Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

3/1/2023

Εργασία

Μελέτη απόδοσης Hamming κώδικα για το δυαδικό συμμετρικό κανάλι και LDPC κωδίκων για το δυαδικό κανάλι με διαγραφές

Μάθημα: Κώδικες Διόρθωσης Σφαλμάτων
Καθηγητής: Τέγος Σωτήριος

Χρηστόφ Πέτρος
ΑΕΜ: 9928
E-mail: pcchristof@ece.auth.gr

Ζαχαριουδάκη Δανάη
ΑΕΜ: 9418
E-mail: zachardd@ece.auth.gr

Γρηγορίου Στέργιος
ΑΕΜ: 9564
E-mail: grigster@ece.auth.gr

Χειμερινό Εξάμηνο
2023-2024

Περιεχόμενα

Παρουσίαση Προβλήματος

1. Για το δυαδικό συμμετρικό κανάλι:

(a) Για ένα γραμμικό κώδικα της επιλογή σας και λαμβάνοντας υπόψη μια τιμή για τη διάρκεια συμβόλου, να μελετηθεί η επίδραση του μήκους της κωδικής λέξης και της τάξης διαμόρφωσης στην πιθανότητα σφάλματος μπλοκ και τον ρυθμό μετάδοσης (bits/sec) που μπορεί να επιτευχθεί, συναρτήσει του λόγου σήματος προς θόρυβο (signal-to-noise ratio, SNR).

(b) Έστω ότι είναι διαθέσιμα δύο κανάλια με διαφορετική πιθανότητα λάθους. Να εξετάσετε αν είναι προτιμότερη η χρήση της ίδιας ή διαφορετικής κωδικολέξης σε κάθε κανάλι για τη μεταφορά πληροφορίας, χρησιμοποιώντας συγχρόνως και τα δύο κανάλια.

2. Για το δυαδικό κανάλι με διαγραφές:

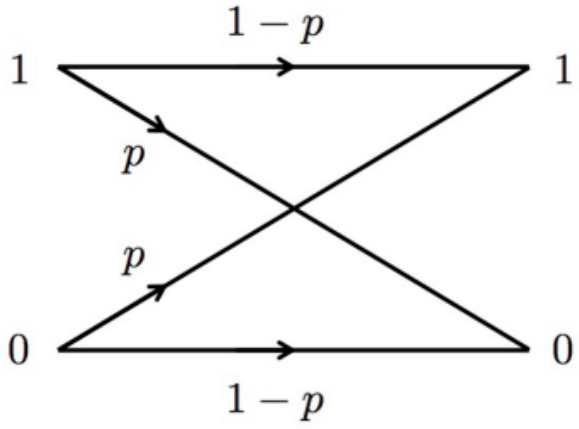
Να γίνει η επιλογή των παραμέτρων σχεδιασμού ενός irregular LDPC κώδικα, υλοποίηση αποκωδικοποιητή και μελέτη της βελτίωσης που προσφέρει συγκριτικά με ένα regular LDPC κώδικα με ίδιο κωδικό ρυθμό και μέγεθος κωδικολέξης.

Κωδικοποίηση στο δυαδικό συμμετρικό κανάλι

Στο παρόν κεφάλαιο εξετάζουμε το πρόβλημα κωδικοποίησης για μετάδοση πληροφορίας μέσω του δυαδικού συμμετρικού καναλιού με χρήση γραμμικών μπλοκ κωδίκων. Στην αρχή παρουσιάζεται με σύντομη θεωρητική επισκόπηση και στην συνέχεια γίνεται πρακτική υλοποίηση του προβλήματος και μελέτη επίδοσης κώδικα Hamming με χρήση λογισμικού, κώδικα Matlab.

Στοιχεία από την θεωρία πληροφορίας

Το δυαδικό συμμετρικό κανάλι (Binary Symmetric Channel ή συνοπτικά BSC) είναι ένα θεωρητικό κανάλι της θεωρίας πληροφορίας και ορίζεται όπως παρακάτω.



Η παράμετρος που το χαρακτηρίζει είναι η πιθανότητα αλλαγής συμβόλου p (crossover probability), και λόγο συμμετρίας οι δεσμευμένες πιθανότητες $P(Y|X)$, όπου X το σήμα που στέλνει ο πομπός και Y το σήμα που λαμβάνει ο δέκτης, είναι $P(Y=0|X=0) = P(Y=1|X=1) = 1-p$ και $P(Y=1|X=0) = P(Y=0|X=1) = p$. Η χωρητικότητα κατά Shannon του BSC είναι ίση με $C_{BSC} = 1 - H_b(p)$ όπου $H_b(x) = -x\log_2 x - (1-x)\log_2(1-x)$ είναι η δυαδική συνάρτηση εντροπίας.

Συνοπτική θεωρία γραμμικών μπλοκ κωδίκων

Παρακάτω παραθέτουμε μερικές χρήσιμες έννοιες που θα χρησιμοποιηθούν κατά την διάρκεια αυτής της εργασίας. Τονίζεται ότι όλες οι έννοιες αναφέρονται σε δυαδικούς κώδικες καθώς η περαιτέρω γενίκευση τους μπορεί να διαφέρει από τον τρόπο που θα τους ορίσουμε.

