ij}^{1} κατά τη δεύτερη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.9985		0.9996		0.3399	0.2102	0.0050				0.7269
2	0.9791		0.9309	0.9917					0.8986			
3		0.9982			0.2157		0.3054					0.6932
4	0.9992			0.9996						0.0909	0.0992	
5			0.9990		0.1706	0.4380				0.0223		
6	0.9946		0.9944				0.9312	0.0290			0.4853	
7		0.9986				0.3153		0.0046	0.9954	0.0761		
8					0.4727				0.9969		0.0684	0.8162

Αυτή τη φορά, και σε σύγκριση με την πρώτη επανάληψη, οι συντελεστές Q_{ij}^{x} δεν εμπεριέχουν μόνο πληροφορία διαύλου αλλά και πληροφορία κώδικα. Αυτό το χαρακτηριστικό προφανώς θα ενυπάρχει και στις επόμενες επαναλήψεις. Προχωρώντας στα βήματα της δεύτερης επανάληψης, υπολογίζουμε, μέσω των γενικών σχέσεων που ισχύουν κάθε φορά, τις τιμές των συντελεστών R_{ij}^{0} , R_{ij}^{1} τις οποίες και παραθέτουμε παρακάτω.

Πίνακας 4.5 Τιμές των συντελεστών R_{ij}^{0} κατά τη δεύτερη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.5416		0.5415		0.3704	0.4284	0.4581				0.5915
2	0.1622		0.1244	0.1709					0.0940			
3		0.4572			0.5750		0.6095					0.3897
4	0.1723			0.1726						0.8999	0.9081	
5			0.5390		0.4409	0.1859				0.4593		
6	0.5118		0.5118				0.5135	0.4876			0.1027	
7		0.3463				0.9150		0.6547	0.3453	0.6808		
8					0.7713				0.4851		0.5172	0.4766

Πίνακας 4.6 Τιμές των συντελεστών R_{ij}^{1} κατά τη δεύτερη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.4584		0.4585		0.6296	0.5716	0.5419				0.4085
2	0.8378		0.8756	0.8291					0.9060			
3		0.5428			0.4250		0.3905					0.6103
4	0.8277			0.8274						0.1001	0.0919	
5			0.4610		0.5591	0.8141				0.5407		
6	0.4882		0.4882				0.4865	0.5124			0.8973	
7		0.6537				0.0850		0.3453	0.6547	0.3192		
8					0.2287				0.5149		0.4828	0.5234

Αφού έχουν υπολογιστεί οι τιμές R_{ij}^{x} μπορεί να προκύψει η εκτίμηση του αποκωδικοποιημένου διανύσματος κατά τη δεύτερη επανάληψη. Ο επόμενος πίνακας περιγράφει αυτή τη διαδικασία.

r	+1.3129	+2.6584	+0.7413	+2.1745	+0.5981	−0.8323	−0.3962	−1.7586	+1.4905	+0.4084	−0.9290	+1.0765
t	+1	+1	+1	+1	+1	−1	−1	−1	+1	−1	−1	−1
d_j^0	0.0001	0.0000	0.0016	0.0000	0.0133	0.0307	0.0503	0.0465	0.0001	0.0298	0.0240	0.0019
d_j^1	0.1564	0.0095	0.0933	0.0534	0.0239	0.0016	0.0118	0.0001	0.1262	0.0066	0.0011	0.0648

Όπως προκύπτει από τη σύγκριση των τιμών d_j^0, d_j^1 το εκτιμώμενο αποκωδικοποιημένο διάνυσμα στο τέλος της δεύτερης απανάληψης είναι **d**=(1 1 1 1 1 0 0 0 1 0 0 1) που διαφέρει από το αρχικό κωδικοποιημένο διάνυσμα μόνο στο τελευταίο bit. Αυτή η διαφορά προκαλεί μη μηδενικό σύνδρομο για το **d,** οπότε μια νέα επανάληψη αρχίζει, η τρίτη κατά σειρά.
Κατά την τρίτη λοιπόν επανάληψη ο αποκωδικοποιητής καταφέρνει να διορθώσει όλα τα σφάλματα που είχε εισάγει ο δίαυλος και να δώσει στην έξοδο του μήνυμα ίδιο με το αρχικό μήνυμα **m.**
Παραθέτουμε πρώτα τις τιμές των συντελεστών Q_{ij}^0 , Q_{ij}^1 που μεταδίδονται μέσω του ακμών του διμερούς γράφου κατά την τρίτη επανάληψη.

Πίνακας 4.7 Τιμές των συντελεστών Q_{ij}^0 κατά την τρίτη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.0001		0.0000		0.9707	0.8503	0.9977				0.0197
2	0.0036		0.1078	0.0003					0.0047			
3		0.0002			0.2909		0.7318					0.0436
4	0.0033			0.0003						0.3358	0.6910	
5			0.0145		0.4130	0.9884				0.8426		
6	0.0007		0.0161				0.8013	0.9974			0.9948	
7		0.0002				0.6442		0.9949	0.0009	0.6807		
8					0.1413				0.0005		0.9538	0.0310

Πίνακας 4.8 Τιμές των συντελεστών Q_{ij}^1 κατά την τρίτη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.9999		1.0000		0.0293	0.1497	0.0023				0.9803
2	0.9964		0.8922	0.9997					0.9953			
3		0.9998			0.7091		0.2682					0.9564
4	0.9967			0.9997						0.6642	0.3090	
5			0.9855		0.5870	0.0116				0.1574		
6	0.9993		0.9839				0.1987	0.0026			0.0052	
7		0.9998				0.3558		0.0051	0.9991	0.3193		
8					0.8587				0.9995		0.0462	0.9690

Στη συνέχεια ακολουθούμε τη γνωστή διαδικασία ανανέωσης των τιμών των συντελεστών R_{ij}^0 , R_{ij}^1 τις οποίες και παραθέτουμε παρακάτω.

Πίνακας 4.9 Τιμές των συντελεστών R_{ij}^0 κατά την τρίτη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.8153		0.8153		0.1651	0.0501	0.1833				0.8282
2	0.1117		0.0085	0.1143					0.1109			
3		0.5885			0.7115		0.3092					0.5969
4	0.5627			0.5623						0.6896	0.3370	
5			0.4418		0.1750	0.5579				0.5825		
6	0.2129		0.2037				0.9758	0.7882			0.7897	
7		0.4485				0.6784		0.5520	0.4485	0.6424		
8					0.9252				0.8053		0.1639	0.8252

Πίνακας 4.10 Τιμές των συντελεστών R_{ij}^1 κατά την τρίτη επανάληψη

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.1847		0.1847		0.8349	0.9499	0.8167				0.1718
2	0.8883		0.9915	0.8857					0.8891			
3		0.4115			0.2885		0.6908					0.4031
4	0.4373			0.4377						0.3104	0.6630	
5			0.5582		0.8250	0.4421				0.4175		
6	0.7871		0.7963				0.0242	0.2118			0.2103	
7		0.5515				0.3216		0.4480	0.5515	0.3576		
8					0.0748				0.1947		0.8361	0.1748

Τέλος, δίνουμε και τον πίνακα που περιγράφει τη διαδικασία εκτίμησης του αποκωδικοποιημένου διανύσματος.

$\boldsymbol{r}$	+1.3129	+2.6584	+0.7413	+2.1745	+0.5981	−0.8323	−0.3962	−1.7586	+1.4905	+0.4084	−0.9290	+1.0765
$\boldsymbol{t}$	+1	+1	+1	+1	+1	−1	−1	−1	+1	−1	−1	−1
d_j^0	0.0001	0.0000	0.0000	0.0000	0.0077	0.0305	0.0057	0.0254	0.0002	0.0273	0.0217	0.0070
d_j^1	0.1412	0.0024	0.2086	0.0122	0.0078	0.0043	0.0017	0.0001	0.0394	0.0176	0.0032	0.0060

Έτσι λοιπόν προκύπτει εκτίμηση **d**=(1 1 1 1 1 0 0 0 1 0 0 0)=**c.** Το διάνυσμα **d** έχει μηδενικό σύνδρομο, οπότε ο αλγόριθμος τερματίζει επιτυχώς μετά την τρίτη επανάληψη. Ο αποκωδικοποιητής τότε οδηγεί στην έξοδο το αποκωδικοποιημένο μήνυμα **m**=(1 0 0 0), που ταυτίζεται με το αρχικό μήνυμα πληροφορίας.

4.4 Απλοποιήσεις του βασικού αλγορίθμου αποκωδικοποίησης

Στα προηγούμενα εδάφια, παρουσιάσαμε τη δομή και τη λειτουργία του βασικού αλγορίθμου αποκωδικοποίησης LDPC κωδίκων, γνωστός και ως BP ή sum-product αλγόριθμος. Στη συνέχεια, θα παρουσιάσουμε παραλλαγές του βασικού αλγορίθμου που στόχο έχουν την απλούστευση της υπολογιστικής διαδικασίας ή την ελάττωση της υπολογιστικής πολυπλοκότητας. Εξ' αιτίας αυτών των χαρακτηριστικών τους, οι παραλλαγές αυτές θα χρησιμοποιηθούν και στην προσομοίωση μας που θα μελετήσει την επίδοση των LDPC κωδίκων.

4.4.1 Μέθοδος Mackay

Όπως φάνηκε από το παράδειγμα εκτέλεσης του βασικού αλγορίθμου στην ενότητα 4.3.3, οι συντελεστές R_{ij}^0 και R_{ij}^1 υπολογίζονται συναρτήσει των συντελεστών Q_{ij}^0 και Q_{ij}^1, λαμβάνοντας υπόψη όλους τους δυνατούς συνδυασμούς κωδικοποιημένων bits που ικανοποιούν την αντίστοιχη εξίσωση ισοτιμίας. Σαφώς κάτι τέτοιο επιβαρύνει σημαντικά την υπολογιστική πολυπλοκότητα, με ολοένα και αυξανόμενο ρυθμό μάλιστα για αυξανόμενο μέγεθος κώδικα.
Σε αυτό το σημείο κρίνεται πολύ σημαντική η εργασία των Mackay και Neal [7] , οι οποίοι εισήγαγαν μία μέθοδο υπολογισμού των εν λόγω συντελεστών χωρίς να είναι απαραίτητο να ληφθούν υπόψη όλοι οι προαναφερθέντες συνδυασμοί. Τη μέθοδο αυτή θα παρουσιάσουμε ευθύς αμέσως.
Σε πρώτη φάση, σημειώνουμε ότι η διαδικασία αρχικοποίησης του τροποποιημένου αλγορίθμου (μέθοδος Mackay) είναι η ίδια με αυτή του αλγορίθμου BP και μπορεί να περιγραφεί συνοπτικά από την παρακάτω σχέση:

$$Q_{ij}^x = f_j^x \tag{4.9}$$

Υπενθυμίζουμε ότι με f_j^x συμβολίζουμε την πιθανότητα το j-οστό bit να είναι x (x=0 ή 1 στην περίπτωση δυαδικών κωδίκων που μας απασχολούν). Αυτή η πιθανότητα προκύπτει από τη ληφθείσα τιμή σήματος που αντιστοιχεί στο j-οστό bit και από τα χαρακτηριστικά του τηλεπικοινωνιακού διαύλου.
Μετά τη φάση αρχικοποίησης, η μέθοδος Mackay διαφοροποιείται εισάγοντας νέες βοηθητικές μεταβλητές και υλοποιώντας δύο βασικά βήματα: το οριζόντιο βήμα (horizontal step) και το κατακόρυφο βήμα (vertical step). Υπολογίζεται λοιπόν σε πρώτη φάση η (νεοεισαχθείσα) ποσότητα δQ_{ij} ως εξής:

$$\delta Q_{ij} = Q_{ij}^0 - Q_{ij}^1 \tag{4.10}$$

Επίσης, υπολογίζεται και η ποσότητα δR_{ij} ως :

$$\delta R_{ij} = \prod_{j' \in N(i) \setminus j} \delta Q_{ij'} \tag{4.11}$$

Οι γνωστοί μας από τα προηγούμενα συντελεστές R_{ij}^x υπολογίζονται με βάση την παρακάτω διαδικασία (horizontal step):

$$R_{ij}^0 = \frac{1}{2}(1 + \delta R_{ij}) \tag{4.12}$$

και

$$R_{ij}^1 = \frac{1}{2}(1 - \delta R_{ij}) \tag{4.13}$$

Σειρά έχουν οι συντελεστές Q_{ij}^x που υπολογίζονται σε κάθε επανάληψη κατά το κατακόρυφο βήμα (vertical step) με βάση την παρακάτω γενική σχέση:

$$Q_{ij}^x = a_{ij} f_j^x \prod_{i' \in M(j) \setminus i} R_{i'j}^x \tag{4.14}$$

Υπενθυμίζουμε ότι όπου αναφερόμαστε σε ζεύγη ij, δηλώνουμε όλα τα συνδεδεμένα ζεύγη κόμβων ελέγχου – κόμβων bit αντίστοιχα (Check nodes- Bit nodes). Επίσης, ο συμβολισμός $i' \in M(j) \setminus i$ που υπάρχει στην (4.14), έχει το ίδιο νόημα με προηγουμένως, δηλαδή ο συμβολισμός M(j) αναπαριστά το σύνολο των δεικτών των κόμβων ελέγχου που είναι συνδεδεμένοι με τον d_j , ενώ ο συμβολισμός M(j) \ i αναπαριστά το ίδιο σύνολο με τον αποκλεισμό όμως του κόμβου ελέγχου i.

Επιστρέφοντας στην (4.14), η σταθερά a_{ij} επιλέγεται έτσι ώστε να ικανοποιείται η κανονικοποιητική σχέση: $Q_{ij}^0 + Q_{ij}^1 = 1$.

Η τελική εκτίμηση για το αποκωδικοποιημένο διάνυσμα κατά το πέρας της τρέχουσας επανάληψης θα γίνει, όπως και στο βασικό αλγόριθμο, υπολογίζοντας την πιθανοτική εκτίμηση για την κατάσταση x κάθε bit ξεχωριστά (x=0 ή 1). Υιοθετώντας και πάλι λοιπόν το συμβολισμό Q_j^x για την παράσταση της εκ των υστέρων πιθανότητας (a posteriori probability) το j-οστό bit να είναι στην κατάσταση x, μπορούμε να γράψουμε την παρακάτω σχέση:

$$Q_j^x = a_j f_j^x \prod_{i \in M(j)} R_{ij}^x \tag{4.15}$$

όπου και πάλι η σταθερά a_j επιλέγεται έτσι ώστε $Q_j^0 + Q_j^1 = 1$

Η τρέχουσα εκτίμηση τώρα για το αποκωδικοποιημένο διάνυσμα **d** λαμβάνεται μέσω της ακόλουθης αλγοριθμικής διαδικασίας: Για κάθε bit j του διανύσματος,

υπολογίζονται οι τιμές Q_j^0 και Q_j^1. Εάν $Q_j^0 > Q_j^1$ τότε το bit j τίθεται ίσο με 0 $(d_j = 0)$, αλλιώς $d_j = 1$.

4.4.1.1 Ισοδυναμία μεθόδου Mackay σε σχέση με τον βασικό sum-product αλγόριθμο

Τονίσαμε ήδη την απλούστευση της υπολογιστικής διαδικασίας που επιφέρει η μέθοδος Mackay σε σύγκριση με τον παραδοσιακό sum-product αλγόριθμο αποκωδικοποίησης. Μένει να δώσουμε στοιχεία για την ισοδυναμία των δύο αλγορίθμων. Προς αυτή την κατεύθυνση, θα 'τρέξουμε' τη μέθοδο Mackay για LDPC κώδικα ίδιο με αυτό του παραδείγματος στην ενότητα 4.3.3.

Όπως είπαμε, οι φάσεις αρχικοποίησης και των δύο μεθόδων ταυτίζονται καθώς και στις δύο περιπτώσεις $Q_{ij}^x = f_j^x$. Στη συνέχεια, κατά τη φάση επαναληπτικής λειτουργίας του αλγορίθμου αποκωδικοποίησης, υπολογίζεται πρώτα η ποσότητα $\delta Q_{ij} = Q_{ij}^0 - Q_{ij}^1$ (4.10) για όλα τα συνδεδεμένα ζεύγη κόμβων ελέγχου ισοτιμίας (i) – κόμβων bit (j). Για να γίνει κάτι τέτοιο απαιτείται η κατασκευή των παρακάτω πινάκων :

$$H_{Q0} = \begin{bmatrix} 0 & Q_{12}^0 & 0 & Q_{14}^0 & 0 & Q_{16}^0 & Q_{17}^0 & Q_{18}^0 & 0 & 0 & 0 & Q_{1,12}^0 \\ Q_{21}^0 & 0 & Q_{23}^0 & Q_{24}^0 & 0 & 0 & 0 & 0 & Q_{29}^0 & 0 & 0 & 0 \\ 0 & Q_{32}^0 & 0 & 0 & Q_{35}^0 & 0 & Q_{37}^0 & 0 & 0 & 0 & 0 & Q_{3,12}^0 \\ Q_{41}^0 & 0 & 0 & Q_{44}^0 & 0 & 0 & 0 & 0 & 0 & Q_{4,10}^0 & Q_{4,11}^0 & 0 \\ 0 & 0 & Q_{53}^0 & 0 & Q_{55}^0 & Q_{56}^0 & 0 & 0 & 0 & Q_{5,10}^0 & 0 & 0 \\ Q_{61}^0 & 0 & Q_{63}^0 & 0 & 0 & 0 & Q_{67}^0 & Q_{68}^0 & 0 & 0 & Q_{6,11}^0 & 0 \\ 0 & Q_{72}^0 & 0 & 0 & 0 & Q_{76}^0 & 0 & Q_{78}^0 & Q_{79}^0 & Q_{7,10}^0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{85}^0 & 0 & 0 & 0 & Q_{89}^0 & 0 & Q_{8,11}^0 & Q_{8,12}^0 \end{bmatrix}$$

$$H_{Q1} = \begin{bmatrix} 0 & Q_{12}^1 & 0 & Q_{14}^1 & 0 & Q_{16}^1 & Q_{17}^1 & Q_{18}^1 & 0 & 0 & 0 & Q_{1,12}^1 \\ Q_{21}^1 & 0 & Q_{23}^1 & Q_{24}^1 & 0 & 0 & 0 & 0 & Q_{29}^1 & 0 & 0 & 0 \\ 0 & Q_{32}^1 & 0 & 0 & Q_{35}^1 & 0 & Q_{37}^1 & 0 & 0 & 0 & 0 & Q_{3,12}^1 \\ Q_{41}^1 & 0 & 0 & Q_{44}^1 & 0 & 0 & 0 & 0 & 0 & Q_{4,10}^1 & Q_{4,11}^1 & 0 \\ 0 & 0 & Q_{53}^1 & 0 & Q_{55}^1 & Q_{56}^1 & 0 & 0 & 0 & Q_{5,10}^1 & 0 & 0 \\ Q_{61}^1 & 0 & Q_{63}^1 & 0 & 0 & 0 & Q_{67}^1 & Q_{68}^1 & 0 & 0 & Q_{6,11}^1 & 0 \\ 0 & Q_{72}^1 & 0 & 0 & 0 & Q_{76}^1 & 0 & Q_{78}^1 & Q_{79}^1 & Q_{7,10}^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{85}^1 & 0 & 0 & 0 & Q_{89}^1 & 0 & Q_{8,11}^1 & Q_{8,12}^1 \end{bmatrix}$$

Στόχος μας είναι να υπολογιστούν οι ποσότητες $\delta Q_{ij} = Q_{ij}^0 - Q_{ij}^1$, οπότε απαιτείται η αφαίρεση των παραπάνω πινάκων. Έστω $H_{\delta Q}$ ο πίνακας που προκύπτει ως :

$$H_{\delta Q} = H_{Q0} - H_{Q1}$$

Τότε,

$$H_{\delta Q}=\begin{bmatrix} 0 & \delta Q_{12} & 0 & \delta Q_{14} & 0 & \delta Q_{16} & \delta Q_{17} & \delta Q_{18} & 0 & 0 & 0 & \delta Q_{1,12} \\ \delta Q_{21} & 0 & \delta Q_{23} & \delta Q_{24} & 0 & 0 & 0 & 0 & \delta Q_{29} & 0 & 0 & 0 \\ 0 & \delta Q_{32} & 0 & 0 & \delta Q_{35} & 0 & \delta Q_{37} & 0 & 0 & 0 & 0 & \delta Q_{3,12} \\ \delta Q_{41} & 0 & 0 & \delta Q_{44} & 0 & 0 & 0 & 0 & 0 & \delta Q_{4,10} & \delta Q_{4,11} & 0 \\ 0 & 0 & \delta Q_{53} & 0 & \delta Q_{55} & \delta Q_{56} & 0 & 0 & 0 & \delta Q_{5,10} & 0 & 0 \\ \delta Q_{61} & 0 & \delta Q_{63} & 0 & 0 & 0 & \delta Q_{67} & \delta Q_{68} & 0 & 0 & \delta Q_{6,11} & 0 \\ 0 & \delta Q_{72} & 0 & 0 & 0 & \delta Q_{76} & 0 & \delta Q_{78} & \delta Q_{79} & \delta Q_{7,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta Q_{85} & 0 & 0 & 0 & \delta Q_{89} & 0 & \delta Q_{8,11} & \delta Q_{8,12} \end{bmatrix}$$

Για να δείξουμε την ισοδυναμία των δύο μεθόδων ας υπολογίσουμε την τιμή του συντελεστή R_{21}^0. Με βάση τη μέθοδο Mackay παίρνουμε:

$$\begin{aligned}
\delta R_{21} &= \delta Q_{23}\,\delta Q_{24}\,\delta Q_{29} \\
&= (Q_{23}^0 - Q_{23}^1)\,(Q_{24}^0 - Q_{24}^1)\,(Q_{29}^0 - Q_{29}^1) \\
&= (Q_{23}^0 Q_{24}^0 - Q_{23}^0 Q_{24}^1 - Q_{23}^1 Q_{24}^0 + Q_{23}^1 Q_{24}^1)\,(Q_{29}^0 - Q_{29}^1) \\
&= Q_{23}^0 Q_{24}^0 Q_{29}^0 - Q_{23}^0 Q_{24}^1 Q_{29}^0 - Q_{23}^1 Q_{24}^0 Q_{29}^0 + Q_{23}^1 Q_{24}^1 Q_{29}^0 - Q_{23}^0 Q_{24}^0 Q_{29}^1 \\
&+ Q_{23}^0 Q_{24}^1 Q_{29}^1 + Q_{23}^1 Q_{24}^0 Q_{29}^1 - Q_{23}^1 Q_{24}^1 Q_{29}^1
\end{aligned}$$