- Μπλοκ κώδικας: Ένας δυαδικός μπλοκ κώδικας ορίζεται πλήρως από $M = 2^k$ δυαδικές ακολουθίες μήκους n που καλούνται κωδικές λέξεις. Ένας κώδικας C αποτελείται από M κωδικές λέξεις $\boldsymbol{c_i}$ όπου $\boldsymbol{c_i}$ είναι μια ακολουθία μήκους n με συνιστώσες 0 ή 1.
- Γραμμικός μπλοκ κώδικας: Ένας μπλοκ κώδικας είναι γραμμικός αν οποιοσδήποτε συνδυασμός δύο κωδικών λέξεων του είναι επίσης κωδική λέξη. Ένας γραμμικός μπλοκ κώδικας ο οποίος μετατρέπει μια λέξη πληροφορίας μήκους k bits σε μια κωδική λέξη μήκους n bits συμβολίζεται με C(n,k).
- Ρυθμός του κώδικα: $r = \frac{k}{n} = \frac{\log_2 M}{n}$
- Δυικός κώδικας: Ο δυικός κώδικας ενός γραμμικού κώδικα είναι ένας άλλος γραμμικός κώδικας που ορίζεται με τη λήψη του ορθογώνιου συμπληρώματος του αρχικού κώδικα. Με άλλα λόγια, ο δυικός κώδικας ενός κώδικα C είναι το σύνολο όλων των διανυσμάτων που είναι ορθογώνια σε κάθε διάνυσμα του C.
- Συστηματικός κώδικας: Σε έναν συστηματικό κώδικα, μια κωδική λέξη ως αναπαράσταση μιας λέξης πληροφορίας έχεις ως πρόθεμα τα ίδια ακριβώς bits με την λέξη πληροφορίας και στην συνέχεια ως επίθεμα προστίθενται τα parity check bits (επιπρόσθετα bits που αποσκοπούν στην ανίχνευση και διόρθωση σφαλμάτων).
- Απόσταση Hamming δύο κωδικολέξεων: Ο αριθμός των bits στα οποία διαφέρουν οι δύο αυτές κωδικολέξεις. Συμβολίζεται με $d(\boldsymbol{c_i}, \boldsymbol{c_j})$.
- Ελάχιστη απόσταση Hamming: Η ελάχιστη απόσταση Hamming μεταξύ δύο οποιονδήποτε διαφορετικών κωδικολέξεων. Συμβολίζεται με $d_{min} = \min_{\substack{\boldsymbol{c_i},\boldsymbol{c_j} \\ \boldsymbol{i \neq j}}} d(\boldsymbol{c_i}, \boldsymbol{c_j})$.

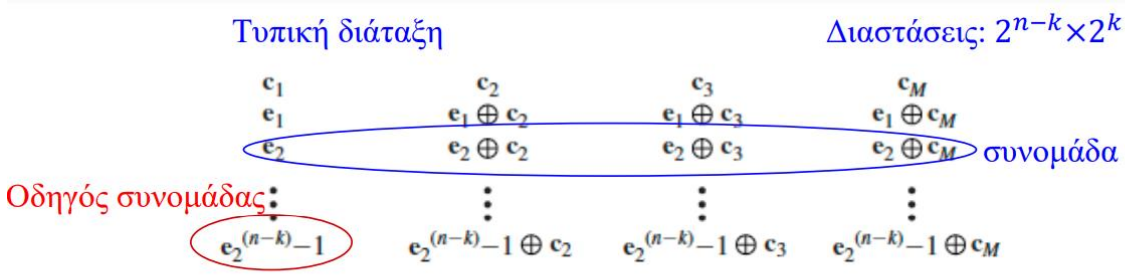
Η ελάχιστη απόσταση Hamming είναι ένα από τα σημαντικότερα χαρακτηριστικά ενός κώδικα καθώς συνδέεται άμεσα με την ικανότητα ανίχνευσης ή/και διόρθωσης σφαλμάτων του κώδικα αυτού. Συγκεκριμένα, ένας κώδικα με ελάχιστη απόσταση Hamming d μπορεί να ανιχνεύσει το πολύ $d-1$ και να διορθώσει το πολύ $\left\lfloor {(d-1)}/{2} \right\rfloor$ σφάλματα. Η ποσότητα αυτή ονομάζεται επίσης packing radius ή error correction capability. Είναι συνεπώς εύλογο κατά την κατασκευή ενός κώδικα να στοχεύουμε στην μεγιστοποίηση της ελάχιστης απόστασης Hamming ώστε να έχουμε υψηλό error correction capability. Πόσο υψηλή μπορεί να είναι όμως η απόσταση Hamming ενός (n,k) γραμμικού μπλοκ κώδικα; Ένα άνω όριο μπορεί να υπολογισθεί μέσω του Singleton Bound το οποίο για την περίπτωση μπλοκ κωδίκων καταλήγει στον περιορισμό $d_{min} \leq n-k+1$.

Για δύο ή περισσότερους διαφορετικούς μπλοκ κώδικες με δεδομένη ελάχιστη απόσταση Hamming, ο ρυθμός κώδικα είναι το μέτρο σύγκρισης αυτών (θέλουμε να είναι όσο το δυνατόν υψηλότερος γίνεται).

Ένα άλλο πολύ σημαντικό μέγεθος που αποτυπώνει το πόσο 'οικονομικός' είναι ένας κώδικας είναι το κέρδος του κώδικα (coding gain). Το κέρδος κώδικα είναι το ποσό που μπορεί να μειωθεί η ενέργεια bit ή το SNR (σε σύγκριση με uncoded μετάδοση πληροφορίας) σύμφωνα με την τεχνική κωδικοποίησης για μια δεδομένη πιθανότητα σφάλματος bit ή μπλοκ.

Σε αυτήν την παράγραφο θα ορίσουμε συνοπτικά τις έννοιες του γεννήτορα πίνακα (generator matrix) και του πίνακα ελέγχου ισοτιμίας (parity check matrix) οι οποίοι χαρακτηρίζουν πλήρως έναν γραμμικό μπλοκ κώδικα (n,k). Ο γεννήτορας πίνακας είναι πίνακας διαστάσεων $k \times n$ και ορίζεται ως $\boldsymbol{G} = \begin{bmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{k1} & \cdots & g_{kn} \end{bmatrix}$. Έχοντας των γεννήτορα πίνακα $\boldsymbol{G}$, για κάθε λέξη πληροφορίας $\boldsymbol{x}$ η κωδική λέξη μπορεί να δοθεί από $\boldsymbol{c} = \boldsymbol{xG}$ (modulo2 πολλαπλασιασμός). Ο πίνακας ελέγχου ισοτιμίας είναι ο γεννήτορας πίνακας του δυϊκού κώδικα του αρχικού. Είναι διαστάσεων $(n-k) \times n$ και συμβολίζεται με $\boldsymbol{H}$. Ισχύει πάντα ότι $\boldsymbol{HG^T} = \boldsymbol{0}$ και αν $\boldsymbol{c}$ είναι κωδική λέξη τότε $\boldsymbol{cH^T} = \boldsymbol{0}$. Από την τελευταία ισότητα μπορούμε να καταλάβουμε ότι ο Parity Check Matrix $\boldsymbol{H}$ ενός κώδικα C εκφράζεται γεωμετρικά ως ένας πίνακας του οποίου ο μηδενοχώρος (null space) περιέχει εξ' ολοκλήρου τον κώδικα C. Μια άλλη εξαιρετικά ενδιαφέρουσα ιδιότητα που συνδέει τον πίνακα $\boldsymbol{H}$ ενός κώδικα C και την ελάχιστη απόσταση Hamming (η απλά απόσταση d) αυτού είναι η εξής: Η απόσταση d ενός γραμμικού κώδικα C ισούται με τον ελάχιστο αριθμό γραμμικά εξαρτημένων στηλών του πίνακα ελέγχου $\boldsymbol{H}$. Αυτό σημαίνει ότι υπάρχουν ακριβώς $d-1$ γραμμικά ανεξάρτητες στήλες στον πίνακα $\boldsymbol{H}$ ενός κώδικα με απόσταση d. Αυτές η ιδιότητες είναι πολύ σημαντικές στην αποκωδικοποίηση γραμμικών κωδίκων βάση συνδρόμου, όπως θα δούμε παρακάτω. Τέλος, αναφέρουμε ότι για συστηματικούς κώδικες (με τους οποίους επιλέξαμε να εργαστούμε) είναι πολύ εύκολη η μετάβαση από τον Generator Matrix στον Parity Check Matrix. Ειδικά για δυαδικούς κώδικες ισχύει η παρακάτω αντιστοιχία: $\boldsymbol{G} = [\boldsymbol{I_k}|\boldsymbol{P}] \rightarrow \boldsymbol{H} = [\boldsymbol{P^T}|\boldsymbol{I_{n-k}}]$. Από την γραμμική άλγεβρα γνωρίζουμε ότι ο μηδενοχώρος ενός πίνακα μπορεί να βρεθεί ευκολότερα με την χρήση απαλοιφής Gauss. Συνεπώς ένας κώδικας C μπορεί να περιγραφεί από περισσότερους από έναν πίνακες $\boldsymbol{H}$ ενώ μια μη συστηματική μορφή ενός πίνακα $\boldsymbol{H}$ μπορεί να οδηγήσει σε συστηματικής μορφής πίνακα $\boldsymbol{G}$ εύκολα με χρήση απαλοιφής Gauss.

Τέλος θα περιγράψουμε την αποκωδικοποίηση που θα χρησιμοποιηθεί σε παρακάτω κεφάλαια, την αποκωδικοποίηση βάσει συνδρόμου. Αυτή πραγματοποιείται με χρήση της τυπικής διάταξης του κώδικα. Παρακάτω, δείχνουμε σε μορφή εικόνας τον τρόπο κατασκευής της τυπικής διάταξης αλλά και τον μηχανισμό υλοποίησης της αποκωδικοποίησης βάση συνδρόμου.



- Στην πρώτη σειρά μπαίνουν οι κωδικές λέξεις, ξεκινώντας με την μηδενική ακολουθία $\mathbf{c}_1$.
- Για την επιλογή της $\mathbf{e}_i$, διαλέγουμε μία ακολουθία με ελάχιστο βάρος από τις ακολουθίες που δεν έχουν χρησιμοποιηθεί στις προηγούμενες i γραμμές.
- Όλα τα στοιχεία της τυπικής διάταξης είναι διαφορετικά μεταξύ τους.
- Αποκωδικοποίηση (λαμβανόμενο $\mathbf{r}$ → δυαδικό λαμβανόμενο $\mathbf{y}$)
  - ✓ Υπολογισμός του συνδρόμου $\mathbf{s}$ (δυαδική ακολουθία μήκους $n-k$, μοναδικό για κάθε συνομάδα) $\mathbf{s}=\mathbf{y}\mathbf{H}^t$.
  - ✓ Σύγκριση με $\mathbf{s}=\mathbf{e}_i\mathbf{H}^t$ για εύρεση της συνομάδας και του οδηγού της συνομάδας που αντιστοιχεί το $\mathbf{s}$.
  - ✓ Αποκωδικοποίηση ως (το $\mathbf{e}$ είναι ο οδηγός της συνομάδας που βρέθηκε)

$$\mathbf{c}=\mathbf{y}\oplus\mathbf{e}$$

Στην συνέχεια θα εξηγήσουμε συνοπτικά τους γραμμικούς μπλοκ κώδικες που επιλέξαμε πριν περάσουμε στο κομμάτι της πρακτικής υλοποίησης της εργασίας.