Καθώς ισχύει η γενική σχέση $Q_{ij}^0 + Q_{ij}^1 = 1$, έπεται ότι $Q_{24}^0 = 1 - Q_{24}^0$ και $Q_{29}^0 = 1 - Q_{29}^0$ κάτι που θα μας βοηθήσει στην τελική έκφραση του αποτελέσματος. Παίρνουμε λοιπόν:

$$\begin{aligned}
R_{21}^0 &= (1/2)(1 + \delta R_{21}) \\
&= (1/2)\,(1 + Q_{23}^0 Q_{24}^0 Q_{29}^0 - Q_{23}^0 Q_{24}^1 Q_{29}^0 - Q_{23}^1 Q_{24}^0 Q_{29}^0 + Q_{23}^1 Q_{24}^1 Q_{29}^0 - Q_{23}^0 Q_{24}^0 Q_{29}^1 \\
&\quad + Q_{23}^0 Q_{24}^1 Q_{29}^1 + Q_{23}^1 Q_{24}^0 Q_{29}^1 - Q_{23}^1 Q_{24}^1 Q_{29}^1) \\
&= (1/2)\left[1 + Q_{23}^0 Q_{24}^0 Q_{29}^0 - Q_{23}^0 (1 - Q_{24}^0)\, Q_{29}^0 - Q_{23}^1 (1 - Q_{24}^1)\, Q_{29}^0 + Q_{23}^1 Q_{24}^1 Q_{29}^0\right. \\
&\quad \left. - Q_{23}^0 (1 - Q_{24}^1)\, Q_{29}^1 + Q_{23}^0 Q_{24}^1 Q_{29}^1 + Q_{23}^1 Q_{24}^0 Q_{29}^1 - Q_{23}^1 (1 - Q_{24}^0)\, Q_{29}^1\right] \\
&= (1/2)\left[1 + Q_{23}^0 Q_{24}^0 Q_{29}^0 - Q_{23}^0 Q_{29}^0 + Q_{23}^0 Q_{24}^0 Q_{29}^0 - Q_{23}^1 Q_{29}^0 + Q_{23}^1 Q_{24}^1 Q_{29}^0\right. \\
&\quad + Q_{23}^1 Q_{24}^1 Q_{29}^0 - Q_{23}^0 Q_{29}^1 + Q_{23}^0 Q_{24}^1 Q_{29}^1 + Q_{23}^0 Q_{24}^1 Q_{29}^1 + Q_{23}^1 Q_{24}^0 Q_{29}^1 - Q_{23}^1 Q_{29}^1 \\
&\quad \left. + Q_{23}^1 Q_{24}^0 Q_{29}^1\right] \\
&= (1/2)\left[2Q_{23}^0 Q_{24}^0 Q_{29}^0 + 2Q_{23}^1 Q_{24}^1 Q_{29}^0 + 2Q_{23}^0 Q_{24}^1 Q_{29}^1 + 2Q_{23}^1 Q_{24}^0 Q_{29}^1\right. \\
&\quad \left. + 1 - Q_{23}^0 Q_{29}^0 - Q_{23}^1 Q_{29}^0 - Q_{23}^0 Q_{29}^1 - Q_{23}^1 Q_{29}^1\right] \\
&= (1/2)\left[2Q_{23}^0 Q_{24}^0 Q_{29}^0 + 2Q_{23}^1 Q_{24}^1 Q_{29}^0 + 2Q_{23}^0 Q_{24}^1 Q_{29}^1 + 2Q_{23}^1 Q_{24}^0 Q_{29}^1\right. \\
&\quad \left. + 1 - Q_{23}^0 (Q_{29}^0 + Q_{29}^1) - Q_{23}^1 (Q_{29}^0 + Q_{29}^1)\right] \\
&= Q_{23}^0 Q_{24}^0 Q_{29}^0 + Q_{23}^1 Q_{24}^1 Q_{29}^0 + Q_{23}^0 Q_{24}^1 Q_{29}^1 + Q_{23}^1 Q_{24}^0 Q_{29}^1
\end{aligned}$$

Καταλήξαμε δηλαδή ότι :

$$R_{21}^{0} = Q_{23}^{0}Q_{24}^{0}Q_{29}^{0} + Q_{23}^{1}Q_{24}^{1}Q_{29}^{0} + Q_{23}^{0}Q_{24}^{1}Q_{29}^{1} + Q_{23}^{1}Q_{24}^{0}Q_{29}^{1}$$

Με βάση τον κλασσικό sum-product αλγόριθμο, ισχύουν τα παρακάτω: η τιμή R_{21}^{0} εκφράζει την πιθανοτική εκτίμηση να ικανοποιείται η εξίσωση ισοτιμίας 2, δηλαδή $c_1 \oplus c_3 \oplus c_4 \oplus c_9 = 0$ όταν το bit στη θέση 1 (c_1) του διανύσματος έχει την τιμή 0. Παρατηρούμε ότι η παραπάνω εξίσωση ικανοποιείται για οποιοδήποτε συνδυασμό των bits c_3, c_4, c_9, στον οποίο ο αριθμός των άσσων είναι άρτιος όπως προκύπτει από τους κανόνες της modulo-2 αριθμητικής. Θα υπολογίσουμε λοιπόν την πιθανότητα R_{21}^{0} αθροίζοντας την πιθανότητα να ισχύει κάθε ένας από τους προαναφερθέντες συνδυασμούς. Για κάθε έναν από αυτούς τους συνδυασμούς, π.χ. τα bits c_3, c_4, c_9 να είναι όλα 0, η πιθανότητα του συνδυασμού ισούται με το γινόμενο των πιθανοτήτων κάθε bit να είναι στην κατάσταση που εξετάζουμε λόγω ανεξαρτησίας. Και με τον κλασσικό αλγόριθμο λοιπόν προκύπτει:

$$R_{21}^{0} = Q_{23}^{0}Q_{24}^{0}Q_{29}^{0} + Q_{23}^{1}Q_{24}^{1}Q_{29}^{0} + Q_{23}^{0}Q_{24}^{1}Q_{29}^{1} + Q_{23}^{1}Q_{24}^{0}Q_{29}^{1}$$

Ομοίως, για τον υπολογισμό της τιμής του συντελεστή R_{21}^{1} με βάση τη μέθοδο Mackay παίρνουμε:

$$R_{21}^{1} = \frac{1}{2}(1 - \delta R_{21}) = Q_{23}^{0}Q_{24}^{1}Q_{29}^{0} + Q_{23}^{1}Q_{24}^{0}Q_{29}^{0} + Q_{23}^{0}Q_{24}^{0}Q_{29}^{1} + Q_{23}^{1}Q_{24}^{1}Q_{29}^{1}$$

κάτι που έρχεται σε συμφωνία με τον κλασσικό αλγόριθμο αποκωδικοποίησης (περιττός αριθμός άσσων ανά συνδυασμό).

4.4.2 Λογαριθμικός αποκωδικοποιητής LDPC κωδίκων

Μία ακόμα σημαντική διαφοροποίηση του βασικού αλγορίθμου αποκωδικοποίησης είναι ο λογαριθμικός αποκωδικοποιητής. Μάλιστα, τυγχάνει ιδιαίτερης εκτίμησης σε πρακτικά συστήματα λόγω της μειωμένης πολυπλοκότητας που μπορεί να εγγυηθεί. Η αιτία πίσω από αυτή τη διαπιστούμενη μειωμένη πολυπλοκότητα είναι η λογαριθμική φύση του αποκωδικοποιητή. Με αυτή τη μέθοδο λοιπόν, όπου υπολογίζονται οι λογάριθμοι των συντελεστών που έχουμε ήδη συναντήσει, καταφέρνουμε να μετατρέψουμε τα γινόμενα συντελεστών σε αθροίσματα συντελεστών και τα πηλίκα συντελεστών σε αφαιρέσεις συντελεστών, όπως προκύπτει από τις γνωστές ιδιότητες των λογαρίθμων. Αυτή η μετατροπή οδηγεί σε μείωση της πολυπλοκότητας και αν συνυπολογίσουμε ότι οι πολλαπλασιασμοί εμπλέκουν αριθμητικά ασταθέστερους υπολογισμούς καταλαβαίνουμε γιατί συνήθως χρησιμοποιούνται λογαριθμικοί LDPC αποκωδικοποιητές. Σημειώνουμε ότι για τους ίδιους λόγους προτιμούνται και λογαριθμικοί αποκωδικοποιητές turbo κωδίκων.

Πριν περάσουμε στην παρουσίαση της λειτουργίας του λογαριθμικού αποκωδικοποιητή θα παραθέσουμε κάποιες χρήσιμες μαθηματικές εκφράσεις που θα μας βοηθήσουν στη συνέχεια.

Πρώτα, θα μας βοηθήσει το γεγονός ότι ένας θετικός αριθμός $z \le 1$ μπορεί ισοδύναμα να εκφραστεί ως:

$$z = e^{-|Lz|} \rightarrow |Lz| = |\ln(z)| \tag{4.16}$$

Επιπλέον, έχουμε :

$$\text{Για m>n, } \ln(e^m + e^n) = \ln[e^m(1+e^{n-m})] = \ln e^m + \ln(1+e^{n-m})$$
$$= m + \ln(1+e^{n-m})$$

Ομοίως για $m<n$ παίρνουμε:

$$\ln(e^m + e^n) = \ln[e^n(1+e^{m-n})] = \ln e^n + \ln(1+e^{m-n})$$
$$= n + \ln(1+e^{m-n})$$

Στη γενική περίπτωση λοιπόν, ισχύει:

$$\ln(e^m + e^n) = \max(m,\ n) + \ln(1+e^{-|m-n|}) \tag{4.17}$$

Με τον ίδιο τρόπο προκύπτει:

$$\ln(e^m - e^n) = \max(m,\ n) + \ln(1-e^{-|m-n|}),\ \text{m>n} \tag{4.18}$$

Είμαστε τώρα σε θέση να αναλύσουμε τον λογαριθμικό αποκωδικοποιητή LDPC κωδίκων, ο οποίος βασικά είναι το λογαριθμικό ισοδύναμο της μεθόδου Mackay όπως αυτή παρουσιάστηκε στην ενότητα 4.4.1. Έτσι, τα βήματα εκτέλεσης του αλγορίθμου είναι το βήμα αρχικοποίησης και το βήμα επαναληπτικής λειτουργίας το οποίο αναλύεται στο οριζόντιο βήμα (horizontal step) και στο κατακόρυφο βήμα (vertical step), όπως θα φανεί και παρακάτω.

4.4.2.1 Αρχικοποίηση (Initialization)

Στην αρχικοποίηση του αλγορίθμου αποκωδικοποίησης έχουμε συναντήσει την παρακάτω βασική σχέση:

$$Q_{ij}^x = f_j^x \tag{4.9}$$

όπου με f_j^x συμβολίζουμε την πιθανότητα το bit στη θέση j του διανύσματος προς αποκωδικοποίηση να έχει την τιμή x. Μιας και η f_j^x είναι πιθανότητα, σαφώς πρόκειται για αριθμό μικρότερο ή ίσο της μονάδας. Σε άμεση αναλογία με τη σχέση (4.16) λοιπόν, προκύπτει:

$$f_j^x = e^{-|Lf_j^x|} \rightarrow \left|Lf_j^x\right| = \left|\ln\left(f_j^x\right)\right| \tag{4.19}$$

καθώς και:

$$Q_{ij}^x = e^{-|LQ_{ij}^x|} \rightarrow \left|LQ_{ij}^x\right| = \left|\ln\left(Q_{ij}^x\right)\right| \tag{4.20}$$

Από εδώ και στο εξής, όπου συναντούμε το σύμβολο L μπροστά από συντελεστή θα καταλαβαίνουμε ότι πρόκειται για λογαριθμικό συντελεστή. Π.χ. στη σχέση (4.20) οι συντελεστές LQ_{ij}^x 'αντιστοιχούν' στη λογαριθμική μορφή των γνωστών μας συντελεστών Q_{ij}^x όπως τους γνωρίσαμε στους μη- λογαριθμικούς αποκωδικοποιητές.

4.4.2.2 Οριζόντιο Βήμα (Horizontal Step)

Η πρώτη ενέργεια του οριζόντιου βήματος της μεθόδου Mackay απεικονίζεται στη σχέση (4.10). Αν γράψουμε αυτή την εξίσωση χρησιμοποιώντας λογαριθμικούς συντελεστές παίρνουμε:

$$e^{-|L\delta Q_{ij}|} = e^{-|LQ_{ij}^0|} - e^{-|LQ_{ij}^1|} \tag{4.21}$$

Την παραπάνω σχέση μπορούμε να την γράψουμε και ως εξής:

$$\delta Q_{ij} = (-1)^{s_{ij}} \left|e^{-|L\delta Q_{ij}|}\right| = (-1)^{s_{ij}} \left|e^{-|LQ_{ij}^0|} - e^{-|LQ_{ij}^1|}\right| \tag{4.22}$$

όπου εάν $\left|LQ_{ij}^0\right| \le \left|LQ_{ij}^0\right|$ τότε $s_{ij} = 0$, αλλιώς $s_{ij} = 1$.

Η σχέση (4.21) τώρα μπορεί να γραφτεί ως:

$$\left|L\delta Q_{ij}\right| = \left|\ln\left(\left|e^{-|LQ_{ij}^0|} - e^{-|LQ_{ij}^1|}\right|\right)\right|$$

Από την έκφραση 4.18 προκύπτει:

$$\left|\ln\left(\left|e^{-|m|} - e^{-|n|}\right|\right)\right| = \min\left(|m|,|n|\right) + \left|\ln\left(1 - e^{-\|m|-|n\|}\right)\right|$$

Από τις δύο τελευταίες σχέσεις παίρνουμε:

$$\left|L\delta Q_{ij}\right| = \min(\left|LQ_{ij}^{0}\right|, \left|LQ_{ij}^{1}\right|) + \left|\ln(1 - e^{-\left\|LQ_{ij}^{0}\right| - \left|LQ_{ij}^{1}\right\|})\right| \quad (4.23)$$

ή

$$\left|L\delta Q_{ij}\right| = \min(\left|LQ_{ij}^{0}\right|, \left|LQ_{ij}^{1}\right|) + f_{-}(\left\|LQ_{ij}^{0}\right| - \left|LQ_{ij}^{1}\right\|) \quad (4.24)$$

όπου f_{-} είναι look-up table με εισόδους στην παραπάνω σχέση $\left|LQ_{ij}^{0}\right|$ και $\left|LQ_{ij}^{1}\right|$. Τέτοιου είδους πίνακες χρησιμοποιούνται σε πρακτικά συστήματα όταν χρειάζεται να υπολογίσουμε πολλές φορές την τιμή μιας συνάρτησης. Αντί λοιπόν να υπολογιστεί αναλυτικά, το υπολογιστικό σύστημα ανατρέχει στον look-up table και βρίσκει την τιμή που αντιστοιχεί στην κατάλληλη εγγραφή του πίνακα. Επειδή βέβαια οι εγγραφές του πίνακα είναι κβαντισμένες υπεισέρχεται γενικά κάποιο σφάλμα στη διαδικασία υπολογισμού. Με κατάλληλη κατασκευή του πίνακα όμως το σφάλμα αυτό μπορεί να είναι αμελητέο και έτσι να μπορούμε να εκμεταλλευτούμε την αυξημένη ταχύτητα επεξεργασίας που μπορούν να προσδώσουν στο σύστημα πίνακες τέτοιου τύπου.
Επιστρέφοντας τώρα στο λογαριθμικό αποκωδικοποιητή, μπορούμε να γράψουμε την εξίσωση (4.11) ως:

$$\delta R_{ij} = e^{-\left| L\delta R_{ij} \right|} = \prod_{j' \in N(i) \setminus j} (-1)^{s_{ij'}} \left| e^{-\left| L\delta Q ij' \right|} \right| = (-1)^{\sum s_{ij'}} \prod_{j' \in N(i) \setminus j} \left| e^{-\left| L\delta Q ij' \right|} \right| \quad (4.25)$$

από την οποία παίρνουμε :

$$\left|L\delta R_{ij}\right| = \sum_{j' \in N(i) \setminus j} \left|L\delta Q_{ij'}\right| \quad (4.26)$$

και

$$s\delta R_{ij} = \sum_{j' \in N(i) \setminus j} s_{ij'} \quad (4.27)$$

δεχόμενοι κατ' αναλογία της (4.22) ότι:

$$\delta R_{ij} = (-1)^{s\delta R_{ij}} \left| e^{-\left| L\delta R_{ij} \right|} \right| \quad (4.28)$$

Σε αυτό το σημείο να σημειώσουμε την ιδιαίτερη βαρύτητα της σχέσης (4.26), μέσω της οποίας φαίνεται ότι ο υπολογισμός των λογαριθμικών συντελεστών απαιτεί τον υπολογισμό αθροισμάτων και όχι γινομένων όπως συνέβαινε στην περίπτωση μη-λογαριθμικού αποκωδικοποιητή. Αυτή η τελευταία διαπίστωση είναι η βασική αιτία υπεροχής του λογαριθμικού αποκωδικοποιητή σε επίπεδο υπολογιστικής πολυπλοκότητας.

Οι συντελεστές R_{ij}^0, R_{ij}^1 υπολογίζονται μέσω των τιμών των συντελεστών δR_{ij} από τις σχέσεις (4.12) και (4.13) αντίστοιχα. Το λογαριθμικό ισοδύναμο των ανωτέρω σχέσεων έχει ως εξής

$$\ln(R_{ij}^0) = -\left|LR_{ij}^0\right| = \ln(1+(-1)^{s\delta R_{ij}}\left|e^{-\left|L\delta R_{ij}\right|}\right|) - \ln(2) \tag{4.29}$$

Από την παραπάνω σχέση, αν η ποσότητα $s\delta R_{ij}$ είναι άρτιος αριθμός προκύπτει

$$\left|LR_{ij}^0\right| = \ln(2) - \left|\ln(1+\left|e^{-\left|L\delta R_{ij}\right|}\right|)\right| = \ln(2) - f_+(\left|L\delta R_{ij}\right|) \tag{4.30}$$

ενώ αν η ποσότητα $s\delta R_{ij}$ είναι περιττός αριθμός προκύπτει

$$\left|LR_{ij}^0\right| = \ln(2) + \left|\ln(1-\left|e^{-\left|L\delta R_{ij}\right|}\right|)\right| = \ln(2) + f_-(\left|L\delta R_{ij}\right|) \tag{4.31}$$

Όπως και πριν, ο f_- αποτελεί look-up table (τον ίδιο με προηγουμένως) κάτι που ισχύει και για το νεοεισαχθέντα f_+. Και στις δύο παραπάνω περιπτώσεις οι πίνακες θα επιστρέψουν την τιμή που αντιστοιχεί στην εγγραφή $\left|L\delta R_{ij}\right|$. Εάν δεν υπάρχει εγγραφή με τιμή ακριβώς $\left|L\delta R_{ij}\right|$, λόγω της κβαντισμένης φύσης των εγγραφών, θα αναζητηθεί η εγγραφή που πλησιάζει περισσότερο στην εν λόγω τιμή. Όμοια τώρα με τους συντελεστές $\left|LR_{ij}^0\right|$, προκύπτει για τους $\left|LR_{ij}^1\right|$:

Για $s\delta R_{ij}$ άρτιο,

$$\left|LR_{ij}^1\right| = \ln(2) + \left|\ln(1-\left|e^{-\left|L\delta R_{ij}\right|}\right|)\right| = \ln(2) + f_-(\left|L\delta R_{ij}\right|) \tag{4.32}$$

και για $s\delta R_{ij}$ περιττό,

$$\left|LR_{ij}^1\right| = \ln(2) - \left|\ln(1+\left|e^{-\left|L\delta R_{ij}\right|}\right|)\right| = \ln(2) - f_+(\left|L\delta R_{ij}\right|) \tag{4.33}$$

4.4.2.3 Κατακόρυφο Βήμα (Vertical Step)

Προχωρούμε στο κατακόρυφο βήμα της επαναληπτικής διαδικασίας, σε αντιστοιχία πάντα με τη μέθοδο Mackay. Με στόχο να επιλυθεί η εξίσωση (4.14), είναι χρήσιμο να ορίσουμε την παρακάτω ποσότητα για x=0, 1:

$$c_{ij}^{x} = f_j^{x} \prod_{i' \in M(j) \setminus i} R_{i'j}^{x} \tag{4.34}$$

η οποία σε λογαριθμική μορφή είναι:

$$\ln(c_{ij}^{x}) = \ln(e^{-|Lc_{ij}^{x}|}) = \ln(e^{-|Lf_j^{x}|}) + \sum_{i' \in M(j) \setminus i} \ln(e^{-|LR_{i'j}^{x}|}) \tag{4.35}$$

Από τους παραπάνω ορισμούς και από τις σχέσεις της μεθόδου Mackay προκύπτει:

$$a_{ij} = \frac{1}{c_{ij}^{0} + c_{ij}^{1}} \tag{4.36}$$

οπότε,

$$Q_{ij}^{0} = e^{-|LQ_{ij}^{0}|} = \frac{e^{-|Lc_{ij}^{0}|}}{e^{-|Lc_{ij}^{0}|} + e^{-|Lc_{ij}^{1}|}} \tag{4.37}$$