Κώδικες Hamming

Οι κώδικες Hamming είναι μια οικογένεια γραμμικών μπλοκ κωδίκων με $n=2^m-1$, $k=2^m-m-1$ και $d_{min}=3$ (m είναι η τάξη του κώδικα Hamming). Από την σταθερή τιμή της d_{min} καταλαβαίνουμε ότι ένας κώδικας Hamming έχει την δυνατότητα διόρθωσης $k=\frac{3-1}{2}=1$ άρα ένα και μόνο λάθος. Άμα έχουμε αλλαγή σε 2 ή περισσότερα bits κατά την μετάδοση της λέξης πληροφορίας, ο κώδικας Hamming αδυνατεί να διορθώσει τα λάθη αυτά. Ο κωδικός ρυθμός ενός κώδικα Hamming είναι ίσος με $r=\frac{2^m-m-1}{2^m-1}$ όπου $m\geq 2, m\in\mathbb{Z}$.

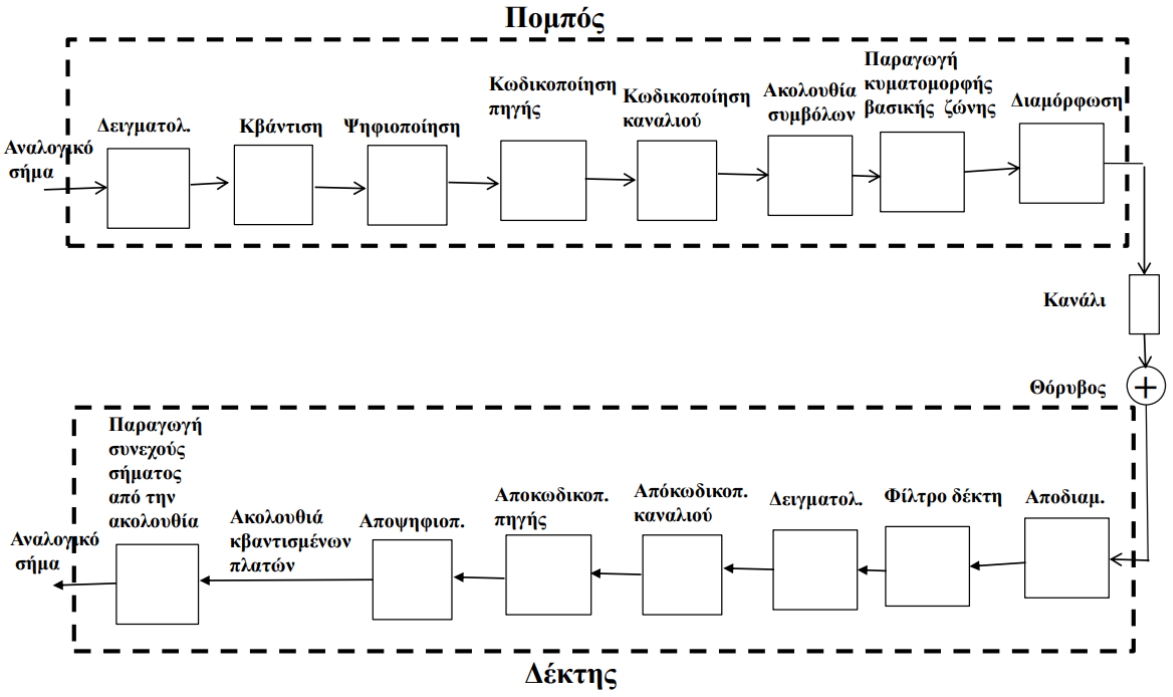
Στα πλαίσια της εργασίας επιλέξαμε να εργαστούμε με κώδικες Hamming. Κριτήριο της επιλογής αποτέλεσε τόσο η ιστορική σημασία αυτής της οικογένειας κωδίκων ως οι πρώτοι κώδικες διόρθωσης σφαλμάτων (error correcting codes) που εφευρέθηκαν, όσο και η πρακτική τους χρήση μέχρι και σήμερα. Όπως αναφέραμε προηγουμένως, στην βιβλιογραφία οι κώδικες Hamming αναφέρονται ως κώδικες διόρθωσης μοναδικού λάθους (single error correction codes) και συνεπώς μπορούν να ανιχνεύσουν και να διορθώσουν σφάλματα μόνο όταν ο ρυθμός σφάλματος είναι χαμηλός. Ένα παράδειγμα όπου συμβαίνει αυτό είναι στη μνήμη υπολογιστών (συνήθως RAM), όπου τα σφάλματα bit είναι εξαιρετικά σπάνια και οι κώδικες Hamming χρησιμοποιούνται ευρέως. Πολύ μεγάλο ενδιαφέρον παρουσιάζουν επίσης και οι δυϊκοί των κωδίκων Hamming γνωστοί ως simplex κώδικες οι οποίοι χρησιμοποιούνται κατά κόρον στον γραμμικό προγραμματισμό.

Τονίζεται ότι οι κώδικες Hamming εξ ορισμού δεν είναι συστηματικοί κώδικες και οι θέσεις στις οποίες προσθέτονται τα parity check bits του κώδικα είναι καθορισμένες. Είναι όμως εφικτή και η συστηματική υλοποίηση τους, στην οποία προβήκαμε στα πλαίσια της εργασίας για λόγους απλότητας.