Από τη σχέση (4.37) παίρνουμε με κατάλληλη επεξεργασία:

$$\left|LQ_{ij}^{0}\right| = \left|Lc_{ij}^{0}\right| - \min\left(\left|Lc_{ij}^{0}\right|, \left|Lc_{ij}^{1}\right|\right) + f_{+}\left(\left\|Lc_{ij}^{0}\right| - \left|Lc_{ij}^{1}\right\|\right) \tag{4.38}$$

Ομοίως,

$$\left|LQ_{ij}^{1}\right| = \left|Lc_{ij}^{1}\right| - \min\left(\left|Lc_{ij}^{0}\right|, \left|Lc_{ij}^{1}\right|\right) + f_{+}\left(\left\|Lc_{ij}^{0}\right| - \left|Lc_{ij}^{1}\right\|\right) \tag{4.39}$$

Συνεχίζοντας, από τη στιγμή που ισχύει:

$$Q_j^{x} = a_j f_j^{x} \prod_{i \in M(j)} R_{ij}^{x} = a_j f_j^{x} R_{ij}^{x} \prod_{i' \in M(j) \setminus i} R_{i'j}^{x}$$

μπορούμε να ορίσουμε την παρακάτω ποσότητα προς διευκόλυνση των υπολογισμών:

$$c_j^{x} = e^{-|Lc_j^{x}|} = f_j^{x} \prod_{i \in M(j)} R_{ij}^{x} \tag{4.38}$$

τέτοια ώστε

$$\left|Lc_j^{x}\right| = \left|Lc_{ij}^{x}\right| + \left|LR_{ij}^{x}\right| \tag{4.39}$$

Όσον αφορά την εκ των υστέρων πιθανότητα (a posteriori probability) Q_j^{x} το j-οστό bit να είναι στην κατάσταση x, μπορούμε να γράψουμε την παρακάτω σχέση η οποία προκύπτει ομοίως με την (4.15):

$$Q_j^0 = e^{-\left|LQ_j^0\right|} = \frac{e^{-\left|Lc_j^0\right|}}{e^{-\left|Lc_j^0\right|} + e^{-\left|Lc_j^1\right|}} \tag{4.42}$$

η οποία λογαριθμικά παίρνει την παρακάτω μορφή:

$$\left|LQ_j^0\right| = \left|Lc_j^0\right| - \min(\left|Lc_j^0\right|, \left|Lc_j^1\right|) + f_+(\left\|Lc_j^0\right| - \left|Lc_j^1\right\|) \tag{4.40}$$

Με εντελώς όμοιο τρόπο παίρνουμε για την πιθανότητα το bit στη θέση j να είναι 1:

$$\left|LQ_j^1\right| = \left|Lc_j^1\right| - \min(\left|Lc_j^0\right|, \left|Lc_j^1\right|) + f_+(\left\|Lc_j^0\right| - \left|Lc_j^1\right\|) \tag{4.41}$$

Είμαστε σε φάση τώρα να πάρουμε την εκτίμηση για το αποκωδικοποιημένο διάνυσμα κατά την τρέχουσα επανάληψη. Κατά τη μη - λογαριθμική αποκωδικοποίηση γνωρίζουμε ότι εάν $Q_j^0 > Q_j^1$ τότε το bit j τίθεται ίσο με 0 $(d_j = 0)$, αλλιώς $d_j = 1$. Αφού λογαριθμικά ισχύει $Q_j^0 = e^{-\left|LQ_j^0\right|}$ και $Q_j^1 = e^{-\left|LQ_j^1\right|}$ ακολουθούμε την εξής διαδικασία που ολοκληρώνει τη βέλτιστη εκτίμηση για κάθε bit ξεχωριστά κατά το πέρας της τρέχουσας επανάληψης:
εάν $\left|LQ_j^0\right| < \left|LQ_j^1\right|$ τότε το bit j τίθεται ίσο με 0 $(d_j = 0)$, αλλιώς $d_j = 1$.

5 *Αποτελέσματα της Προσομοίωσης μας*

Σε αυτό το κεφάλαιο θα παρουσιαστούν μέσω γραφικών παραστάσεων τα αποτελέσματα της προσομοίωσης που στοχεύει στην εκτίμηση της επίδοσης LDPC κωδίκων. Η μελέτη θα γίνει πάνω στα χαρακτηριστικά των κωδίκων αλλά και πάνω στο πως επηρεάζεται η επίδοση τους υπό διάφορα σχήματα διαμόρφωσης και υπό διάφορους τύπους διαύλου. Όλα τα αποτελέσματα προέρχονται από το περιβάλλον Matlab όπου και γράφτηκε ο εκτελέσιμος κώδικας της προσομοίωσης.

Πρώτο βήμα, αλλά πολύ ουσιώδες, για την εργασία μας ήταν η κατασκευή του πίνακα ελέγχου ισοτιμίας **H**. Ο **H** κατασκευάστηκε με βάση τις απαιτήσεις που τέθηκαν στο κεφάλαιο 4. Πιο συγκεκριμένα, το βασικό στοιχείο που τον χαρακτηρίζει είναι η αραιότητα του κάτι που είναι γενικότερα αλληλένδετο με τη φύση των LDPC κωδίκων. Παράλληλα, καταβλήθηκε σημαντική προσπάθεια ώστε στον αντίστοιχο γράφο Tanner να μην υπάρχουν κύκλοι μικρού μήκους σαν συνέχεια της ανάλυσης που έγινε στην ενότητα 4.3.1. Οι διαστάσεις του **H** είναι 128 x 256, κατά συνέπεια ο κώδικας μας είναι **LDPC (256, 128)**. Ο πίνακας είναι κανονικός έχοντας 6 άσσους ανά γραμμή και 3 άσσους ανά στήλη. Όλα τα αποτελέσματα που θα ακολουθήσουν αφορούν τον ίδιο πίνακα. Ένα άλλο σταθερό στοιχείο στοιχείο της προσομοίωσης είναι ο μέγιστος αριθμός επαναλήψεων του επαναληπτικού αλγορίθμου αποκωδικοποίησης. Αν δεν αναφερθεί κάτι άλλο ο μέγιστος αριθμός επαναλήψεων είναι κάθε φορά 80 (**max_iter=80**).

Στη συνέχεια, δημιουργήθηκαν δύο βασικά αρχεία: το ένα αφορούσε την υλοποίηση της μεθόδου Mackay και το άλλο τη λογαριθμική εκδοχή της. Από αυτό το σημείο και μετά εκτελέστηκαν διάφορα σενάρια που θα παρουσιαστούν στη συνέχεια.

5.1 Μελέτη πάνω στα χαρακτηριστικά του LDPC κώδικα

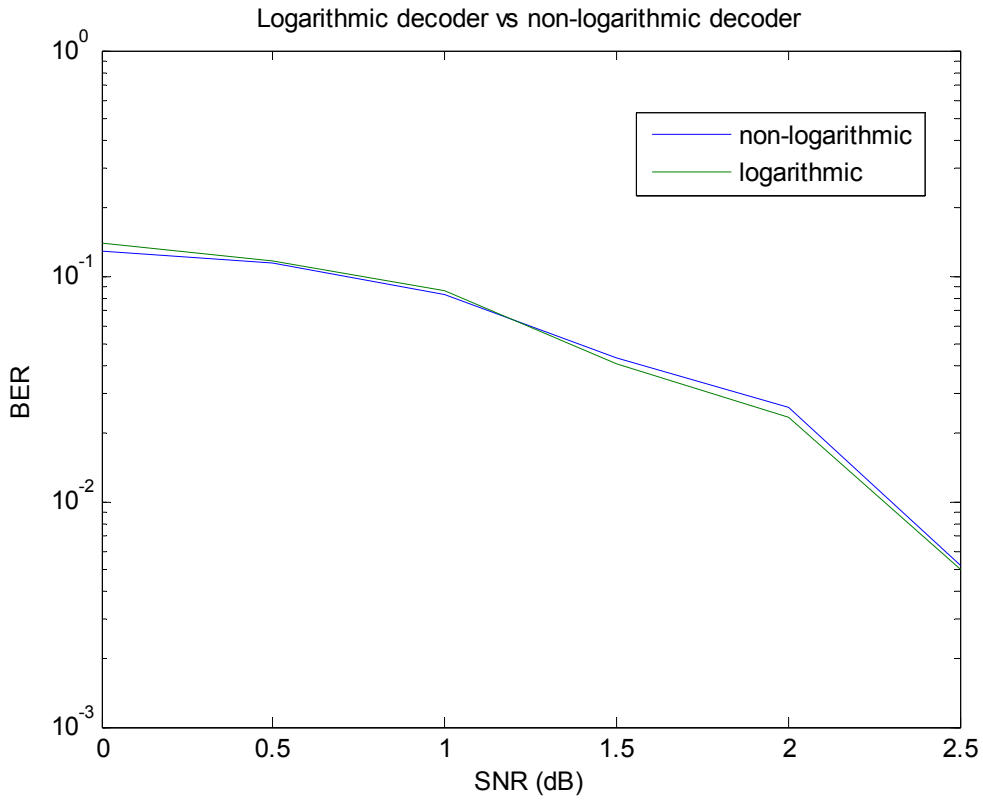
Σε πρώτη φάση εστιάσαμε στη μελέτη των χαρακτηριστικών του κώδικα και έτσι αφήσαμε σταθερά το σχήμα της διαμόρφωσης και τον τύπο διαύλου. Σε όλη αυτή την ενότητα λοιπόν παραθέτουμε αποτελέσματα για

- Σχήμα διαμόρφωσης: **BPSK**
- Τύπος διαύλου: **AWGN**

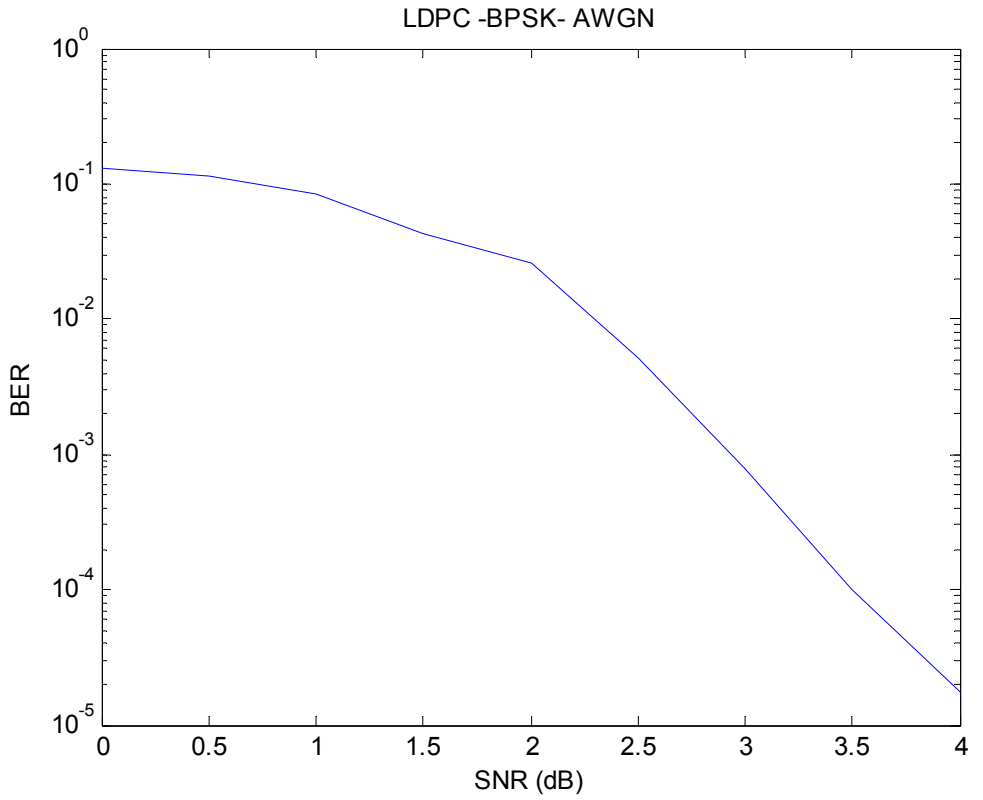
5.1.1 Σύγκριση λογαριθμικού με μη λογαριθμικό αποκωδικοποιητή

Το πρώτο σενάριο που εκτελέσαμε αφορούσε την επίδοση λογαριθμικού και μη αποκωδικοποιητή για εύρος τιμών σηματοθορυβικού λόγου (**SNR**) από 0 έως 2.5 dB με βήμα 0.5 dB. Η επίδοση μετριέται σε όρους ρυθμού λαθών ανά ψηφίο (**BER**) κάτι που θα γίνεται κατ' εξακολούθηση. Τα αποτελέσματα αυτής της προσομοίωσης δίνονται στο σχήμα **5.1**

Όπως φαίνεται, οι δύο καμπύλες διακρίνονται μετά δυσκολίας. Το αποτέλεσμα είναι ικανοποιητικό καθώς καταδεικνύει την ισοδυναμία του λογαριθμικού με το μη-λογαριθμικό αποκωδικοποιητή. Αυτή η ισοδυναμία είχε αναλυθεί και θεωρητικά όταν παρουσιάσαμε το λογαριθμικό αποκωδικοποιητή ως το λογαριθμικό ισοδύναμο της μεθόδου Mackay πάνω στην οποία βασίζεται ο μη-λογαριθμικός αποκωδικοποιητής.



Σχήμα 5.1 Σύγκριση λογαριθμικού με μη λογαριθμικό αποκωδικοποιητή



Σχήμα 5.2 Επίδοση LDPC (256, 128) υπό δίαυλο AWGN και διαμόρφωση BPSK

Αυτή η ελάχιστη απόκλιση των δύο καμπύλων σαφώς και δικαιολογείται στα πλαίσια μιας προσομοίωσης. Από τη στιγμή που έχουμε αποδείξει την ισοδυναμία των δύο αποκωδικοποιητών σε όρους επίδοσης, στο εξής θα αναφερόμαστε απλά σε LDPC αποκωδικοποίηση. Υπενθυμίζουμε ότι, όπως εξηγήσαμε στο προηγούμενο κεφάλαιο, σε πρακτικά συστήματα χρησιμοποιείται κυρίως λογαριθμικός αποκωδικοποιητής λόγω του ταχύτερου χρόνου εκτέλεσης . Ο χρόνος εκτέλεσης όμως είναι κάτι που δεν θα μας απασχολήσει περαιτέρω.

5.1.2 Επίδοση του κώδικα για εύρος SNR από 0-4 dB

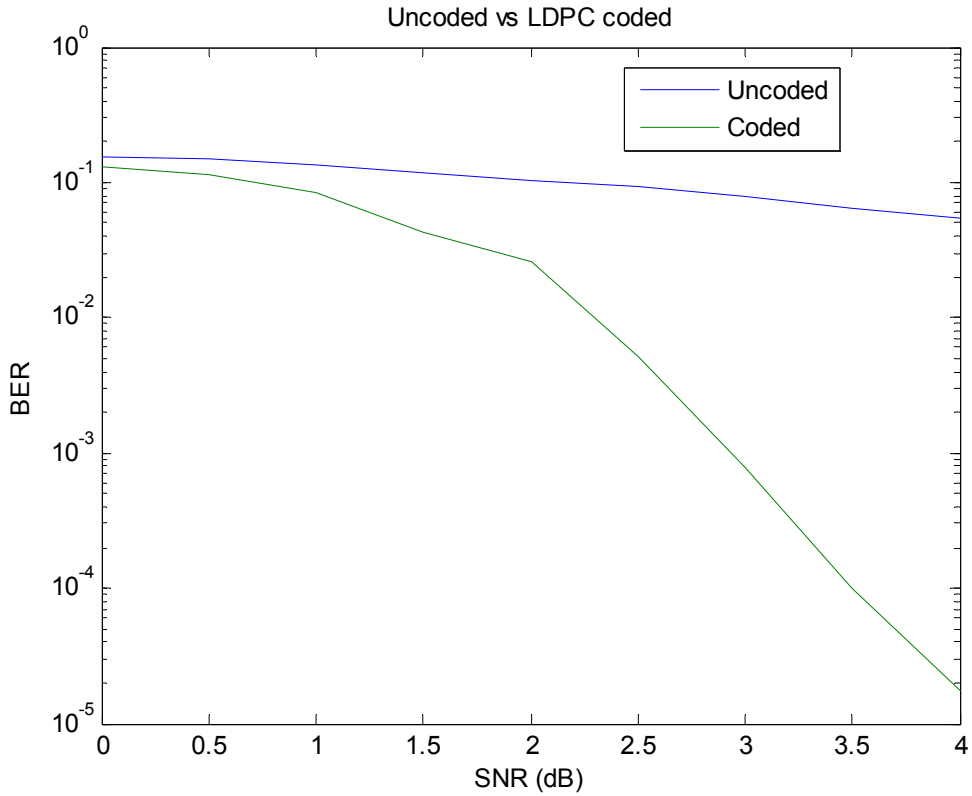
Για να εκτιμηθεί ακόμα καλύτερα η επίδοση του κώδικα αυξάνουμε το εύρος τιμών του σηματοθορυβικού λόγου μέχρι τα 4 dB. Η αντίστοιχη γραφική παράσταση δίνεται στο σχήμα **5.2**. Αναλογιζόμενοι το μέγεθος του κώδικα πρόκειται για αρκετά καλή επίδοση. Σαφώς, έχουν κατασκευαστεί πολύ ισχυρότεροι LDPC κώδικες που όμως έχουν και πολύ μεγάλα μεγέθη κώδικα. Χαρακτηριστικά αναφέρουμε ότι το μήκος της κωδικής λέξης του LDPC κώδικα που έχει υιοθετηθεί στο πρότυπο DVB-S2 είναι 64800 bits. Το τίμημα για τέτοια μεγέθη κώδικα είναι αυξημένες απαιτήσεις αποθηκευτικού χώρου, καθώς μιλάμε για πίνακες ελέγχου ισοτιμίας μεγάλων διαστάσεων, και αύξηση της πολυπλοκότητας τόσο της κωδικοποίησης όσο και της αποκωδικοποίησης.

Επανερχόμενοι στο σχολιασμό του σχήματος **5.2**, βλέπουμε ότι η καμπύλη επίδοσης ακολουθεί την "κλασσική" μορφή καμπυλών επίδοσης τεχνικών κωδικοποίησης. Βεβαίως, όπως αναμενόταν με την αύξηση του σηματοθορυβικού λόγου παρατηρούμε και μείωση του ρυθμού λαθών. Σαν αριθμητικό παράδειγμα, βλέπουμε ότι για SNR 3.5 dB ο ρυθμός λαθών ανά ψηφίο είναι περίπου 10^{-4}. Με άλλα λόγια, για αυτή την τιμή σηματοθορυβικού λόγου εμφανίστηκε ένα λάθος ανά 10000 ψηφία (bits) ή, αλλιώς, η εκτίμηση της προσομοίωσης μας για την πιθανότητα λάθους είναι 10^{-4}. Τέλος, να σημειώσουμε ότι όπως σε όλες τις σχετικές καμπύλες της βιβλιογραφίας, ο κατακόρυφος άξονας (BER) δίνεται σε λογαριθμική κλίμακα.

5.1.3 Κέρδος από τη χρήση Κωδικοποίησης

Επόμενο σενάριο προς προσομοίωση είναι η μελέτη και σύγκριση συστήματος που κάνει χρήση της κωδικοποίησης μας με σύστημα που δεν χρησιμοποιεί κωδικοποίηση (uncoded). Για την εγκυρότητα της σύγκρισης, και τα δύο συστήματα πρέπει να χρησιμοποιούν την ίδια τεχνική διαμόρφωσης και ο τηλεπικοινωνιακός δίαυλος να είναι ίδιου τύπου. Όπως προαναφέρθηκε, προς το παρόν εξετάζουμε μόνο **BPSK** διαμόρφωση και δίαυλο **AWGN**.

Το σχήμα **5.3** οπτικοποιεί αυτή τη σύγκριση και μας δίνει τη δυνατότητα για εξαγωγή χρήσιμων συμπερασμάτων. Βασικά, σημειώνουμε το ολοένα και αυξανόμενο κέρδος από τη χρήση κωδικοποίησης με την αύξηση του σηματοθορυβικού λόγου. Για μικρές τιμές SNR η παρουσία του θορύβου σε σχέση με το σήμα μας είναι ιδιαίτερα έντονη. Κατά συνέπεια, η μεταδιδόμενη κωδική λέξη αλλοιώνεται σημαντικά δυσχεραίνοντας κατά πολύ το έργο του αποκωδικοποιητή για διόρθωση λαθών και έτσι μέχρι SNR ίσο με 1 dB δεν υπάρχει σημαντική βελτίωση της επίδοσης. Μάλιστα, από τη γενική και θεωρητική καμπύλη του σχήματος 2.11



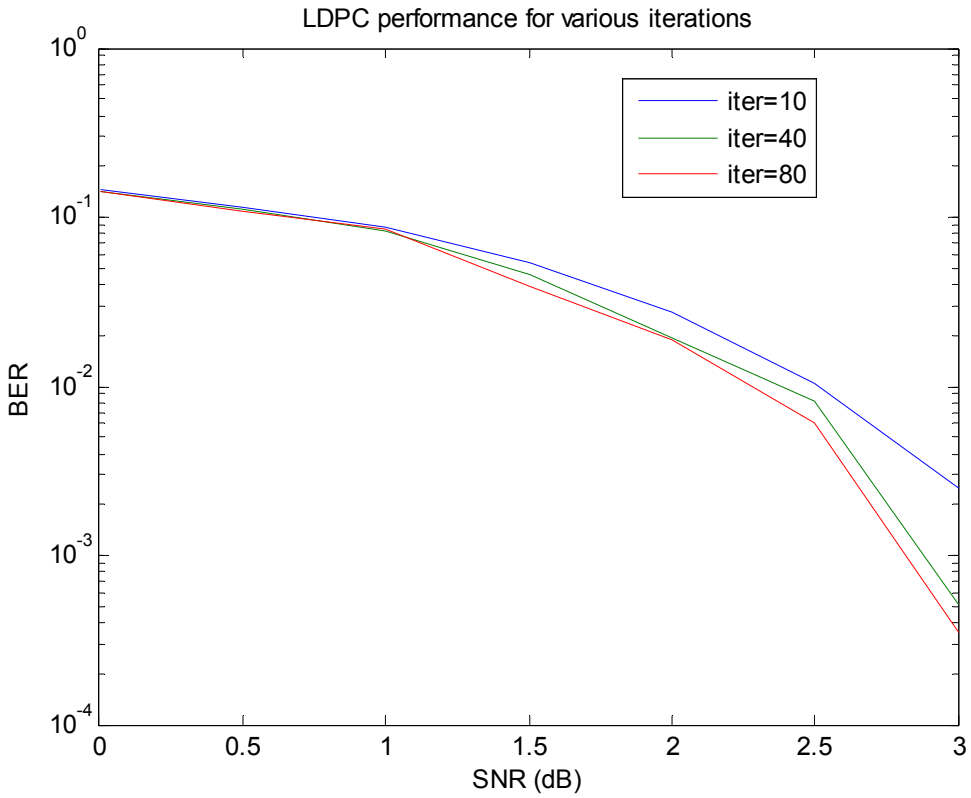
Σχήμα 5.3 Συστήματα με και χωρίς χρήση κωδικοποίησης

προβλέπεται ακόμα και χειροτέρευση της επίδοσης με τη χρήση κωδικοποίησης για ακόμα μικρότερες τιμές του σηματοθορυβικού λόγου. Κάτι τέτοιο μπορεί να εξηγηθεί αν αναλογιστούμε ότι για SNR μικρότερο από 0 dB η αλλοίωση της κωδικής λέξης πιθανότατα θα είναι τόσο μεγάλη ώστε να "μετατίθεται" από την περιοχή της αρχικής κωδικής λέξης σε περιοχές άλλων κωδικών λέξεων. Σε τέτοια περίπτωση, ο αποκωδικοποιητής θα κάνει ακόμα χειρότερα τα πράγματα. Από την άλλη βλέπουμε ότι για ικανοποιητικές τιμές SNR η κωδική λέξη καταφθάνει αλλοιωμένη αλλά όχι "εκφυλισμένη" δίνοντας τη δυνατότητα στον αποκωδικοποιητή να κάνει διόρθωση λαθών. Έτσι, παρατηρούμε κέρδος από τη χρήση κωδικοποίησης που με την αύξηση του σηματοθορυβικού λόγου γίνεται τεράστιο. Για παράδειγμα, η τιμή του BER για 4 dB που προέκυψε από την προσομοίωση είναι 0.0552 χωρίς χρήση κωδικοποίησης και 0.000017188 με χρήση κωδικοποίησης. Δηλαδή, με χρήση κωδικοποίησης καταφέραμε να έχουμε ρυθμό λαθών περίπου 3200 φορές μικρότερο από ότι χωρίς χρήση κωδικοποίησης. Παρ' ότι δεν τρέξαμε ανάλογη προσομοίωση, μπορούμε να προβλέψουμε από τη μορφή των καμπυλών ότι για SNR μεγαλύτερο από 4 dB το κέρδος από τη χρήση κωδικοποίησης θα γίνει ακόμα μεγαλύτερο. Όσον αφορά τώρα το κόστος για αυτή τη σημαντική μείωση του ρυθμού λαθών άρα και την αύξηση της ποιότητας του συστήματος μέσω κωδικοποίησης, ισχύει το γενικό ισοζύγιο που παρουσιάστηκε στην ενότητα **2.6.5**.

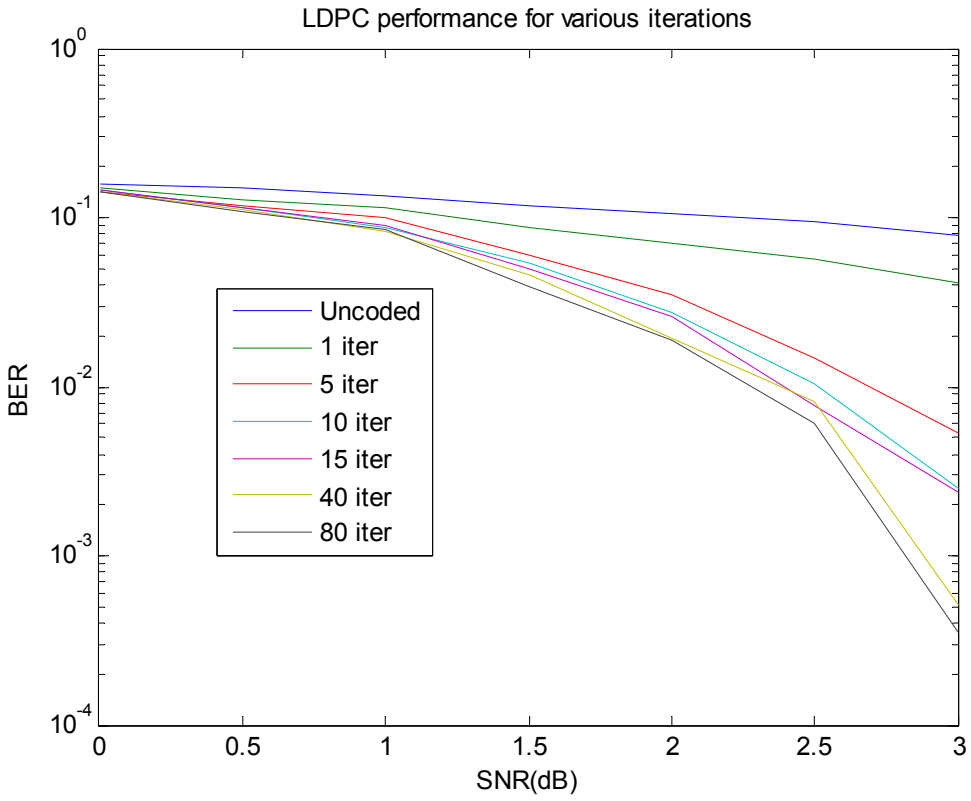
5.1.4 Μελέτη επίδοσης του κώδικα για διάφορες τιμές του μέγιστου αριθμού επαναλήψεων του επαναληπτικού αλγορίθμου αποκωδικοποίησης

Συνεχίζοντας τη σκιαγράφηση του LDPC(256, 128) κώδικα επιθυμούμε να συγκρίνουμε την επίδοση του για διάφορες τιμές του μέγιστου αριθμού επαναλήψεων του επαναληπτικού αλγορίθμου αποκωδικοποίησης. Βεβαίως, για να γίνει κάτι τέτοιο έπρεπε να τροποποιηθεί ανάλογα ο εκτελέσιμος κώδικας των προηγούμενων προσομοιώσεων. Έτσι, ενώ μέχρι τώρα είχαμε μια δομή δεδομένων στην οποία αποθηκεύαμε τιμές BER ανά SNR, για αυτό το σενάριο χρειάστηκε η κατασκευή νέας δομής δεδομένων όπου αποθηκεύουμε τιμές BER ανά SNR και ανά επανάληψη (iteration). Έχοντας στο τέλος της προσομοίωσης αυτή την τρισδιάστατη δομή, μπορούμε μέσω του Matlab να παράξουμε γραφικές παραστάσεις επίδοσης του κώδικα για διάφορες τιμές επαναλήψεων του αλγορίθμου αποκωδικοποίησης.

Το πρώτο αποτέλεσμα αυτού του σεναρίου φαίνεται στο σχήμα **5.4** όπου φαίνεται η επίδοση του LDPC κώδικα για 10, 40 και 80 επαναλήψεις. Όπως αναμενόταν, μεγαλύτερος αριθμός επαναλήψεων οδηγεί σε καλύτερη επίδοση. Όμως, πρέπει να επισημάνουμε ότι πρώτον για χαμηλές τιμές SNR δεν υπάρχει ουσιαστική διαφοροποίηση και δεύτερον ότι οι 40 και οι 80 επαναλήψεις δεν διαφέρουν πολύ ως προς την επίδοση. Το πρώτο συμπέρασμα αιτιολογείται, όπως και προηγουμένως, από τη μεγάλη αλλοίωση που προκαλεί στην κωδική λέξη η μικρή τιμή σηματοθορυβικού λόγου. Έτσι, αν η αλλοίωση είναι μεγάλη, ξεφεύγουμε από τις διορθωτικές ικανότητες του κώδικα και οι περισσότερες επαναλήψεις δεν μπορούν να προσφέρουν ουσιαστικό έργο κάτι που δεν ισχύει για ικανοποιητικές τιμές σηματοθορυβικού λόγου. Όσον αφορά το δεύτερο συμπέρασμα, ενισχύεται και από το σχήμα **5.5** όπου παρουσιάζουμε τη μελέτη επίδοσης για 1, 5, 10, 15, 40 και 80 επαναλήψεις καθώς και την επίδοση χωρίς χρήση κωδικοποίησης. Όπως φαίνεται, εξετάζοντας πάντα ικανοποιητικές τιμές SNR, ενώ η μετάβαση από 1 σε 5 επαναλήψεις οδηγεί σε μεγάλη βελτίωση αυτός ο "ρυθμός βελτίωσης" μειώνεται όσο αυξάνεται ο αριθμός επαναλήψεων. Εμφανίζεται δηλαδή ένα είδος κορεσμού του κώδικα, όπου πάνω από έναν αριθμό επαναλήψεων δεν επιτυγχάνουμε σημαντική βελτίωση. Από αυτό το σχήμα αιτιολογείται και η επιλογή μας για αριθμό επαναλήψεων ίσο με 80, όπου φαίνεται να επέρχεται και ο προαναφερθείς κορεσμός. Σε αυτό τον κώδικα δηλαδή, ακόμα και 1000 επαναλήψεις να κάναμε δεν θα παρατηρούσαμε σημαντική βελτίωση σε σχέση με τις 80 επαναλήψεις. Είναι προφανές ότι ο αριθμός 80 επαναλήψεων που θέσαμε εμείς για τον κώδικα μας μετά από τα πειραματικά αποτελέσματα δεν αποτελεί γενική συνταγή για τους LDPC κώδικες αφού προβλέπεται διαφορετική συμπεριφορά ανάλογα με το μέγεθος κώδικα αλλά και με την κατασκευή του πίνακα ελέγχου ισοτιμίας. Παράλληλα, δίνουμε και τη δυνατότητα σύγκρισης και με σύστημα χωρίς κωδικοποίηση, παρατηρώντας ότι το σύστημα με κωδικοποίηση υπερτερεί ακόμα και με μία μόνο επανάληψη, με αυξητική μάλιστα τάση υπεροχής για αυξανόμενες τιμές του σηματοθορυβικού λόγου.



Σχήμα 5.4 Επίδοση για 10, 40 και 80 επαναλήψεις (iterations)



Σχήμα 5.5 Επίδοση χωρίς κωδικοποίηση και επίδοση για 1, 5, 10, 15, 40 και 80 επαναλήψεις

5.2 Επίδοση του LDPC(256, 128) Κώδικα σε σχέση με διάφορα Σχήματα Διαμόρφωσης και διάφορους Τύπους Διαύλου

Αφού μελετήσαμε τα εγγενή χαρακτηριστικά του κώδικα, επόμενος μας στόχος είναι η μελέτη της επίδοσης του όταν μεταβάλλονται εξωτερικοί από τον κώδικα παράγοντες, όπως η τεχνική διαμόρφωσης ή ο τύπος διαύλου. Σε όλη αυτή την ενότητα χρησιμοποιείται ο, ίδιος με προηγουμένως, **LDPC(256, 128)** κώδικας με μέγιστο αριθμό επαναλήψεων του αλγορίθμου αποκωδικοποίησης ίσο με **80.** Ακολουθούν τα αποτελέσματα της προσομοίωσης μας, πάλι μέσω του περιβάλλοντος Matlab.

5.2.1 Διαμόρφωση 4-QAM και δίαυλος τύπου Rayleigh

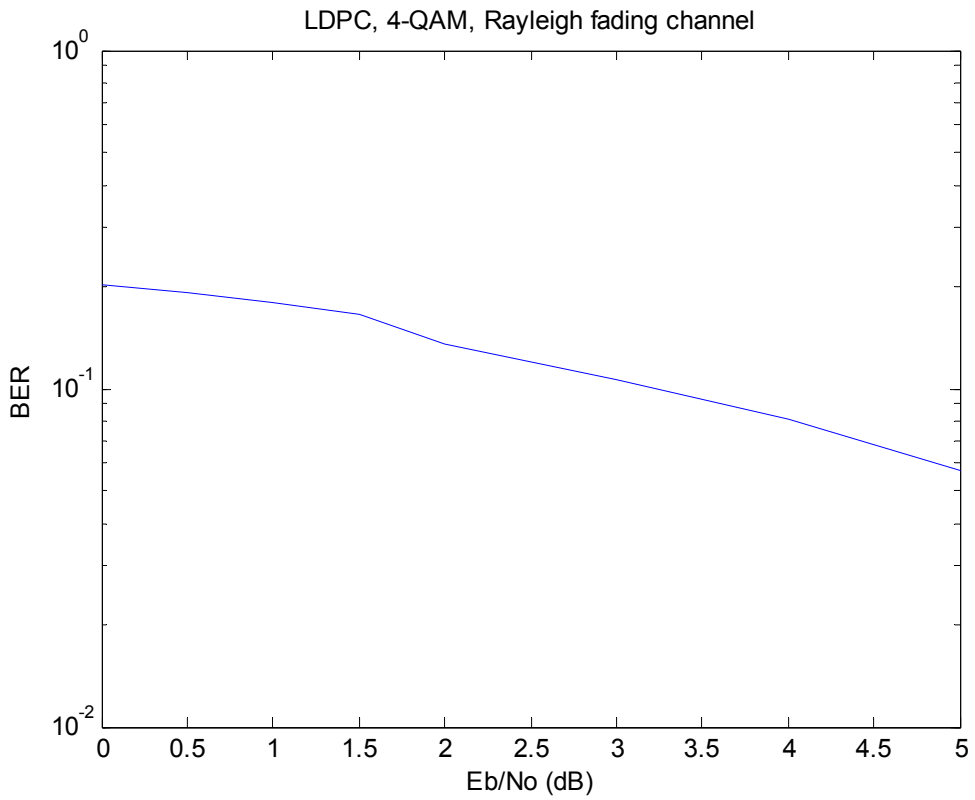
Εμπλουτίζοντας την εργασία μας, προσομοιώνουμε και άλλες διαμορφώσεις εκτός από την "απλή" BPSK ενώ προσομοιώνουμε και διαύλους που περιγράφουν καλύτερα από ότι το μοντέλο AWGN περιβάλλοντα κινητών επικοινωνιών, όπου ως γνωστόν η πολυδιαδρομική διάδοση προκαλεί διαλείψεις. Αρχικά λοιπόν, εξετάζουμε την επίδοση του LDPC κώδικα μας υπό 4-QAM διαμόρφωση και υπό τύπου Rayleigh δίαυλο με διαλείψεις.

Τα αποτελέσματα της προσομοίωσης μας δίνονται στα σχήματα **5.6**, **5.7**. Η επίδοση μετριέται πάλι σε όρους **BER** (Bit Error Rate) συναρτήσει τώρα του λόγου E_b / N_o, του λόγου δηλαδή της ενέργειας ψηφίου (bit) προς την πυκνότητα θορύβου. Υπενθυμίζουμε τη σχέση που συνδέει το σηματοθορυβικό λόγο SNR με το λόγο ενέργειας ψηφίου προς πυκνότητα θορύβου E_b / N_o:

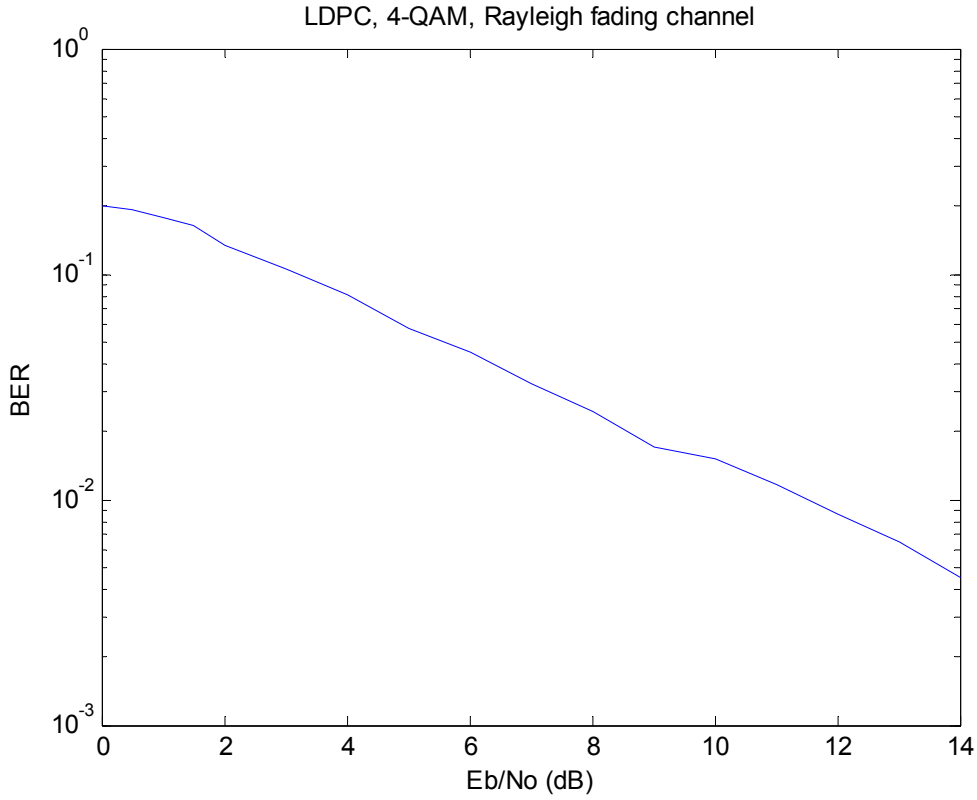
$$SNR = \frac{S}{N} = \frac{E_b}{N_0} \cdot \frac{R_i}{B}$$

Στην προσομοίωση μας θεωρήσαμε ότι ο ρυθμός μετάδοσης R_i καταλαμβάνει όλο το διαθέσιμο εύρος ζώνης του συστήματος B. Συνεπώς, σε ότι μας αφορά, R_i=B άρα για τα αποτελέσματα των προσομοιώσεων μας οι λόγοι SNR και E_b / N_o είναι **ισοδύναμοι**. Να σημειώσουμε ότι σε σχετικές καμπύλες στη βιβλιογραφία ψηφιακών τηλεπικοινωνιακών συστημάτων χρησιμοποιείται κυρίως ο λόγος E_b / N_o. Στο σχήμα **5.6** τώρα, παρατηρούμε τη γνωστή μας πια πτώση του ρυθμού λαθών με την αύξηση του σηματοθορυβικού λόγου, όμως η επίδοση είναι πολύ χαμηλότερη συγκρίνοντας τη με προηγούμενα αποτελέσματα, όπως φαίνεται στο σχήμα 5.2 για παράδειγμα. Μάλιστα, ακόμα και για πολύ μεγάλες τιμές του σηματοθορυβικού λόγου δεν παρατηρείται σημαντική βελτίωση της επίδοσης, όπως χαρακτηριστικά φαίνεται στο σχήμα **5.7**. Αυτή η πτώση στην επίδοση του συστήματος, σε σύγκριση με την επίδοση για BPSK διαμόρφωση και AWGN δίαυλο, οφείλεται στους κάτωθι παράγοντες:

- Ο δίαυλος μας τώρα χαρακτηρίζεται από διαλείψεις, επιβαρύνει λοιπόν το σήμα πολύ περισσότερο από τον AWGN δίαυλο όπου το σήμα δεν υποφέρει από διαλείψεις.
- Η 4-QAM ως υψηλότερης φασματικής απόδοσης τεχνική από ότι η BPSK, προκαλεί αύξηση του ρυθμού λαθών για ίδιο σηματοθορυβικό λόγο (trade-off). Ανάλογη ανάλυση έγινε και στην ενότητα περί αποδιαμόρφωσης (3.5.5).



Σχήμα 5.6 Διαμόρφωση 4-QAM και δίαυλος τύπου Rayleigh



Σχήμα 5.7 Διαμόρφωση 4-QAM και δίαυλος τύπου Rayleigh

5.2.2 Διαμόρφωση 4-QAM και δίαυλος τύπου Rice

Συνεχίζοντας με το ίδιο σχήμα διαμόρφωσης (4-QAM), μεταβάλλουμε το δίαυλο σε τύπου Rice. Υπενθυμίζουμε ότι σε δίαυλο Rice έχουμε διαλείψεις, υπάρχει όμως και οπτική επαφή πομπού-δέκτη κάτι που δεν συμβαίνει σε δίαυλο Rayleigh. Ξεχωρίζουμε λοιπόν δύο ειδών κύματα που καταφθάνουν στο δέκτη: το απευθείας κύμα (direct wave) και τα κύματα που καταφθάνουν σκεδασμένα (scattered). Κεφαλαιώδους σημασίας για το χαρακτηρισμό ενός διαύλου Rice είναι με πόσο μεγαλύτερη ισχύ καταφθάνει το απευθείας κύμα από ότι τα σκεδασμένα. Ορίζεται έτσι ο παράγοντας **K** (Rician K factor) ως

$$K = \frac{\text{ΙΣΧΥΣ ΑΠΕΥΘΕΙΑΣ ΚΥΜΑΤΟΣ}}{\text{ΙΣΧΥΣ ΣΚΕΔΑΣΜΕΝΩΝ ΚΥΜΑΤΩΝ}} \tag{5.1}$$

όπου όπως προαναφέρφηκε μας ενδιαφέρει η ληφθείσα ισχύς στο δέκτη. Με μια πρώτη ματιά, βλέπουμε ότι για K πολύ μικρό ($K \to 0$) είναι σαν να μην υπάρχει οπτική επαφή πομπού – δέκτη, οπότε ο δίαυλος αναμένεται να εμφανίζει χαρακτηριστικά Rayleigh διαύλου. Αντίθετα, για K πολύ μεγάλο ($K \to \infty$) κυριαρχεί απολύτως η απευθείας διάδοση χωρίς να επηρεάζεται από φαινόμενα πολύοδης διάδοσης. Αναμένουμε λοιπόν ότι σε τέτοια περίπτωση ο δίαυλος θα τείνει να αποκτήσει τα χαρακτηριστικά του AWGN μοντέλου.