Η κωδικοποίηση με χρήση κώδικα Hamming είναι πολύ εύκολη. Αρκεί κανείς να διαλέξει την τάξη του κώδικα Hamming (m) και προκύπτει αμέσως το ζευγάρι (n, k) του κώδικα. Η απόσταση του κώδικα όπως αναφέρθηκε παραπάνω είναι πάντα σταθερή και ίση με 3. Αυτό σημαίνει ότι οποιοσδήποτε γραμμικός συνδυασμός 2 ή παραπάνω στηλών του πίνακα $\boldsymbol{H}$ είναι επίσης στήλη του $\boldsymbol{H}$. Ως εκ τούτου, αρκεί να κανείς να πάρει τις όλους τους γραμμικούς συνδυασμούς των ορθοκανονικών βάσεων του διανυσματικού χώρου διάστασης $n-k$ πάνω στο δυαδικό πεδίο Galois $GF(2)$ και τους τοποθετήσει στον πίνακα $\boldsymbol{H}$, μαζί με τις βάσεις. Ύστερα, εύκολα μπορεί να προκύψει η συστηματική μορφή του πίνακα $\boldsymbol{G}$ με όσα έχουν ήδη αναφερθεί. Εξίσου εύκολη είναι και η αποκωδικοποίηση βάση συνδρόμου. Αρκεί κανείς να παρατηρήσει ότι τα σύνδρομα ενός κώδικα Hamming είναι οι στήλες του πίνακα $\boldsymbol{H}$! Για να εξάγουμε μια στήλη του πίνακα $\boldsymbol{H_{n-k,n}}$ σαν αποτέλεσμα modulo2 πολλαπλασιασμού ενός διανύσματος $\boldsymbol{c_{1,n}}$ με τον ανάστροφο πίνακα, το διάνυσμα $\boldsymbol{c}$ θα πρέπει να είναι της μορφής $\boldsymbol{c_{1,i}} = 1, \boldsymbol{c_{1,j}} = 0 \; \forall \; j \neq i$. Κατά συνέπεια οι οδηγοί της κάθε συνομάδας στην τυπική διάταξη είναι όλες οι δυαδικές ακολουθίες με μοναδιαίο βάρος διάστασης n και η μηδενική ακολουθία που μπαίνει ως ο πρώτος οδηγός. Αυτό είναι απολύτως λογικό και αναμενόμενο αν θυμηθεί κανείς ότι οι κώδικες Hamming έχουν την δυνατότητα διόρθωσης ενός μόνο λάθους. Τα σύνδρομα κάθε συνομάδας με οδηγό την ακολουθία $\boldsymbol{e_i}$ (μια δυαδική ακολουθία διαστάσεων n με μοναδικό μη μηδενικό στοιχείο στην θέση i) προκύπτουν εξάγοντας την στήλη i του πίνακα $\boldsymbol{H}$. Έτσι η τυπική διάταξη κατασκευάζεται αβίαστα για την αποκωδικοποίηση ενός κώδικα Hamming.

Είμαστε πλέον έτοιμοι να περάσουμε στο κομμάτι της μοντελοποίησης.

Μοντελοποίηση



Παραπάνω φαίνεται μια εποπτική εικόνα ενός ψηφιακού τηλεπικοινωνιακού συστήματος. Μπορούμε να παρατηρήσουμε την πληθώρα των διαδικασιών που είναι απαραίτητες ούτως ώστε να είναι εφικτή η επικοινωνία μεταξύ του πομπού και του δέκτη μέσω ενός ψηφιακού συστήματος. Στην παρούσα εργασία, ασχολούμαστε εξ' ολοκλήρου με το κομμάτι της κωδικοποίησης καναλιού. Μολαταύτα, κρίθηκε απαραίτητο να δοθεί σημασία σε δύο ακόμα σημεία του συστήματος, συγκεκριμένα στο ίδιο το κανάλι και στο κομμάτι της διαμόρφωσης. Εξηγούμε παρακάτω.

Στα πλαίσια της εργασίας έγινε αντιστοίχιση της πιθανότητας αλλαγής συμβόλου του BSC με την πιθανότητα σφάλματος bit (P_b) του καναλιού προσθετικού λευκού γκαουσιανού θορύβου (AWGN), ούτως ώστε οι επιλεγμένες τιμές να ανταποκρίνονται όσο το δυνατόν περισσότερο σε πραγματικά σενάρια μετάδοσης πληροφορίας. Συγκεκριμένα λήφθηκαν υπόψη τρεις μορφές διαμόρφωσης, PAM, PSK και QAM, για τις οποίες υπολογίστηκαν προσεγγιστικά οι πιθανότητες σφάλματος bit P_b συναρτήσει τις τάξης διαμόρφωσης και του λόγου σήματος προς θόρυβο (SNR), ενώ έγινε και υπόθεση κωδικοποίησης Gray. Με αυτόν τον τρόπο, η παράμετρος p του BSC αποκτά πλέον και φυσική σημασία. Παρ' όλα αυτά πρέπει να τονιστεί ότι δεν υφίσταται η έννοια του SNR στο δυαδικό συμμετρικό κανάλι, ούτε αυτή της διαμόρφωσης. Η παραπάνω παραδοχή έγινε επειδή ζητήθηκε στην εκφώνηση της εργασίας να συμπεριληφθούν αυτές οι δύο έννοιες στο δυαδικό συμμετρικό κανάλι και έτσι θεωρήσαμε ότι η πιθανότητα λάθους p είναι το μόνο μέγεθος θα μπορούσε να επηρεαστεί από αυτές. Έτσι, καθώς υπάρχει πλήρης ελευθερία στην επιλογή τιμών εντός του διαστήματος $(0,1)$ για την παράμετρο p, επιλέξαμε οι τιμές αυτές να ταυτίζονται με την P_b ενός AWGN.