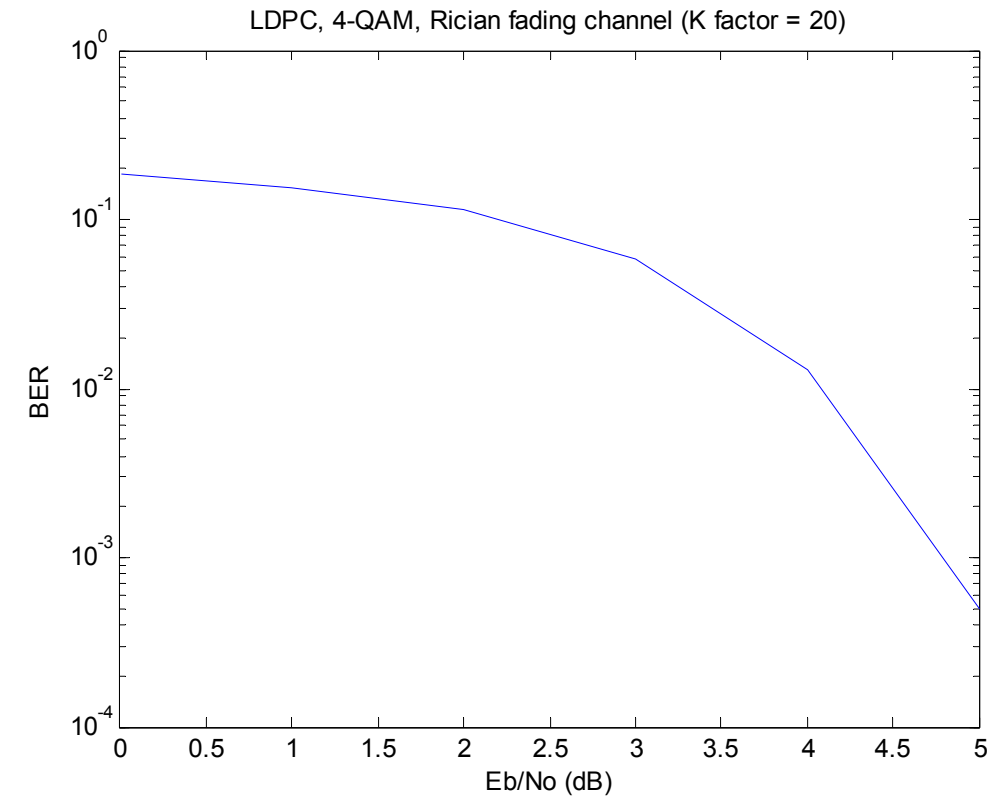
Στην προσομοίωση μας, υλοποιήθηκε δίαυλος Rice με **K=20**. Η γραφική απεικόνιση του ρυθμού λαθών ανά ψηφίο (BER) συναρτήσει του λόγου E_b / N_o αποδίδεται στα σχήματα **5.8**, **5.9**. Επιβεβαιώνεται ότι η επίδοση σε δίαυλο Rice είναι καλύτερη από ότι σε δίαυλο Rayleigh. Μάλιστα, για K=20 η διαφορά είναι σημαντική. Απομένει να συγκρίνουμε το δίαυλο Rice με δίαυλο τύπου AWGN κάτι που θα γίνει στη συνέχεια. Τελικός στόχος αυτής της επί μέρους πορείας είναι η παρουσίαση κοινού συγκριτικού διάγραμματος για διαύλους τύπου AWGN, Rice, Rayleigh.

5.2.3 Διαμόρφωση 4-QAM και δίαυλος τύπου AWGN

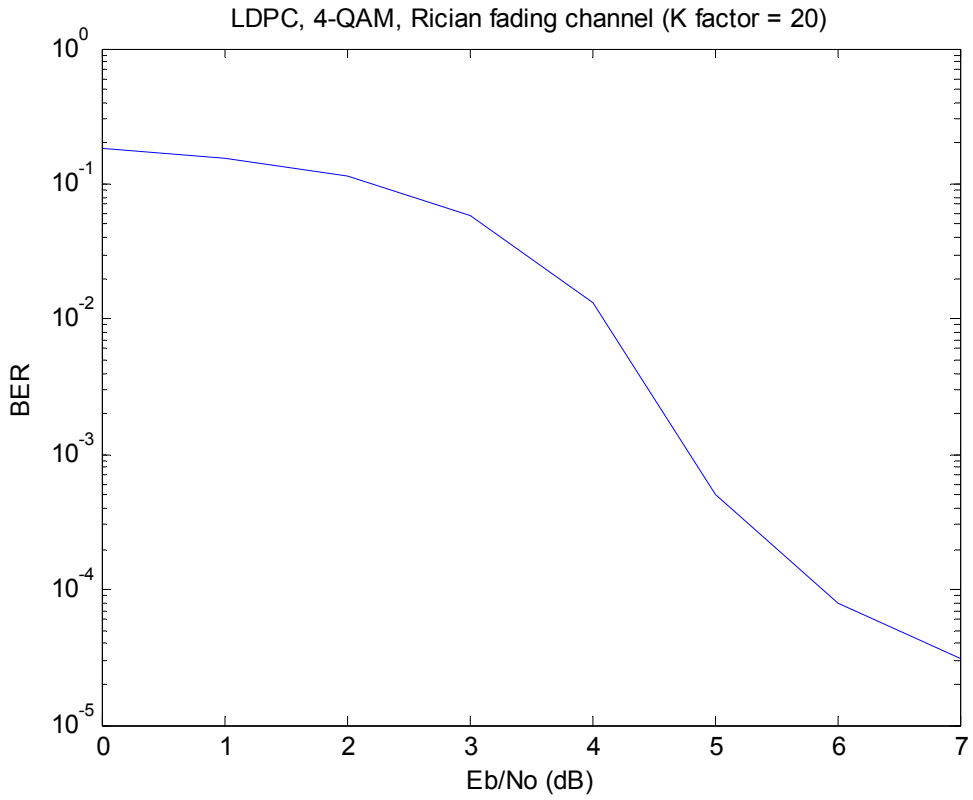
Συνεχίζοντας στην πορεία σύγκρισης των τύπων διαύλου, κρατάμε σταθερή τη διαμόρφωση σε 4-QAM και, βέβαια, χρησιμοποιούμε την ίδια LDPC κωδικοποίηση. Το αποτέλεσμα αυτής της προσομοίωσης δίνεται στο σχήμα **5.10**. Η επίδοση είναι σαφώς καλύτερη από ότι με Rayleigh δίαυλο, όμως για τη σύγκριση με δίαυλο Rice απαιτείται κοινό διάγραμμα κάτι που θα γίνει στη συνέχεια.

5.2.4 Σύγκριση Επίδοσης για διαύλους τύπου AWGN, Rice, Rayleigh

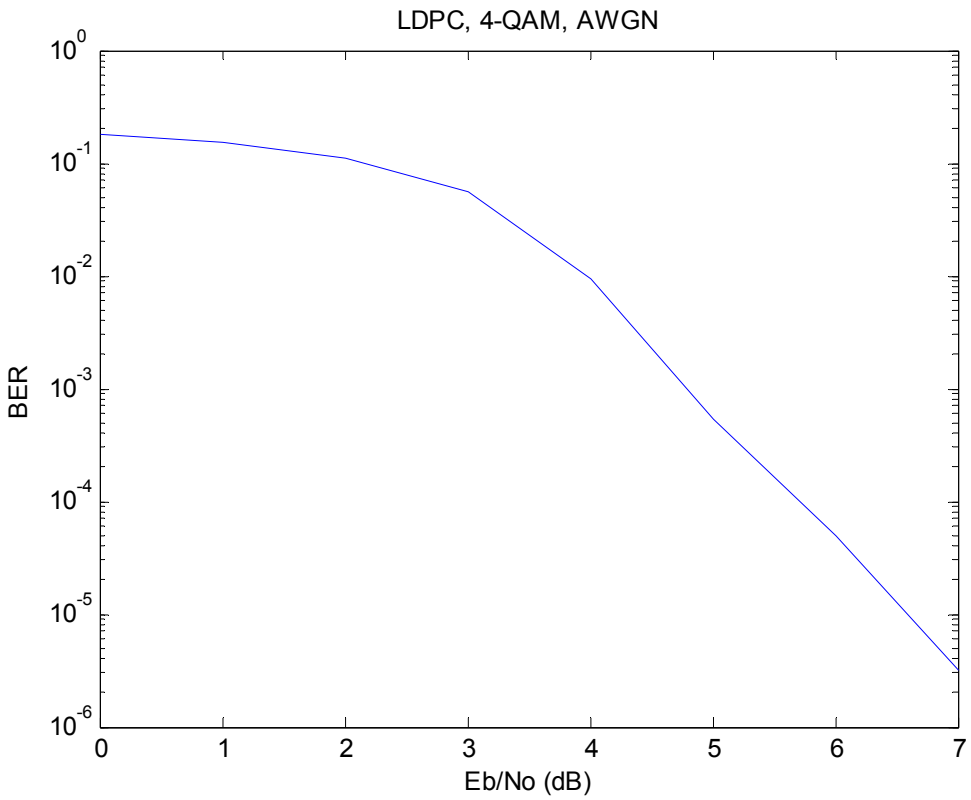
Τα συγκριτικά διαγράμματα επίδοσης φαίνονται στα σχήματα **5.11, 5.12**. Από το σχήμα 5.11 φαίνεται ότι ο συγκεκριμένος δίαυλος Rice οδηγεί σε συγκρίσιμη επίδοση με AWGN δίαυλο. Βέβαια, από $E_b / N_o >$ 6 dB υπάρχει σαφής υπεροχή σε AWGN δίαυλο, ενώ από την κλίση των γραφικών παραστάσεων προβλέπεται αύξηση της



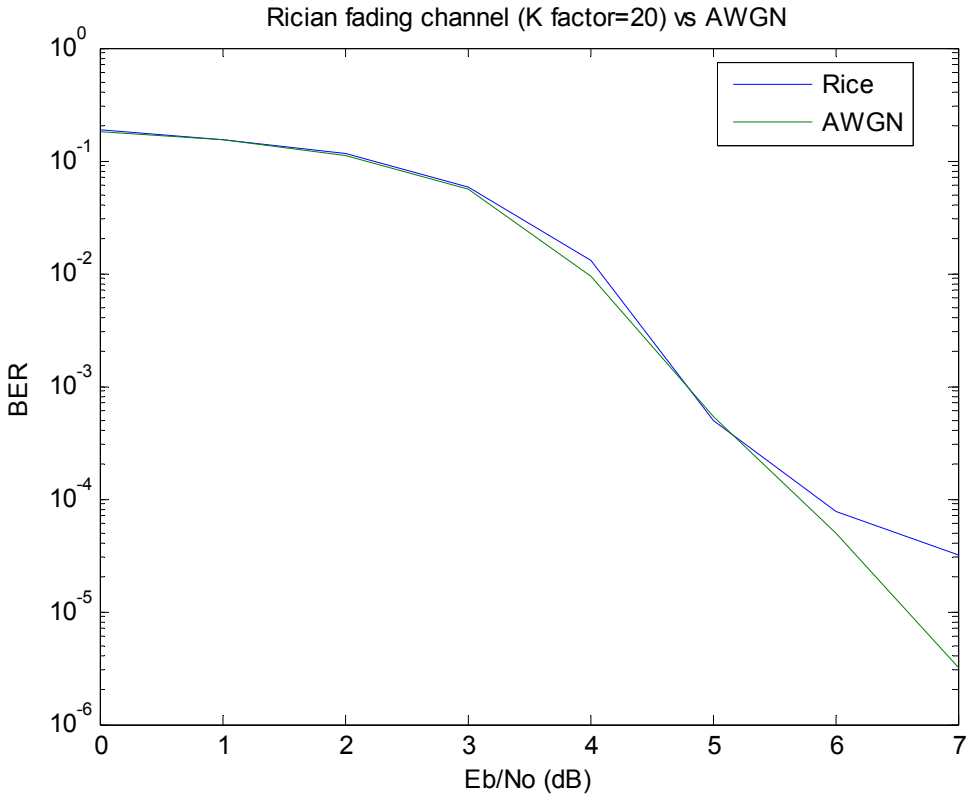
Σχήμα 5.8 Διαμόρφωση 4-QAM και δίαυλος τύπου Rice (K=20)



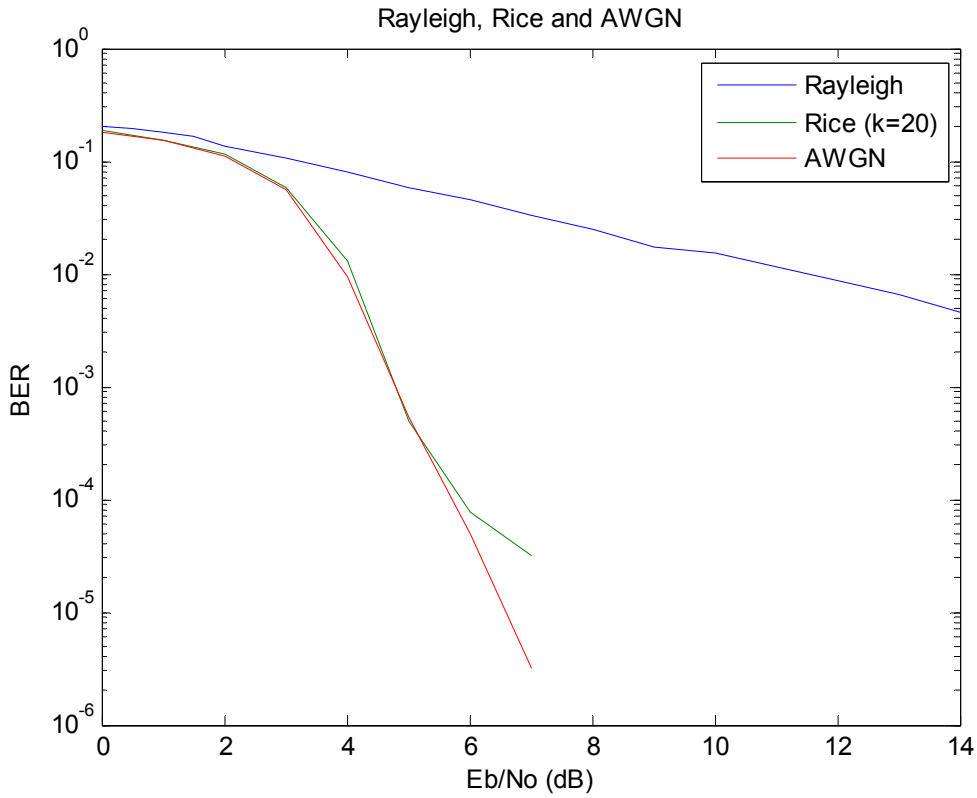
Σχήμα 5.9 Διαμόρφωση 4-QAM και δίαυλος τύπου Rice (K=20)



Σχήμα 5.10 Διαμόρφωση 4-QAM και δίαυλος τύπου AWGN



Σχήμα 5.11 Σύγκριση AWGN, Rice (K=20)

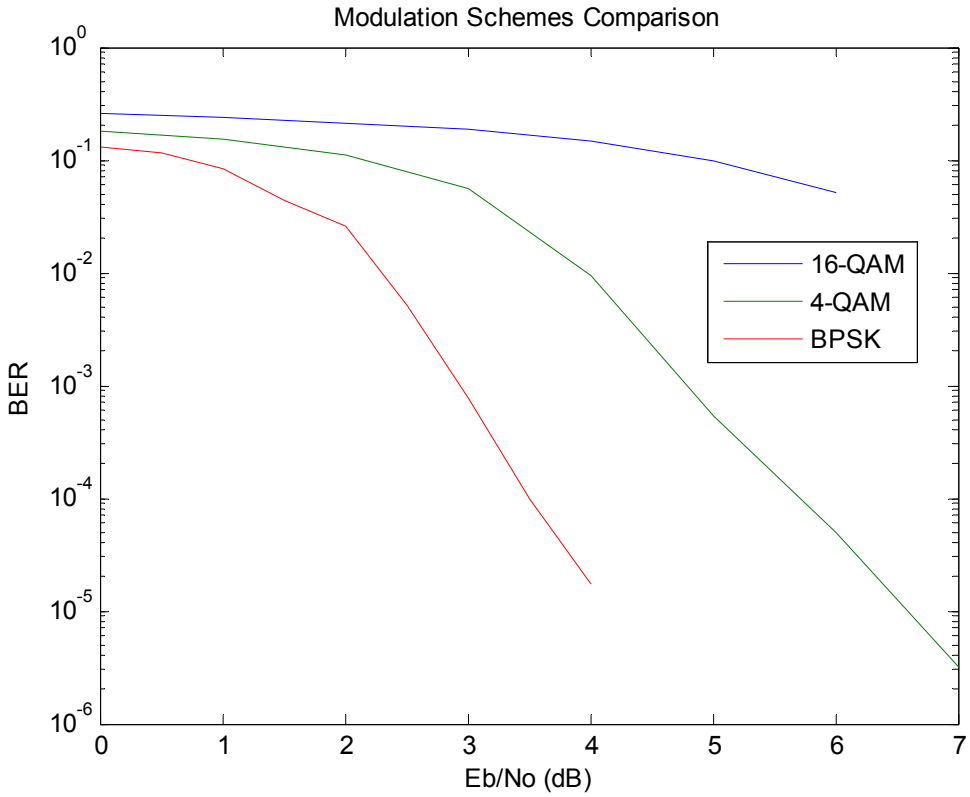


Σχήμα 5.12 Σύγκριση AWGN, Rice (K=20), Rayleigh

μεταξύ τους διαφοράς. Παραταύτα, το σχήμα 5.12 αποδεικνύει ότι για K=20 ο δίαυλος Rice συμπεριφέρεται πολύ περισσότερο σαν AWGN παρά σαν Rayleigh. Συμπερασματικά λοιπόν, αναφερόμενοι στους εξεταζόμενους διαύλους, έχουμε στο ένα άκρο το δίαυλο Rayleigh (μέγιστη επιβάρυνση) και στο άλλο το δίαυλο AWGN (ελάχιστη επιβάρυνση). Ανάμεσα τους ως προς την εισαγόμενη επιβάρυνση είναι ο δίαυλος Rice. Ο ρόλος της προσομοίωσης είναι να μελετήσει τη συμπεριφορά του διαύλου Rice για τις διάφορες τιμές του παράγοντα K, συγκρίνοντας τον με τους άλλους δύο διαύλους. Για παράδειγμα, εμείς εξάγαμε το συμπέρασμα ότι για K=20 δηλαδή όταν η ισχύς του απευθείας κύματος είναι 20 φορές μεγαλύτερη από την ισχύ των κυμάτων που καταφθάνουν ύστερα από σκέδαση, τότε ο δίαυλος Rice εμφανίζει χαρακτηριστικά πολύ μεγαλύτερης ομοιότητας με AWGN δίαυλο παρά με Rayleigh δίαυλο.

5.2.5 Σύγκριση μεταξύ των Τεχνικών Διαμόρφωσης

Έχουμε ήδη μελετήσει την επίδοση του κώδικα πάνω από από AWGN δίαυλο και με BPSK διαμόρφωση και με 4-QAM διαμόρφωση. Επόμενο σενάριο ήταν η μελέτη της επίδοσης χρησιμοποιώντας 16-QAM διαμόρφωση. Μετά την υλοποίηση και αυτού του σεναρίου ήταν δυνατή η κατασκευή κοινού συγκριτικού διαγράμματος για τις 3 διαμορφώσεις. Έτσι λοιπόν παίρνουμε το σχήμα **5.13** όπου φαίνεται ότι καλύτερη επίδοση επιτυγχάνεται με BPSK και χειρότερη με 16QAM. Επιβεβαιώνεται



Σχήμα 5.13 Σύγκριση BPSK, 4-QAM, 16-QAM (με LDPC κωδικοποίηση)

δηλαδή ότι τα σχήματα χαμηλότερης φασματικής απόδοσης εμφανίζουν καλύτερη επίδοση, μικρότερη πιθανότητα λάθους για ίδιο σηματοθορυβικό λόγο δηλαδή. Παραταύτα, η επιλογή ενός σχήματος διαμόρφωσης δεν καθορίζεται μόνο από τη σύγκριση της μεταξύ τους επίδοσης, από καμπύλες σαν του σχήματος 5.13 δηλαδή. Παράλληλα, πρέπει να συγκριθεί και η φασματική τους απόδοση. Για παράδειγμα, ανάμεσα στις 3 συγκρινόμενες τεχνικές διαμόρφωσης υπάρχει μια αντίστροφη σχέση επίδοσης – φασματικής απόδοσης. Θεωρώντας τη μετάδοση μιας κωδικής λέξης μήκους 256 bits του προσομοιούμενου συστήματος μας έχουμε:

- για τη μετάδοση της χρησιμοποιώντας BPSK διαμόρφωση απαιτείται η εκπομπή (256 / 1)=256 συμβόλων.
- για τη μετάδοση της χρησιμοποιώντας 4-QAM διαμόρφωση απαιτείται η εκπομπή (256 / $\log_2 4$)=128 συμβόλων.
- για τη μετάδοση της χρησιμοποιώντας 16-QAM διαμόρφωση απαιτείται η εκπομπή (256 / $\log_2 16$)=64 συμβόλων

Άρα, με BPSK διαμόρφωση καταναλώνεται διπλάσιο εύρος ζώνης σε σύγκριση με 4-QAM και τετραπλάσιο σε σύγκριση με 16-QAM. Η τελική επιλογή λοιπόν για το σύστημα θα εξαρτηθεί πέρα από την προσδοκώμενη επίδοση και από τη διαθεσιμότητα εύρους ζώνης. Το όλο ζήτημα σύγκρισης τεχνικών διαμόρφωσης απασχολεί γενικότερα τους σχεδιαστές τηλεπικοινωνιακών συστημάτων, με μια από τις πλέον αποδεκτές λύσεις για σύγχρονα συστήματα να είναι η προσαρμοστική διαμόρφωση (adaptive modulation).