Επιπλέον, δεδομένης της τάξης διαμόρφωσης ($\kappa = \log_2 M$, όπου M το σύνολο των bits ανά σύμβολο του αστερισμού που θα χρησιμοποιηθεί για διαμόρφωση) του μήκους της λέξης πληροφορίας (k) και της κωδικολέξης (n), είναι απαραίτητο να σταλθεί επαρκής αριθμός bits και αυτά να χωριστούν σε πακέτα συγκεκριμένου μεγέθους ώστε να μπορούν να επιτευχθούν σωστά οι διαδικασίες κωδικοποίησης και διαμόρφωσης. Σημειώνεται ότι οι παράμετροι n και k υπολογίζονται απευθείας διαλέγοντας την τάξη του κώδικα Hamming $(Matlab: hammgen(m))$. Συγκεκριμένα, κάθε πακέτο από bits πρέπει να έχει ελάχιστο μέγεθος ίσο με $s = LCM(n, \log_2 M)$. Για ποιον λόγο ισχύει αυτό; Επειδή η διαδικασία της κωδικοποίησης προηγείται αυτής της διαμόρφωσης, πρέπει τα bits του πακέτου που θα προκύψει μετά από την κωδικοποίηση να μπορούν να διαμορφωθούν από έναν αστερισμό τάξης κ, για αυτό κάθε πακέτο πρέπει να περιέχει τουλάχιστον s bits. Αυτά τα s bits αντιστοιχούν σε s/n ελάχιστο αριθμό κωδικών λέξεων ανά πακέτο οι οποίες δημιουργήθηκαν από $k * {}^{s}/_{n}$ bits πληροφορίας με χρήση ενός Hamming κώδικα (n, k). Συνεπώς, τα δεδομένα (σε bits) που θα στείλει ο πομπός, ύστερα από την διαδικασία της κωδικοποίησης, θα πρέπει να είναι ακέραιο πολλαπλάσιο του s και έπεται ότι τα ψηφιακά δεδομένα που θα παραχθούν μετά την δειγματοληψία και την κβάντιση θα πρέπει να διαιρούνται τέλεια με $k * {}^{s}/_{n}$.

Κώδικας MATLAB

Για το πρώτο μέρος δημιουργήθηκαν τέσσερεις συναρτήσεις και ένα script στο οποίο παρουσιάζεται η λειτουργία τους. Ο κώδικας είναι επαρκώς σχολιασμένος, οπότε εδώ θα γίνει αναφορά στην λειτουργία που επιτελεί.

Η πρώτη (*pb_err)* αφορά την σύνδεση του συμμετρικού δυαδικού καναλιού (BSC) με το SNR ενός AWGN καναλιού. Δεδομένου ενός SNR, μιας τάξης διαμόρφωσης M κι έναν τύπο διαμόρφωσης, επιστρέφει την αντίστοιχη πιθανότητα σφάλματος bit για το BSC. Οι προσεγγιστικοί τύποι που χρησιμοποιήθηκαν είναι οι εξής:

- PAM: $p_b = \left(\frac{2\cdot(M-1)}{Mk}\right) \cdot Q\left(\sqrt{\frac{6\gamma}{(M^2-1)}}\right)$
- PSK: $p_b = \frac{1}{k} \cdot Q\left(\sqrt{2\gamma}\, sin\left(\frac{\pi}{M}\right)\right)$
- QAM: $p_b = \frac{\left(1-\left(1-\left(\frac{2\cdot(\sqrt{M}-1)}{\sqrt{M}}\right)\cdot Q\left(\sqrt{\frac{3\gamma}{(M-1)}}\right)\right)^2\right)}{k}$

Όπου γ το SNR συμβόλου και k = $\log_2$(M).

Η δεύτερη (*hamming_sim*) προσομοιώνει την διέλευση bit πληροφορίας από BSC με κωδικοποίηση Hamming και εκτιμά την επίδοση του συστήματος για τις παραμέτρους εισόδου. Οι παράμετροι εισόδου είναι το SNR, η τάξη διαμόρφωσης M, ο τύπος διαμόρφωσης (PAM,PSK ή QAM) και το m = n – k του κώδικα Hamming (που ουσιαστικά ελέγχει άμεσα το μέγεθος της κωδικολέξης n του κώδικα). Προαιρετικά μπορεί να ρυθμιστεί και ο αριθμός των bit πληροφορίας για τα οποία θα διενεργηθεί η προσομοίωση, καθώς και η μέγιστη μνήμη που θα δεσμεύσει. Τέλος, η συνάρτηση επιστρέφει το bit error rate (BER), το block error rate καθώς και τον αριθμό των bit πληροφορίας για τα οποία εντέλει διενεργήθηκε η προσομοίωση (bits τελικά = ceil(bits αρχικά/ μέγεθος πακέτου)).

Τα βήματα που ακολουθούνται είναι τα εξής:

1. Υπολογίζεται, αν επαρκεί η μνήμη, που έχει οριστεί από τον χρήστη (προεπιλογή τα 400MB/πυρήνα), για να διενεργηθεί η προσομοίωση στον αριθμό bits πληροφορίας, που ζητήθηκε (προεπιλογή 10^6). Αν δεν επαρκεί, η προσομοίωση θα γίνει επαναληπτικά σε μικρότερα κομμάτια.
2. Υπολογίζεται ο συστηματικός πίνακας ελέγχου ισοτιμίας **H** του κώδικα Hamming που ζητήθηκε από τον χρήστη με χρήση της συνάρτησης *hammgen(m)* του MATLAB. Από αυτόν εξάγεται ο πίνακας ισοτιμίας **P**, καθώς και ο ανάστροφος του **H**, έτσι ώστε να μην χρειάζεται να υπολογιστούν ξανά σε περίπτωση επαναληπτικής προσομοίωσης. Επίσης, υπολογίζεται η πιθανότητα σφάλματος bit με χρήση της *pb_err* και δημιουργούνται τυχαία τα bit πληροφορίας, τα οποία χωρίζονται σε block μήκους k (πίνακας με k στήλες).
3. Κωδικοποιούνται τα bits πληροφορίας υπολογίζοντας τα bit ισοτιμίας με την εξίσωση «λέξη πληροφορίας * **P**» και προσαυξάνοντας την λέξη πληροφορίας με αυτά.
4. Προσομοιώνεται η επίδραση του BSC αλλάζοντας «p_b * πλήθος bit κωδικοποιημένης πληροφορίας» bit. Κάτι το οποίο γενικά ισχύει για μεγάλο αριθμό bit και είναι πολύ πιο γρήγορο από το να χρησιμοποιηθεί μια τυχαία μάσκα που θα ήταν λίγο πιο αντιπροσωπευτικός τρόπος.
5. Αποκωδικοποιούνται τα bit βάσει συνδρόμου.
    i. Υπολογίζονται τα σύνδρομα όλων των κωδικολέξεων.
    ii. Σύγκριση με τα σύνδρομα των οδηγών συνομάδας του κώδικα ($\mathbf{H}^T$).
    iii. Αλλαγή του bit που αντιστοιχεί στο συγκεκριμένο σύνδρομο.
    iv. Εξαγωγή των αποκωδικοποιημένων bits με συστηματικό τρόπο.
6. Υπολογισμός λαθών: BER και block error rate.