6 *Εφαρμογές των LDPC κωδίκων – Σύγχρονη Έρευνα*

Το κεφάλαιο αυτό έχει διττό στόχο. Πρώτον, να παρουσιαστούν οι κυριότερες πρακτικές εφαρμογές των LDPC κωδίκων σε ασύρματα συστήματα νέας γενιάς και δεύτερον να γίνει μια συνοπτική παρουσίαση των ερευνητικών θεμάτων γύρω από τους LDPC κώδικες. Είναι αλήθεια ότι και οι δύο παραπάνω περιοχές είναι πλούσιες. Οι πρακτικές εφαρμογές λόγω των εξαιρετικών επιδόσεων που εμφανίζουν οι LDPC κώδικες και τα ερευνητικά θέματα λόγω του ότι το ενδιαφέρον της επιστημονικής κοινότητας για αυτούς τους κώδικες είναι σχετικά πρόσφατο και εξαιτίας του ότι συνεχώς προκύπτουν νέες απαιτήσεις για τα τηλεπικοινωνιακά συστήματα, οπότε απαιτείται επανεξέταση όλων των βαθμίδων άρα και της κωδικοποίησης.

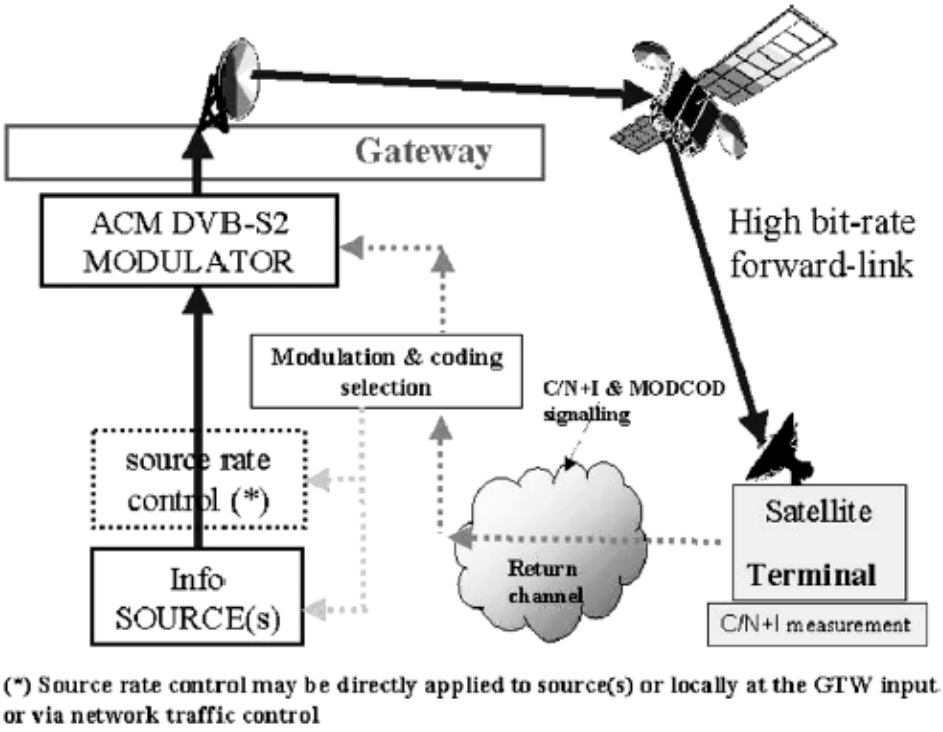
6.1 Πρακτικές Εφαρμογές

Οι LDPC κώδικες, μαζί με τους Turbo κώδικες, έχουν τις καλύτερες επιδόσεις από όλους τους γνωστούς κώδικες μέχρι σήμερα. Έτσι, έχουν υιοθετηθεί από διάφορα τηλεπικοινωνιακά πρότυπα και αναμένεται η χρήση τους να διαδοθεί ακόμα περισσότερο στο μέλλον. Οι πλέον γνωστές εφαρμογές των LDPC κωδίκων αφορούν το νέο πρότυπο δορυφορικής ευρυεκπομπής **DVB-S2**, κυρίως, καθώς και το πρότυπο **IEEE 802.16e** (WiMAX).

6.1.1 LDPC κώδικες για το DVB-S2

Το DVB-S2 αποτελεί τη δεύτερη γενιά του, ιδιαίτερα διαδεδομένου, προτύπου DVB-S (Digital Video Broadcasting-Satellite). Το νέο αυτό πρότυπο έχει σχεδιασθεί για να εξυπηρετεί μία μεγάλη ποικιλία ευρυζωνικών υπηρεσιών και εφαρμογών, όπως εμπορική τηλεόραση, υψηλής ευκρίνειας τηλεόραση (HDTV), διαδραστικές υπηρεσίες (Internet), DSNG (Digital Satellite News Gathering), διανομή TV σε επίγειους πομπούς VHF/UHF, διανομή περιεχομένου (content delivery) και υπηρεσίες κορμού Internet. Ένα γενικό λειτουργικό διάγραμμα δίνεται στο σχήμα 6.1.

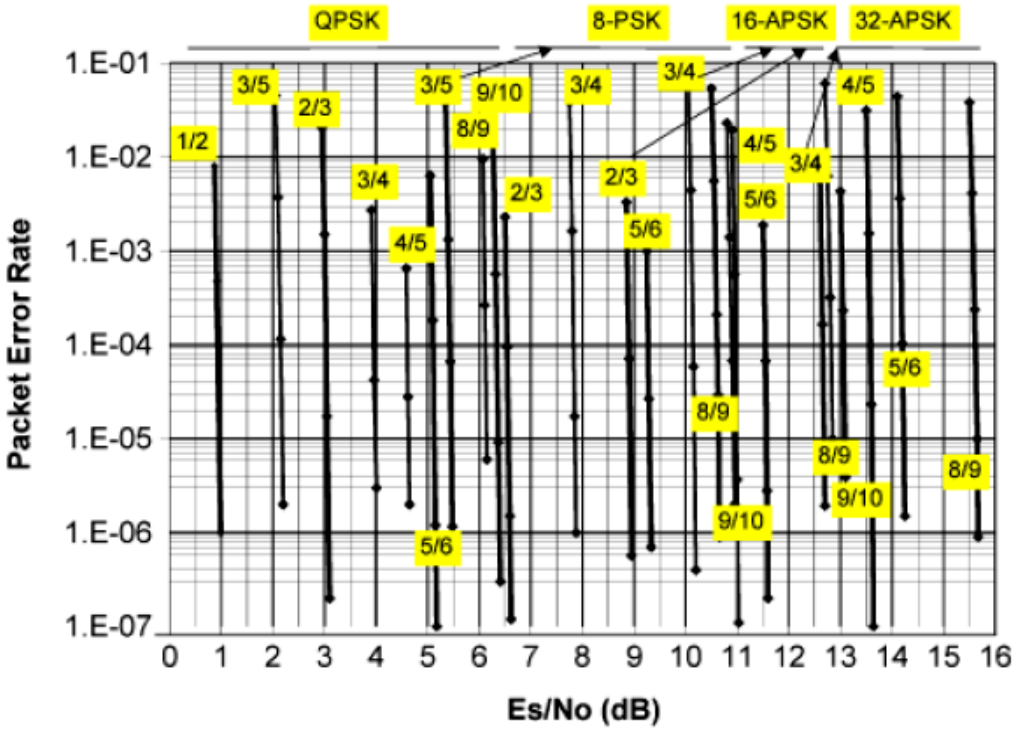
Όσον αφορά την κωδικοποίηση, να επισημάνουμε αρχικά ότι σε συστήματα ευρυεκπομπής (broadcasting) όπως το DVB-S2 η τεχνική ARQ απορρίπτεται για πρακτικούς λόγους. Έτσι, τέτοια συστήματα χρησιμοποιούν υποχρεωτικά πρόσθια διόρθωση λαθών (FEC). Μάλιστα, λόγω των ιδιαίτερων απαιτήσεων του DVB-S2 αλλά και λόγω των σημαντικών απωλειών που εισάγει ο δορυφορικός δίαυλος η επίδοση του χρησιμοποιούμενου κώδικα διόρθωση λαθών πρέπει να είναι εξαιρετική. Μετά απο διεξοδικές συγκρίσεις, μέσω προσομοιώσεων, μεταξύ αλυσιδωτών συνελικτικών κωδίκων, Turbo κωδίκων και LDPC κωδίκων επελέγησαν τελικά οι LDPC κώδικες. Φαίνεται λοιπόν και ο πρωτεύον ρόλος της προσομοίωσης στα σύγχρονα τηλεπικοινωνιακά συστήματα. Ο κώδικας που τελικά υιοθετήθηκε έχει μήκος κωδικής λέξης **n=64800** bits και αυτό γιατί όπως γνωρίζουμε οι εξαιρετικές επιδόσεις των LDPC κωδίκων επιτυγχάνονται για πολύ μεγάλες τιμές του μήκους της



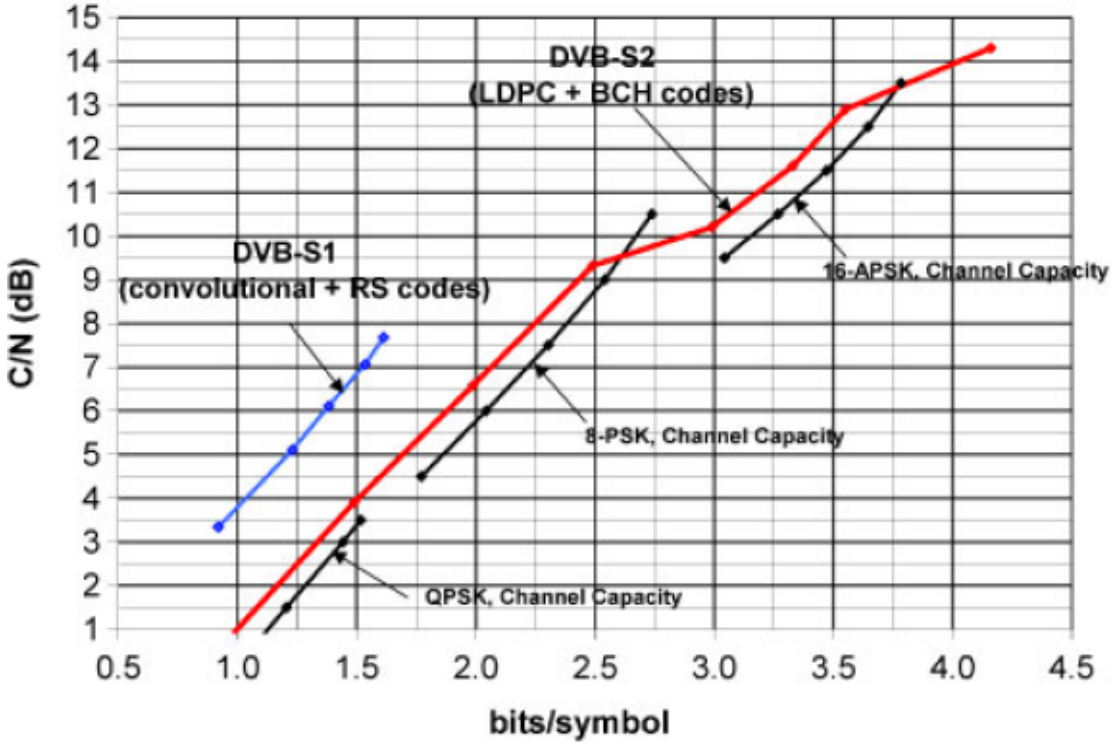
Σχήμα 6.1 Μπλοκ Διάγραμμα Συστήματος DVB-S2

κωδικής λέξης. Παράλληλα, υπάρχει η δυνατότητα επιλογής από το σύστημα εναλλακτικού LDPC κώδικα με μήκος n=16200 bits. Αυτό το μικρότερο μέγεθος κώδικα συνεπάγεται και μικρότερη καθυστέρηση κατά τις διαδικασίες κωδικοποίησης / αποκωδικοποίησης και επιλέγεται για εφαρμογές πραγματικού χρόνου, δηλαδή για εφαρμογές ευαίσθητες ως προς την καθυστέρηση. Για τις υπόλοιπες εφαρμογές χρησιμοποιείται το μπλοκ των 64800 bits που επιτυγχάνει συγκριτικά καλύτερες επιδόσεις για ίδιο σηματοθορυβικό λόγο. Επιπλέον, για πληρέστερη περιγραφή της μονάδας FEC του προτύπου DVB-S2 να σημειώσουμε ότι χρησιμοποιείται και εξωτερικός κώδικας (outer code) **BCH** για τη διασφάλιση κατώτατου ορίου σφαλμάτων (Σχ. 6.3). Σαν αποτέλεσμα των παραπάνω, με τη χρήση των LDPC κωδίκων έγινε εφικτή η αύξηση της επίδοσης κατά 35% σε σύγκριση με το προϋπάρχον DVB-S1 όπου χρησιμοποιούνταν συνελικτικοί και Reed-Solomon (RS) κώδικες. Η επίδοση του DVB-S2 παρουσιάζεται στα σχήματα 6.2, 6.3.

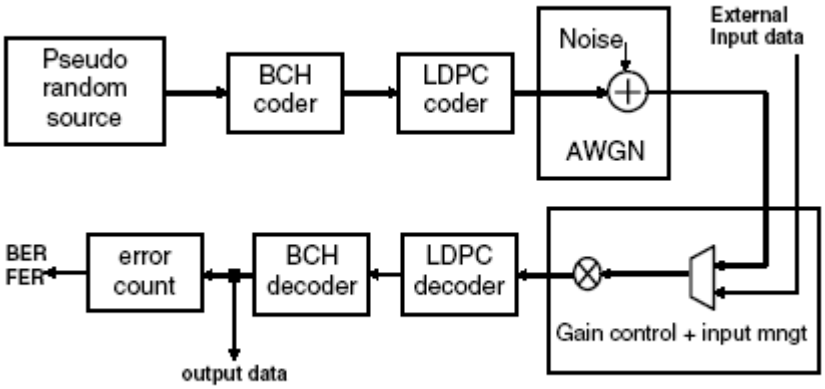
Παράλληλα, το μεγάλο μέγεθος του κώδικα εγείρει ζήτημα πολυπλοκότητας στις διαδικασίες κωδικοποίησης / αποκωδικοποίησης. Η λύση που προτάθηκε είναι κωδικοποίηση χωρίς τη χρήση γεννήτορα πίνακα **G**, αλλά μέσω αναδρομικής διαδικασίας βασισμένης στον πίνακα ελέγχου ισοτιμίας **H**. Έτσι, ελευθερώνεται αποθηκευτικός χώρος και γλιτώνουμε από τον πολλαπλασιασμό πινάκων ($c = m \circ G$), διεργασία με σημαντική υπολογιστική πολυπλοκότητα. Πιο συγκεκριμένα, ο πίνακας ελέγχου ισοτιμίας **H** είναι της μορφής $H_{(n-k)x(n)} = \left[A_{(n-k)x(k)} B_{(n-k)x(n-k)} \right]$. Ο υποπίνακας **A** διαμορφώνεται με βάση τα bits πληροφορίας (information bits) ενώ ο υποπίνακας **B** είναι της μορφής (6.1) όπως φαίνεται στη συνέχεια



Σχήμα 6.2 Επίδοση DVB-S2 (LDPC & BCH) για AWGN δίαυλο και n=64800



Σχήμα 6.3 Σύγκριση DVB-S2, DVB-S1 και χωρητικότητας διαύλου (channel capacity)



Σχήμα 6.4 Κωδικοποίηση / Αποκωδικοποίηση στο DVB-S2

$$B = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 1 \end{bmatrix} \tag{6.1}$$

Δηλαδή, ο **H** αποτελείται από ένα στατικό υποπίνακα (**B**) και από ένα δυναμικά διαμορφώμενο υποπίνακα (**A**). Αν τώρα συμβολίσουμε την κωδική λέξη ως $c = (u_1, \dots, u_K, p_1, \dots, p_M)$ όπου u_k είναι τα bits πληροφορίας για 1<=k<=K και p_m είναι τα bits ισοτιμίας για 1<=m<=M με M=N-K, τότε η "γρήγορη" κωδικοποίηση μπορεί να επιτευχθεί αναδρομικά ως

$$\begin{aligned} p_1 &= \sum_{k=1}^{K} u_k h_{1,k} \\ p_m &= p_{m-1} + \sum_{k=1}^{K} u_k h_{m,k}, \; 2 \le \mathrm{m} \le \mathrm{M} \end{aligned} \tag{6.2}$$

όπου με $h_{m,k}$ συμβολίζουμε το στοιχείο (m, k) του πίνακα **H.** Όπως προκύπτει, δεν υπάρχει πουθενά ο γεννήτορας πίνακας **G**. Έχει αναφερθεί ότι η απώλεια στην επίδοση, σε σχέση με την κλασσική κωδικοποίηση, λόγω αυτής της διαδικασίας είναι της τάξης του 0.1 dB και θεωρείται αμελητέα για τη χρήση του DVB-S2 [9]. Αυτή που δεν είναι αμελητέα είναι η ελάττωση της πολυπλοκότητας μέσω αυτής της αναδρομικής διαδικασίας.

Επιπλέον, στο DVB-S2 χρησιμοποιείται μια αρκετά διαδεδομένη τεχνική που δεν έχουμε αναφερθεί μέχρι τώρα στην εργασία: η προσαρμοστική κωδικοποίηση (adaptive coding). Υπενθυμίζουμε ότι αν n το μήκος της κωδικής λέξης και k από τα n bits αντιστοιχούν σε ψηφία πληροφορίας, τότε ορίζουμε το ρυθμό κώδικα (code rate) ως **r** = **k**/**n**. Μικρές τιμές του r αντιστοιχούν σε περισσότερα πλεονάζοντα ψηφία, συνεπώς και σε μεγαλύτερη διορθωτική ικανότητα. Όμως, μεταδίδονται λιγότερα ψηφία πληροφορίας. Αντίθετα, μεγάλες τιμές του r εμφανίζουν μειωμένη

διορθωτική ικανότητα αλλά επιτρέπουν την μετάδοση περισσότερων ψηφίων πληροφορίας. Η προσαρμοστική κωδικοποίηση συνίσταται στο να γίνεται χρήση κωδικοποίησης με **r** κοντά στο 1 όταν ισχύουν ευνοϊκές συνθήκες στο δίαυλο (π.χ. συνθήκες καθαρού ουρανού) και μετάβαση σε περισσότερο ανθεκτικά / εύρωστα σχήματα κωδικοποίησης (με μικρότερο **r**) όταν ισχύουν δυσμενείς συνθήκες στο δίαυλο (π.χ. υπό ισχυρή βροχόπτωση). Στο DVB-S2 μπορεί να επιλεγεί ένας από τους ακόλουθους ρυθμούς κώδικα: 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9, 9/10, όπως φαίνεται και στο σχήμα 6.2. Η επιλογή γίνεται και με βάση ποιο σχήμα διαμόρφωσης έχει επιλεγεί αφού χρησιμοποείται και προσαρμοστική διαμόρφωση(**QPSK**, **8PSK**, **16APSK** ή **32APSK**). Η τεχνική συνδυαστικής χρήσης προσαρμοστικής κωδικοποίησης (AC) και προσαρμοστικής διαμόρφωσης (AM) ονομάζεται τεχνική **ACM** και είναι από τα χαρακτηριστικά στοιχεία του DVB-S2, όπως φαίνεται και στο σχήμα 6.1.

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

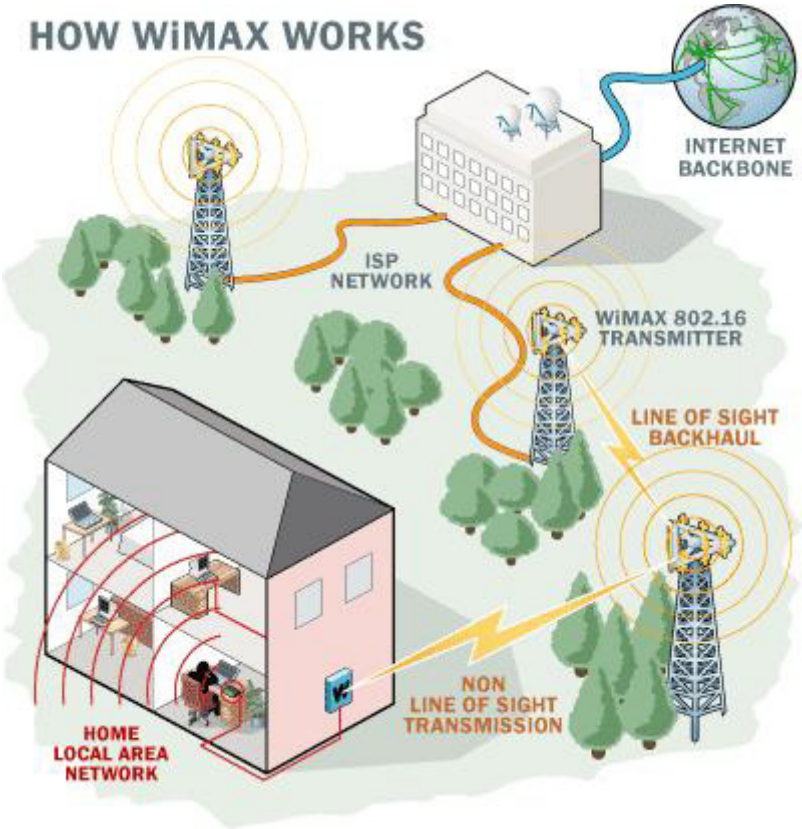
- Βέλτιστη επίδοση
- Πλήρης ευελιξία
- Ανεκτή πολυπλοκότητα δέκτη

Στον τομέα που μας αφορά στην εργασία, ο στόχος της επίδοσης οδήγησε στην επιλογή LDPC κώδικα πολύ μεγάλου μεγέθους, η ευελιξία καλύπτεται από την τεχνική ACM και η πολυπλοκότητα του δέκτη ελαττώθηκε μέσω της αναδρομικής διαδικασίας κωδικοποίησης που γίνεται χωρίς τη χρήση γεννήτορα πίνακα.

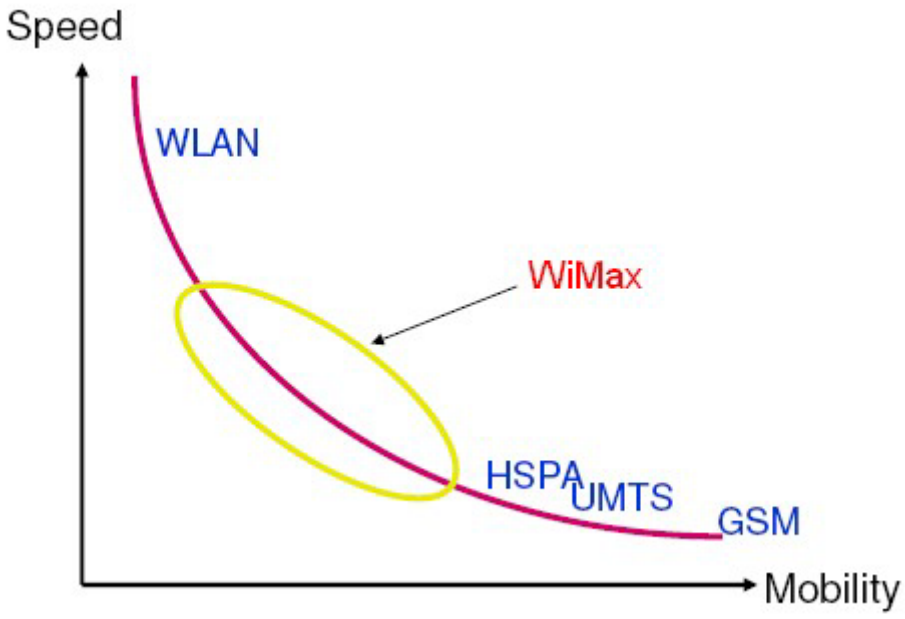
6.1.2 WiMAX και LDPC Κώδικες

Το **WiMAX** (**W**orldwide **I**nteroperability for **M**icrowave **Acc**ess) είναι μια νέα πολλά υποσχόμενη τεχνολογία ευρυζωνικής ασύρματης δικτύωσης. Θα μπορούσαμε να παραλληλίσουμε τη λειτουργία του WiMAX με τη γνωστή μας τεχνολογία **Wi-Fi**, με τη βασική διαφορά όμως ότι το WiMAX δημιουργήθηκε για να έχει πολύ μεγαλύτερη εμβέλεια. Συγκεκριμένα, ενώ το Wi-Fi υποστηρίζει εμβέλεια επικοινωνίας έως 100 μέτρα, το WiMAX φτάνει τα 35 χιλιόμετρα ή και παραπάνω. Έτσι λοιπόν, τα τελευταία χρόνια το Wi-Fi επιτρέπει την ασύρματη πρόσβαση στο διαδίκτυο (Internet) σε μικρή εμβέλεια γύρω από τα λεγόμενα hotspots, καλύπτοντας με αυτόν τον τρόπο περιοχές όπως αεροδρόμια ή αίθουσες συνεδριάσεων. Το WiMAX στοχεύει να κάνει το ίδιο σε εμβέλεια ολόκληρης πόλης, μέσω του σήματος των παροχέων υπηρεσιών Internet (ISPs) με τρόπο που φαίνεται στο σχήμα 6.5. Μάλιστα, ο τελικός χρήστης της ασύρματης ευρυζωνικής πρόσβασης απαιτείται να εγκαταστήσει στο δέκτη του (π.χ. υπολογιστής) απλά μια κάρτα δικτύωσης WiMAX, όπως γίνεται μέχρι σήμερα με τις κάρτες δικτύωσης Wi-Fi.

Βασικός στόχος του WiMAX είναι και η υποστήριξη κινητικότητας του χρήστη. Έτσι, ενώ σε ασύρματο τοπικό δίκτυο (WLAN) η κινητικότητα είναι περιορισμένη, ο χρήστης του WiMAX θα μπορεί να χρησιμοποιεί τις ευρυζωνικές υπηρεσίες πρόσβασης στο διαδίκτυο και εν κινήσει. Όπως δείχνουμε και στο σχήμα 6.6, η υποστηριζόμενη κινητικότητα είναι μικρότερη από το σημερινό υιοθετημένο πρότυπο για συστήματα κινητών επικοινωνιών GSM, το οποίο όμως δεν μπορεί να



Σχήμα 6.5 Λειτουργία του WiMAX



Σχήμα 6.6 Ταχύτητα Μετάδοσης και Κινητικότητα Χρήστη στο WiMAX

προσφέρει τις υψηλές ταχύτητες μετάδοσης δεδομένων του WiMAX. Μέσω αυτών των υψηλών ταχυτήτων, το WiMAX θα επιτρέψει, με καλύτερη ποιότητα απ' ότι το GSM, την εκτέλεση υπηρεσιών απαιτητικών σε εύρος ζώνης, όπως βιντεοκλήσεις. Από τα παραπάνω συμπεραίνουμε ότι το WiMAX θα έχει πρωτεύοντα ρόλο στην

ανάπτυξη συστημάτων κινητών επικοινωνιών 4ης γενιάς (4G), τα οποία έχουν ως βασικές αρχές τη δυνατότητα χρήσης ευρυζωνικών υπηρεσιών και την ενοποίηση με το διαδικτυακό πρωτόκολλο IP.

Οι υψηλοί ρυθμοί μετάδοσης συνοδευόμενοι από την απαιτούμενη αξιοπιστία του συστήματος κάνουν επιτακτική και στην περίπτωση του WiMAX τη χρήση κωδίκων διόρθωσης λαθών που πρέπει να εμφανίζουν εξαιρετικές επιδόσεις μάλιστα. Ως γνωστόν, οι LDPC κώδικες πληρούν αυτό το κριτήριο και είναι οι μόνοι κώδικες μαζί με τους turbo που αγγίζουν το θεωρητικό όριο επίδοσης του Shannon. Συνεπώς, το τελευταίο έως σήμερα πρότυπο του WiMAX, το **IEEE 802.16e**, έχει υιοθετήσει τους LDPC κώδικες. Η LDPC κωδικοποίηση στο πρότυπο 802.16e αποτελείται από έξι διαφορετικές κλάσεις LDPC κωδίκων που υποστηρίζουν τέσσερις διαφορετικούς ρυθμούς κώδικα (από 1/2 έως 5/6). Και οι έξι διαφορετικές κλάσεις έχουν την ίδια γενική δομή πίνακα ελέγχου ισοτιμίας **H** ενώ υπάρχει η δυνατότητα επιλογής από 19 συνολικά μεγέθη κωδικών λέξεων (576 έως 2304 bits). Αυτή η μεγάλη ευελιξία στο μήκος της κωδικής λέξης αποτελεί και τη μεγαλύτερη πρόκληση σε σχεδιαστικό επίπεδο. Στη δημοσίευση [11] προτείνεται μια αποδοτική σχεδίαση hardware το οποίο μπορεί να χρησιμοποιηθεί για την κωδικοποίηση και αποκωδικοποίηση όλων των παραπάνω κατηγοριών LDPC κωδίκων. Η αποκωδικοποίηση που περιγράφεται βασίζεται στη διαστρωματωμένη αρχιτεκτονική (layered architecture).

Συνοψίζοντας τη χρήση LDPC κωδίκων στα πρότυπα DVB-S2 και WiMAX IEEE 802.16e παραθέτουμε τον πίνακα 6.1 όπου φαίνονται τα βασικά χαρακτηριστικά των χρησιμοποιούμενων LDPC κωδίκων. Ξεχωρίζουμε τη μεγάλη ευελιξία στην επιλογή μεγέθους κωδικής λέξης στο WiMAX 802.16e και το πολύ μεγάλο μήκος κωδικής λέξης στο DVB-S2 το οποίο αιτιολογείται από την απαίτηση πολύ καλής επίδοσης για πολύ χαμηλές τιμές του σηματοθορυβικού λόγου, φαινόμενο συχνό στις δορυφορικές επικοινωνίες. Τέλος, να σημειώσουμε απλά ότι LDPC κώδικες έχουν υιοθετηθεί και στο πρότυπο **WiFi (IEEE 802.11n)** το οποίο μπορεί να χαρακτηριστεί εξίσου ως τηλεπικοινωνιακό πρότυπο νέας (ή ακόμα και επόμενης!) γενιάς.

Πίνακας 6.1 Παράμετροι LDPC Κωδίκων στο DVB-S2 και στο WiMAX

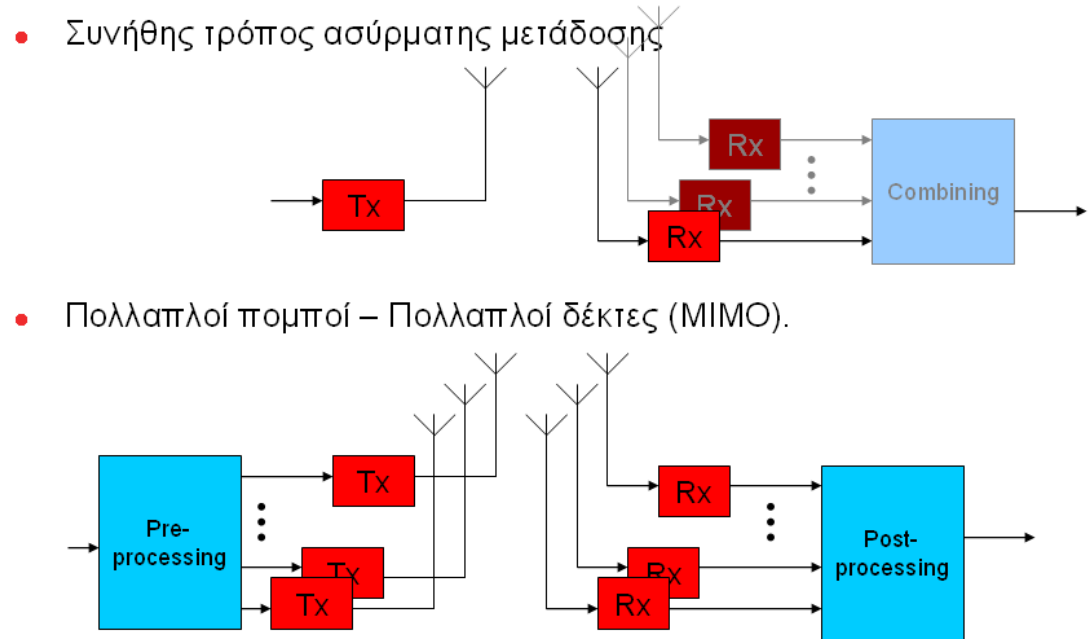
	Μήκος Κωδικής Λέξης			**Ρυθμός Κώδικα**		
	#	min	max	#	min	max
DVB-S2	2	16200	64800	11	1/4	9/10
WiMAX 802.16e	19	576	2304	4	1/2	5/6

6.2 Ερευνητικά θέματα LDPC κωδίκων

Τα ερευνητικά θέματα γύρω από τους LDPC κώδικες προκύπτουν και εγγενώς και λόγω εξωτερικών παραγόντων. Στην πρώτη κατηγορία κατατάσσουμε ζητήματα όπως κατασκευή κωδίκων με μικρό ελάχιστο μήκος κύκλου στον αντίστοιχο γράφο Tanner και στη δεύτερη σχεδίαση LDPC κωδίκων που να ικανοποιούν τις γενικές απαιτήσεις του συστήματος στο οποίο πρόκειται να χρησιμοποιηθούν. Είδαμε για παράδειγμα τη σχεδίαση κωδίκων για το DVB-S2 που προέκυψαν με πολύ μεγάλο μήκος κωδικής λέξης ακριβώς για να ικανοποιούν τις απαιτήσεις επίδοσης του DVB-S2. Είναι σαφές ότι όσο αναπτύσσονται συνεχώς νέες απαιτήσεις για τα σύγχρονα τηλεπικοινωνιακά συστήματα τόσο συνεχίζεται η έρευνα πάνω στη σχεδίαση LDPC κωδίκων, που

αποτελούν κώδικες αιχμής στις μέρες μας. Μέσα στην πληθώρα λοιπόν ερευνητικών θεμάτων, μπορούμε να σταθούμε σε κάποια που φαίνεται να απασχολούν κατά κύριο λόγο την επιστημονική κοινότητα.

Πρώτον, ερευνάται η βέλτιστη σχεδίαση LPDC κωδίκων για συστήματα πολλαπλών εισόδων – πολλαπλών εξόδων (multiple input – multiple output, **MIMO**). Η τεχνολογία MIMO είναι μία μέθοδος χρησιμοποίησης πολλαπλών κεραιών και στον πομπό και στον δέκτη στις ασύρματες επικοινωνίες (Σχ. 6.7) και κερδίζει συνεχώς έδαφος καθώς φαίνεται να καλύπτει τις σύγχρονες απαιτήσεις για μεγάλους ρυθμούς μετάδοσης πληροφοριών, βέλτιστη αξιοποίηση της χωρητικότητας του διαύλου και υψηλή ποιότητα υπηρεσιών. Σαφώς, για να ικανοποιήσει το MIMO σύστημα τις παραπάνω απαιτήσεις πρέπει να χρησιμοποιεί και κατάλληλη τεχνική κωδικοποίησης, με τους LDPC κώδικες να είναι μία από τις καλύτερες λύσεις.



Σχήμα 6.7 MIMO συστήματα

Παράλληλα, δίνεται ιδιαίτερο βάρος στη σχεδίαση LDPC κωδίκων που θα συνοδεύουν την τεχνική **HARQ**. Η τεχνική HARQ (Hybrid ARQ) ονομάζεται έτσι γιατί αποτελεί συνδυασμό των τεχνικών ARQ και FEC. Έχει αποδειχθεί μάλιστα, ότι με κατάλληλη σχεδίαση η εν λόγω τεχνική μπορεί να αυξήσει την απόδοση του συστήματος. Και σε αυτή την περίπτωση, η επιλογή του σχήματος κωδικοποίησης είναι κομβικής σημασίας, με τους LDPC κώδικες να έχουν ήδη χρησιμοποιηθεί σε HARQ συστήματα αλλά και με την έρευνα να συνεχίζεται όσον αφορά τη χρησιμοποίηση τους σε ακόμα πιο εξελιγμένα HARQ σχήματα.

Τέλος, να αναφέρουμε την προσπάθεια που επιτελείται ώστε να μειωθεί η πολυπλοκότητα των LDPC κωδίκων σε επίπεδο hardware. Για αυτό το σκοπό, συνεισφέρουν και επιστήμονες ειδικοί στην αρχιτεκτονική VLSI συστημάτων. Κάτι τέτοιο έχει ιδιαίτερη αξία σε υψηλών απαιτήσεων συστήματα, καθώς όπως είδαμε γίνεται κυρίως χρήση LDPC κωδίκων με μεγάλα μεγέθη κωδικής λέξης που οδηγούν σε αύξηση της πολυπλοκότητας.

Ένας γενικός οδηγός για τα τρέχοντα ερευνητικά θέματα των LDPC κωδίκων μπορεί να αναζητηθεί στη δημοσίευση [6].

7 Συμπεράσματα

7.1 Συμβολή της εργασίας

Βασικός στόχος της παρούσας εργασίας ήταν η μελέτη και προσομοίωση LDPC κωδίκων με σκοπό να εξαχθούν ποιοτικά αλλά και ποσοτικά συμπεράσματα για την συμπεριφορά τους. Κάτι τέτοιο έχει ιδιαίτερη αξία καθώς οι LDPC κώδικες χαρακτηρίζονται ως κώδικες αιχμής και βρίσκουν πρακτική εφαρμογή σε ολοένα και περισσότερα τηλεπικοινωνιακά συστήματα. Βέβαια, για την ανάπτυξη μιας ολοκληρωμένης προσομοίωσης δεν αρκούσε μόνο η μοντελοποίηση του υποσυστήματος κωδικοποίησης καναλιού. Έτσι, δώσαμε ιδιαίτερη έμφαση στις τεχνικές διαμόρφωσης καθώς και στη μοντελοποίηση του διαύλου. Παράλληλα με τη μελέτη των κωδίκων λοιπόν, εξάγαμε και συμπεράσματα σχετικά με τη συγκριτική μελέτη τεχνικών διαμόρφωσης (BPSK, 4-QAM, 16-QAM) καθώς και τη συγκριτική μελέτη τύπων διαύλου, με έμφαση στους διαύλους που απαντώνται στις κινητές και δορυφορικές επικοινωνίες (AWGN, Rayleigh, Rice). Συνοψίζοντας τη συμβολή της εργασίας μπορούμε να αναφερθούμε στα παρακάτω στοιχεία:

- Επισκόπηση των τεχνικών κωδικοποίησης καναλιού με έμφαση στην παρουσίαση των γραμμικών συμπαγών κωδίκων, κατηγορία στην οποία ανήκουν οι LDPC κώδικες. Ανάλυση των δυνατοτήτων των κωδίκων όσον αφορά τον εντοπισμό και τη διόρθωση λαθών και παρουσίαση τρόπων αξιολόγησης τεχνικών κωδικοποίησης.
- Θεωρητική θεμελίωση της προσομοίωσης και ανάδειξη της σημασίας της στα σύγχρονα τηλεπικοινωνιακά συστήματα. Παρουσίαση της σχεδιαστικής πορείας για την ανάπτυξη προσομοιωτή. Τεκμηρίωση του Matlab ως σημαντικού προσομοιωτικού εργαλείου.
- Μελέτη των LDPC κωδίκων και ανάλυση των αλγορίθμων αποκωδικοποίησης τους. Ανάπτυξη προσομοιωτή για την εκτίμηση της επίδοσης τους υπό διάφορες τεχνικές διαμόρφωσης και υπό διάφορους τύπους διαύλου. Σχεδίαση και προσομοίωση ενός LDPC (256, 128) κώδικα που οδήγησε στην εξαγωγή πολύ χρήσιμων συμπερασμάτων σχετικά με τις ιδιότητες του. Εκτός από τις ιδιότητες του κώδικα, τα αποτελέσματα της προσομοίωσης μας επέτρεψαν να εξάγουμε συμπεράσματα σχετικά με τις διάφορες τεχνικές διαμόρφωσης και τους διάφορους τύπους διαύλου.
- Παρουσίαση των κυριότερων τηλεπικοινωνιακών συστημάτων που βρίσκουν σήμερα εφαρμογή οι LDPC κώδικες και ανάδειξη του ρόλου τους μέσα σε αυτά τα συστήματα.

7.2 *Προτάσεις για Μελλοντική Έρευνα*

Η προσομοίωση που δημιουργήθηκε σε αυτή την εργασία μπορεί να λειτουργήσει ως πλατφόρμα για την ανάπτυξη νέων προσομοιώσεων. Μέσω της μελέτης του πηγαίου κώδικα Matlab της προσομοίωσης (στοιχεία από τον κώδικα υπάρχουν στο Παράρτημα Α) υπάρχει η δυνατότητα εκτέλεσης πολλών και διαφορετικών σεναρίων. Για παράδειγμα, μπορεί να σχεδιασθεί νέος LDPC κώδικας (ουσιαστικά να σχεδιασθεί νέος πίνακας ελέγχου ισοτιμίας) και ύστερα να μελετηθεί εύκολα η επίδοση του καθώς οι αλγόριθμοι κωδικοποίησης / αποκωδικοποίησης παραμένουν ίδιοι. Ύστερα από τη διεξοδική μελέτη μας πάνω στους LDPC κώδικες, μπορούμε να δώσουμε κάποιες βασικές κατευθύνσεις μελλοντικής έρευνας που μπορούν να υλοποιηθούν εξ' αρχής ή, γιατί όχι, με βάση αυτή την εργασία:

- Σχεδίαση και εκτίμηση της επίδοσης LDPC κώδικα. Να ληφθεί υπόψη ότι η επίδοση αυξάνεται με αυξανόμενο μήκος κωδικής λέξης. Παράλληλα, βοηθά σημαντικά η απουσία κύκλων μικρού μήκους στον αντίστοιχο γράφο Tanner.
- Σαν προέκταση αυτής της εργασίας, μπορεί να γίνει μελέτη επίδοσης LDPC κωδίκων και υπό άλλα σχήματα διαμόρφωσης και τύπων διαύλου. Απαιτείται η αλλαγή μόνο του αντίστοιχου κομματιού του πηγαίου κώδικα Matlab.
- Είναι αλήθεια ότι μέσω της λογαριθμικής αποκωδικοποίησης LDPC κωδίκων έχουμε σχετικά μικρή πολυπλοκότητα αποκωδικοποίησης (decoding) σε σύγκριση και με άλλους κώδικες ανάλογων επιδόσεων. Η χρήση όμως γεννητόρων πινάκων μεγάλων διαστάσεων (αναπόφευκτα χάριν της επίδοσης) δημιουργεί σημαντική πολυπλοκότητα κωδικοποίησης (encoding). Ενθαρρύνεται λοιπόν η αναζήτηση αλγορίθμων απευθείας κωδικοποίησης (direct encoding) χωρίς τη χρήση γεννήτορα πίνακα, όπως για παράδειγμα δείξαμε ότι γίνεται στο πρότυπο DVB-S2. Μετά από τέτοιες διαδικασίες δημιουργείται κατά κανόνα ένα ισοζύγιο (μείωση πολυπλοκότητας αλλά και μείωση επίδοσης) που χρήζει ανάλυσης για το αν τελικά είναι θετικό με βάση τις απαιτήσεις του συστήματος.
- Οι LDPC κώδικες, όπως και οι περισσότεροι κώδικες, αποδίδουν καλύτερα όταν τα λάθη που εμφανίζονται διακρίνονται από ομοιόμορφη κατανομή, καθώς έτσι έχουν σχεδιαστεί. Υπάρχουν όμως περιπτώσεις στις τηλεπικοινωνίες (π.χ. υπό έντονες διαλείψεις) όπου τα σφάλματα εμφανίζονται κατά ομάδες (burst errors) και δεν είναι ανεξάρτητα μεταξύ τους. Σε τέτοιες περιπτώσεις, απαιτείται η χρήση εξωτερικού κώδικα που μπορεί να διορθώνει τέτοιου είδους σφαλμάτων. Σε αυτή την κατηγορία ανήκουν οι Reed-Solomon (RS) κώδικες. Ενθαρρύνεται λοιπόν η σχεδίαση και εξέταση αλυσιδωτού σχήματος κωδικοποίησης με εξωτερικό κώδικα Reed-Solomon και εσωτερικό κώδικα LDPC.