Η τρίτη (*parameter_sweep*) ουσιαστικά εκτελεί μια προσομοίωση για κάθε σημείο στο τετραδιάστατο πλέγμα των παραμέτρων της δεύτερης, το οποίο ορίζεται από τον χρήστη. Έχει 4 ανεξάρτητες μεταβλητές όσες και οι παράμετροι εισόδου της δεύτερης κι επιστρέφει έναν 5Δ πίνακα με το BER και το block error rate για κάθε συνδυασμό τιμών των ανεξάρτητων μεταβλητών. Επίσης δεδομένου χρόνου συμβόλου, επιστρέφει και τους ρυθμούς μετάδοσης για κάθε συνδυασμό τάξης διαμόρφωσης M και μήκους κωδικολέξης n.

Τέλος, η τέταρτη (*ecc_plot_1)* απλά αυτοματοποιεί την δημιουργία διαγραμμάτων για το πρώτο μέρος, οπότε δεν κρίθηκε απαραίτητο να σχολιαστεί περαιτέρω.

Αποτελέσματα

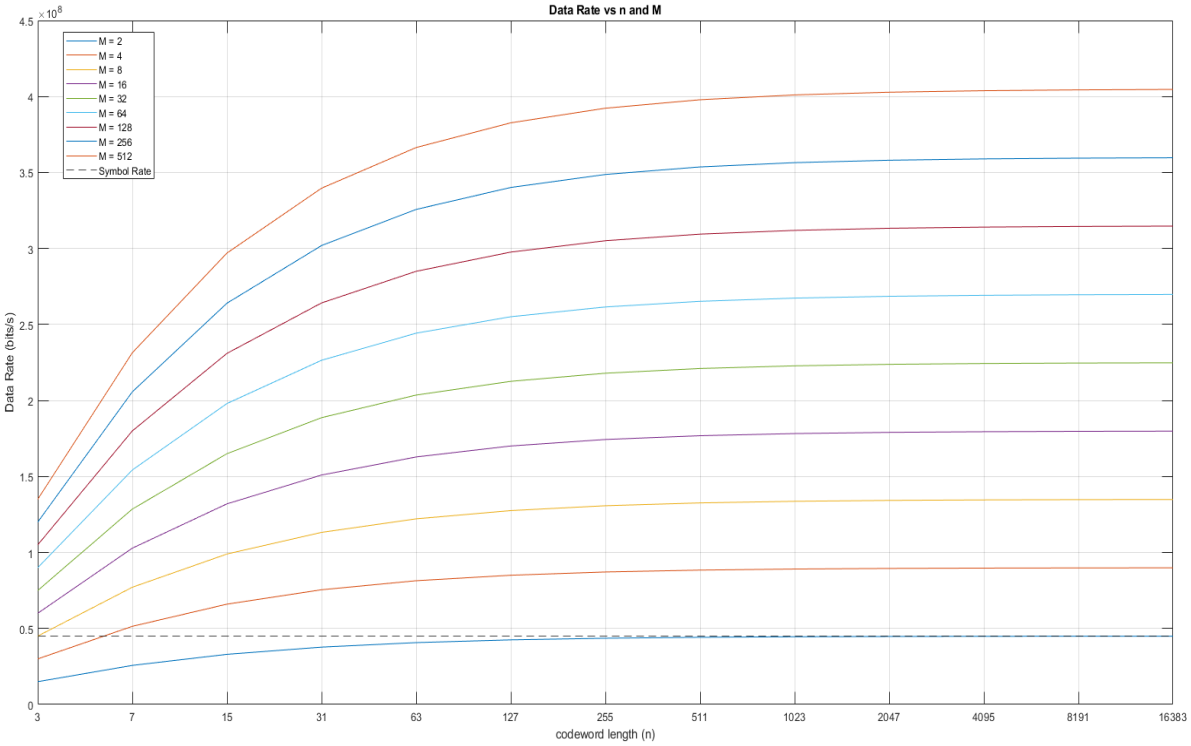
Για το πρώτο ερώτημα, διενεργήθηκε προσομοίωση για κάθε συνδυασμό των παρακάτω παραμέτρων:

- SNR = 0, 5, 10, ...,35 dB
- M = 2, 4, 8, …,512
- m = 2, 3, …, 14 => n = 3, 7, 15, 31, …, 16383