Παράρτημα: Κώδικας Matlab

Στο παράρτημα της εργασίας παραθέτουμε στοιχεία από τον πηγαίο κώδικα Matlab η εκτέλεση του οποίου μας έδωσε τα αποτελέσματα της προσομοίωσης που παρουσιάζονται στο κεφάλαιο 5. Για λόγους έμφασης στις διαδικασίες κωδικοποίησης / αποκωδικοποίησης παραθέτουμε τον κώδικα που αντιστοιχεί μόνο σε διαμόρφωση **BPSK** και δίαυλο τύπου **AWGN**. Τα βασικά αρχεία της εφαρμογής είναι τα **ldpc_main.m** και **ldpc_main_log.m**, με το πρώτο να αντιστοιχεί σε μη-λογαριθμική και το δεύτερο σε λογαριθμική αποκωδικοποίηση. Παράλληλα, παρατίθενται και όλα τα **.m** αρχεία που καλούνται από τα δύο προαναφερθέντα βασικά αρχεία.
Η αλγοριθμική διαδικασία που ακολουθείται για την ανάπτυξη προσομοιωτή που μελετά την επίδοση του LDPC κώδικα δίνεται σε γενικές γραμμές παρακάτω υπό τη μορφή ψευδοκώδικα.

Ψευδοκώδικας

```
Load H % Φόρτωση του πίνακα ελέγχου ισοτιμίας
Get_G_from_H % Δημιουργία, με βάση τον H,του γεννήτορα πίνακα

SNRRange=[0:4:0.5] % Εύρος τιμών SNR, εδώ 0 έως 4 με βήμα 0.5
Packets_per_SNR=1000 % Προσομοιούμενα πακέτα ανά SNR

for SNRrange_index=1:length(SNRrange) % για κάθε τιμή του SNR

   for packetnumber=1:Packets_per_SNR % για κάθε πακέτο (κωδική λέξη)

m=random_input_message %τυχαίο δυαδικό μήνυμα πληροφορίας
c=encode_LDPC(m, G) % κωδικοποιημένο διάνυσμα
t=modulate(c, modulation)_type)  % διαδικασία διαμόρφωσης
r=after_channel(t, channel_type) % ληφθέν διάνυσμα ύστερα από τη μετάδοση
m_after_decoding=decode_LDPC(r, H) % μήνυμα πληροφορίας ύστερα από τη διαδικασία αποκωδικοποίησης
errors_for_this_SNR(packetnumber)=compare(m, m_after_decoding) % υπολογισμός λαθών για το συγκεκριμένο πακέτο (εσωτερικό for loop) και για τον τρέχοντα SNR(εξωτερικό for loop) συγκρίνοντας το αρχικό μήνυμα πληροφορίας με το μήνυμα πληροφορίας ύστερα από τη διαδικασία αποκωδικοποίησης

   end % τερματισμός εσωτερικού βρόχου
mean_error_rate(SNRrange_index)=mean(errors_for_this_SNR) %εύρεση της μέσης τιμών σφαλμάτων για την τρέχουσα τιμή SNR
end % τερματισμός εξωτερικού βρόχου

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

Στη συνέχεια, παραθέτουμε τον πηγαίο κώδικα της προσομοίωσης. Προς διευκόλυνση του αναγνώστη, επισημαίνουμε ότι η εκτέλεση του κώδικα που ακολουθεί γίνεται είτε τρέχοντας το αρχείο **ldpc_main.m** είτε το αρχείο

ldpc_main_log.m. Η εκτέλεση της παρακάτω προσομοίωσης θα οδηγήσει στη δημιουργία γραφικής παράστασης όμοιας με του σχήματος **5.2**.

Πηγαίος Κώδικας Matlab

Αρχείο ldpc_main.m

```
% LDPC Code
clear all;
%load parity check matrix H
load 128x256regular.mat H

[rowsh,colsh]=size(H);
n=colsh ;% codeword length
k=n-rowsh ; %message length

A=H(1:n-k,1:n-k);
B=H(1:n-k,n-k+1:n) ;%H=[A B]
I=eye(k); %identity matrix
%I=eye(n-k);
load inverse_of_A.mat invA %inverse of A, over the binary field


p_aux=bin_multi(invA,B);
P=p_aux.';
%get the corresponding generator matrix G
G=[P I];

%run various simulations for various SNR (or Eb/No) values

number_of_Packets_per_SNR=1000;
SNRrange=[0:0.5:4];
mean_error_rate=zeros(1, length(SNRrange));
for SNRrange_index=1:length(SNRrange)
     if (SNRrange(SNRrange_index)>3.5)
         number_of_Packets_per_SNR=5000;
     end
    error_rate=zeros(1, number_of_Packets_per_SNR);
    for packetnumber=1:number_of_Packets_per_SNR

        m=randint(1,k);  % random input message

        c=bin_multi(m,G); %codeword production

        t=bpsk(c); %transmitted vector after BPSK mod

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        SNR=SNRrange(SNRrange_index);
        No=10^(-SNR/10);
        sigma=sqrt(No/2);

        %sprintf('SNR: %.4f dB \n',SNR)
        r=awgn(t,SNR); %Gaussian noise


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%decoding
%init
max_iter=80;
count_iter=0;
f1=1./(1+exp(-2*r/sigma^2));
f0=1-f1;

fnd=find(H==1);
[ii,jj]=ind2sub([rowsh,colsh],fnd);
len=length(ii);
%Mackay

Q0=zeros(rowsh,colsh);
Q1=zeros(rowsh,colsh);
dQ=zeros(rowsh,colsh);
dR=zeros(rowsh,colsh);
R0=zeros(rowsh,colsh);
R1=zeros(rowsh,colsh);
q0=zeros(1,n);
q1=zeros(1,n);
for i=1:len
    Q0(ii(i),jj(i))=f0(jj(i));
end
for i=1:len
    Q1(ii(i),jj(i))=f1(jj(i));
end
while (count_iter<max_iter)
    dQ=Q0-Q1;
    for i=1:len
        dR(ii(i),jj(i))=produ(dQ,ii,jj,i); %HORIZONTAL STEP
    end
    for i=1:len
        R0(ii(i),jj(i))=(1/2)*(1+dR(ii(i),jj(i)));
    end
    for i=1:len
        R1(ii(i),jj(i))=(1/2)*(1-dR(ii(i),jj(i)));
    end
    % VERTICAL
    for i=1:len
        Q0(ii(i),jj(i))=f0(jj(i))*vert_prod(R0,ii,jj,i);
    end
    for i=1:len
        Q1(ii(i),jj(i))=f1(jj(i))*vert_prod(R1,ii,jj,i);
    end
    %VERTICAL_END

    for i=1:len

[Q0(ii(i),jj(i)),Q1(ii(i),jj(i))]=normalize(Q0(ii(i),jj(i)),Q1(ii(i),
jj(i)));
    end

    % a posteriori symbol probabilities q0,q1
    for i=1:n
        q0(i)= f0(i)*column_product(R0,ii,jj,i);
    end
    for i=1:n
        q1(i)= f1(i)*column_product(R1,ii,jj,i);
        [q0(i),q1(i)]=normalize(q0(i),q1(i));
    end
    vector_est=estimate(q0,q1,n);
```

```
            vector_est;
            if bin_multi(vector_est,H')==0
                break;
            end;
            count_iter=count_iter+1;


        end
        disp('iterations: ');disp(count_iter);
        vector_est;
        error_counter=0;

        final_msg=vector_est(n-k+1:n);
        for p=1:k
            if final_msg(p)~=m(p)
                 disp('error in position ');disp(p);
                error_counter=error_counter+1;
            end

        end

        %disp('iterations: ');disp(count_iter);
        sprintf('Errors: %d %d dB \n',error_counter,
SNRrange(SNRrange_index))
        error_rate(1, packetnumber)=error_counter/k;
    end
    sprintf('For SNR= %d \n mean error rate is ',
SNRrange(SNRrange_index))
    mean_error_rate(1, SNRrange_index)=double(mean(error_rate));
    mean_error_rate(1, SNRrange_index);

    disp('new SNR');

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

Αρχείο ldpc_main_log.m

```
 % LDPC Code with logarithmic decoding

clear all;
    %load parity check matrix H
    load 128x256regular.mat H


   [rowsh,colsh]=size(H);
   n=colsh ;% codeword length
   k=n-rowsh ; %message length

   A=H(1:n-k,1:n-k);
   B=H(1:n-k,n-k+1:n) ;%H=[A B]
   I=eye(k); %identity matrix
   %I=eye(n-k);
   load inverse_of_A.mat invA %inverse of A, over the binary field
   %%%%%test
```

```
    p_aux=bin_multi(invA,B);
    P=p_aux.';
    %get the corresponding generator matrix G
    G=[P I];

    %run various simulations for various SNR (or Eb/No) values
number_of_Packets_per_SNR=1000;
SNRrange=[0:0.5:4];
mean_error_rate=zeros(1, length(SNRrange));
for SNRrange_index=1:length(SNRrange)
    if (SNRrange(SNRrange_index)>2.5)
        number_of_Packets_per_SNR=5000;
    end
    error_rate=zeros(1, number_of_Packets_per_SNR);
    for packetnumber=1:number_of_Packets_per_SNR

    m=randint(1,k);
    %m=[1 0 0 0];
    c=bin_multi(m,G); %codeword production
    disp('codeword production');
    t=bpsk (c); %transmitted vector after BPSK mod


        SNR=SNRrange(SNRrange_index);
        No=10^(-SNR/10);
        sigma=sqrt(No/2);

      r=awgn(t,SNR); % AWGN channel


    % Logarithmic LDPC Decoder

    max_iter=80;

    count_iter=0;
    f1=1./(1+exp(-2*r/sigma^2));
    f0=1-f1;
    for i=1:length(f1)
        logf1(i)=abs(log(f1(i)));
    end
    for i=1:length(f0)
        logf0(i)=abs(log(f0(i)));
    end


    fnd=find(H==1);
    [ii,jj]=ind2sub([rowsh,colsh],fnd);
    len=length(ii);
    %Mackay in logarithmic form
    s=zeros(rowsh,colsh);
    sdR=zeros(rowsh,colsh);

    logQ0=zeros(rowsh,colsh); %LOG
    Q1=zeros(rowsh,colsh);
    logQ1=zeros(rowsh,colsh);  %log

    LdQ=zeros(rowsh,colsh); % log

    LdR=zeros(rowsh,colsh); % log
    LR0=zeros(rowsh,colsh); %log
    LR1=zeros(rowsh,colsh); %log
```

```
    Lc0=zeros(rowsh,colsh); %log
    Lc1=zeros(rowsh,colsh); %log
    Lq0=zeros(1,n); %log
    Lc_zero=zeros(1,n); %log
    Lq1=zeros(1,n); %log
    Lc_one=zeros(1,n); %log

    for i=1:len
        Q0(ii(i),jj(i))=f0(jj(i));
        logQ0(ii(i),jj(i))=abs(log(Q0(ii(i),jj(i))));
    end
    for i=1:len
        Q1(ii(i),jj(i))=f1(jj(i));
        logQ1(ii(i),jj(i))=abs(log(Q1(ii(i),jj(i))));
    end
    while (count_iter<max_iter)

    for i=1:len
        if logQ0(ii(i),jj(i))<= logQ1(ii(i),jj(i))
            s(ii(i),jj(i))=0;
        elseif logQ0(ii(i),jj(i))> logQ1(ii(i),jj(i))
           s(ii(i),jj(i))=1;
        end
    end

    for i=1:len
    LdQ(ii(i),jj(i))
=min(logQ0(ii(i),jj(i)),logQ1(ii(i),jj(i)))+f_minus(abs(logQ0(ii(i),j
j(i))-logQ1(ii(i),jj(i))));
    end

    for i=1:len
        LdR(ii(i),jj(i))=sum_log(LdQ,ii,jj,i); %HORIZONTAL STEP
        sdR(ii(i),jj(i))=sum_log(s,ii,jj,i);
    end
    for i=1:len

        if mod(sdR(ii(i),jj(i)),2)==0
        LR0(ii(i),jj(i))=log(2)-f_plus(LdR(ii(i),jj(i)));
        elseif mod(sdR(ii(i),jj(i)),2)==1
            LR0(ii(i),jj(i))=log(2)+f_minus(LdR(ii(i),jj(i)));
        end
    end
    %%%%
    for i=1:len
        if mod(sdR(ii(i),jj(i)),2)==0
        LR1(ii(i),jj(i))=log(2)+f_minus(LdR(ii(i),jj(i)));
        elseif mod(sdR(ii(i),jj(i)),2)==1
            LR1(ii(i),jj(i))=log(2)-f_plus(LdR(ii(i),jj(i)));
        end
    end
    % VERTICAL
    length(logf0);
    for i=1:len
        Lc0(ii(i),jj(i))=logf0(jj(i))+vertical_sum_log(LR0,ii,jj,i);
    end
    for i=1:len
        Lc1(ii(i),jj(i))=logf1(jj(i))+vertical_sum_log(LR1,ii,jj,i);
    end

    for i=1:len
```

```
        logQ0(ii(i),jj(i))=Lc0(ii(i),jj(i))-min(Lc0(ii(i),jj(i)),
Lc1(ii(i),jj(i)))+f_plus(abs(Lc0(ii(i),jj(i))- Lc1(ii(i),jj(i))));
    end
    for i=1:len
        logQ1(ii(i),jj(i))=Lc1(ii(i),jj(i))-min(Lc0(ii(i),jj(i)),
Lc1(ii(i),jj(i)))+f_plus(abs(Lc0(ii(i),jj(i))- Lc1(ii(i),jj(i))));
    end


    % a posteriori symbol probabilities Lq0, Lq1
    for j=1:n

        x=find(jj==j);
        i=ii(x(1));
        Lc_zero(j)= Lc0(i, j)+LR0(i, j);
        Lc_one(j)= Lc1(i, j)+LR1(i, j);
        Lq0(j)=Lc_zero(j)-min(Lc_zero(j),
Lc_one(j))+f_plus(abs(Lc_zero(j)-Lc_one(j)));
        Lq1(j)=Lc_one(j)-min(Lc_zero(j),
Lc_one(j))+f_plus(abs(Lc_zero(j)-Lc_one(j)));
    end

     vector_est=estimate_log(Lq0,Lq1,n);
     if bin_multi(vector_est,H')==0
         break;
     end;
     count_iter=count_iter+1;


    end

    error_counter=0;

    final_msg=vector_est(n-k+1:n);
    for p=1:k
        if final_msg(p)~=m(p)
            disp('error in position ');disp(p);
            error_counter=error_counter+1;
        end
    end

         sprintf('Errors: %d %d dB \n',error_counter,
SNRrange(SNRrange_index))
         error_rate(1, packetnumber)=error_counter/k;
     end
     sprintf('For SNR= %d \n mean error rate is ',
SNRrange(SNRrange_index))
     mean_error_rate(1, SNRrange_index)=double(mean(error_rate));
     mean_error_rate(1, SNRrange_index);

     disp('new SNR');

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

Αρχείο bin_multi.m

```
function res=bin_multi(A,B)
    % sum=0;
    [ra,ca]=size(A); %get A dimensions
    [rb,cb]=size(B); %get B dimensions
  res=zeros(ra,cb);
    if (ca==rb) ,
  for i=1:ra
      for k=1:cb
          for j=1:ca
              res(i,k)=xor(res(i,k),A(i,j)*B(j,k));
          end;
      end;
  end;

    elseif (ca~=rb)
        disp('error in matrix multiplication');
    else disp('...');
    end;
```

Αρχείο column_product.m

```
% a posteriori symbol probs qX(j)=fX(j) * Π RX(i,j) , i e M(j) ,
X={0,1}

function cp=column_product(R,ii,jj,n)

cp=1;
for j=1:length(ii)
    if jj(j)==n %same column

            cp=cp*R(ii(j),jj(j));

     end;
end;
```

Αρχείο estimate.m

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

function d=estimate(q0,q1,n)

for j=1:n
    if q0(j)>q1(j)
        d(j)=0;
    else
        d(j)=1;
    end;
end;
```

Αρχείο estimate_log.m

```
% final vector estimation, in current iteration
% if q0(j)>q1(j) ->  d(j)=0, else d(j)=1
% so, logaritmically : if Lq0(j)<Lq1(j) ->  d(j)=0, else d(j)=1

function d=estimate_log(Lq0,Lq1,n)
```

```
for j=1:n
    if Lq0(j)<Lq1(j)
        d(j)=0;
    else
        d(j)=1;
    end;
end;
```

Αρχείο f_minus.m

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

Αρχείο f_minus.m

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

Αρχείο myXor.m

```
%modulo-2 addition
function mxr=myXor (a , b)
if ((a==0) && (b==0)), mxr=0;

elseif ((a==0) && (b==1)), mxr=1;

elseif ((a==1) && (b==0)), mxr=1;

elseif ((a==1) && (b==1)), mxr=0;
 else disp('error...');
end;
```

Αρχείο normalize.m

```
% normalize Q0,Q1 so that Q0(i,j)+Q1(i,j)=1

function [a,b]=normalize(inp1,inp2)

aux=1./(eps+(inp1+inp2));

a=inp1*aux;
b=inp2*aux;
```

Αρχείο bpsk.m

```
%BPSK modulation / mapping
function tr=bpsk(c)
    for i=1:length(c)
```

```
        if c(i)==0
            tr(i)=-1;
        else
            tr(i)=c(i);
        end
    end
```

Αρχείο produ.m

```
% product calculation, sum-product algorithm
%δR(i,j)=Π δQ(i,j') , j' is_member N(i)\j
%HORIZONTAL STEP
function product=produ(dQ,ii,jj,i)

product=1;
for j=1:length(ii)
    if ii(j)==ii(i) %same row
        if jj(j)~=jj(i) % \j

            product=product*dQ(ii(j),jj(j));

        end;
    end;
end;
```

Αρχείο sum_log.m

```
% product calculation, sum-product algorithm
%δR(i,j)=Π δQ(i,j') , j' is_member N(i)\j
%HORIZONTAL STEP, logarithmic decoding (product -> sum)
function sum=sum_log(LdQ,ii,jj,i)

sum=0;
for j=1:length(ii)
    if ii(j)==ii(i) %same row
        if jj(j)~=jj(i) % \j

            sum=sum+LdQ(ii(j),jj(j));

        end;
    end;
end;
```

Αρχείο vert_prod.m

```
% VERTICAL STEP
%SUM-PRODUCT ALGORITHM, MACKAY'S SIMPLIFICATION
%Q0(i,j)=f0(j)* Π R0(i',j) , i' is_member M(j)\i
%same with Q1

function vp=vert_prod(R,ii,jj,i)

vp=1;
for j=1:length(ii)
    if jj(j)==jj(i) %same column
```

```
        if ii(j)~=ii(i) % \i

            vp=vp*R(ii(j),jj(j));

        end;
    end;
end;
```

Αρχείο vertical_sum.m

```
% VERTICAL STEP
%SUM-PRODUCT ALGORITHM, MACKAY'S SIMPLIFICATION in  logarithmic form
%LQ0(i,j)=Lf0(j)+ Σ LR0(i',j) , i' is_member M(j)\i
%same with LQ1

function vs=vertical_sum_log(R,ii,jj,i)

vs=0;
for j=1:length(ii)
    if jj(j)==jj(i) %same column
        if ii(j)~=ii(i) % \i

            vs=vs+R(ii(j),jj(j));

        end;
    end;
end;
```

Βιβλιογραφία - Αναφορές

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

[2] B. Sklar, "Digital Communications, Fundamentals and Applications", Prentice-Hall, 1988.

[3] J. Moreira, P. Farell, "Essentials of Error Control Coding", Wiley, 2006.

[4] Wiliam H. Tranter, K. Sam Shanmugan, Theodore S. Rappaport, Kurt L. Kosbar, "Principles of Communications Systems Simulation with Wireless Applications", Prentice Hall Communications Engineering and Emerging Technologies Series, 2004.

[5] Χ. Καψάλη, Π. Κωττή, "Δορυφορικές Επικοινωνίες", Εκδόσεις Τζιόλα, 2006.

[6] T. Ohtsuki, "LDPC Codes in Communications and Broadcasting", IEICE TRANS. COMMUN., VOL.E-90B, NO.3 MARCH 2007.

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

[8] 3GPP, "Comparison of structured LDPC codes and 3GPP turbocodes," 3GPP TSG RAN WG1 #43, R1-051360, Nov. 2005.

[9] A. Morello and V. Mignone, "DVB-S2: The second generation standard for satellite broad-band services," Proc. IEEE, vol.94, pp.210–227, Jan. 2006.

[10] Tang, H., Xu, J., Kou, Y., Lin, S. and Abdel-Ghaffar, K., "On algebraic construction of Gallager and circulant low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1269–1279, June 2004.

[11] T. Brack, M. Alles, T. Lehnigk-Emden, F. Kienle, N. When, N.E. L'Insalata, F. Rossi, M. Rovini, L. Fanucci, "Low Complexity LDPC Code Decoders for Next Generation Standards".

[12] www.mathworks.com