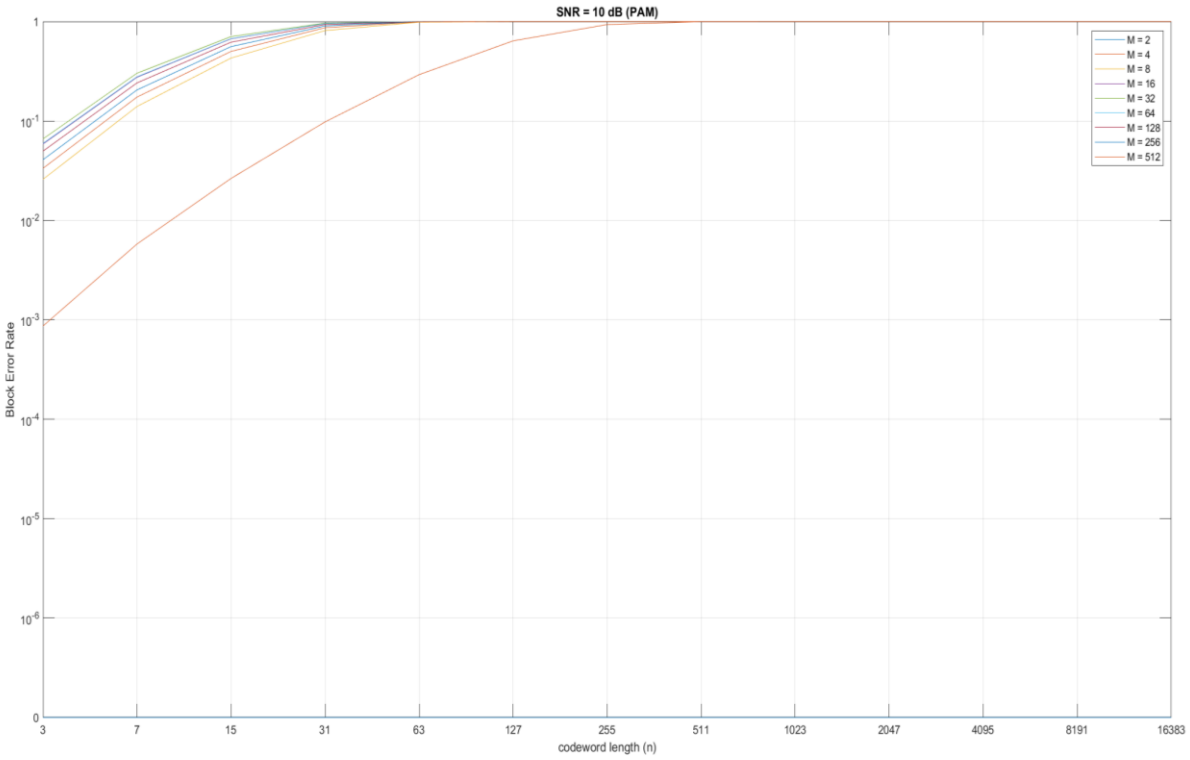
- τύπος διαμόρφωσης: MPAM, MPSK, MQAM

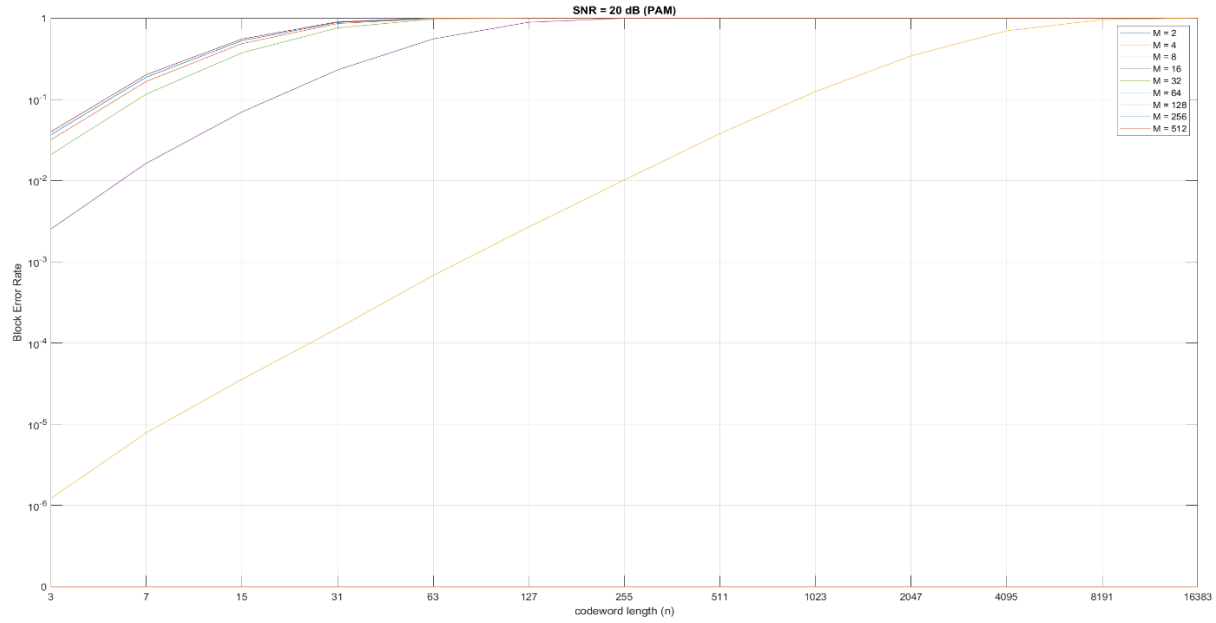
Επίσης, για να έχει κάποια φυσική σημασία, αυθαίρετα επιλέχθηκε bandwidth καναλιού 45MHz, και χρόνος συμβόλου $T_s = \frac{10^{-6}}{45}$ s (για να μπορούμε να θεωρήσουμε μηδενική ISI κι επομένως η αντιστοίχιση p_b με SNR να ισχύει).

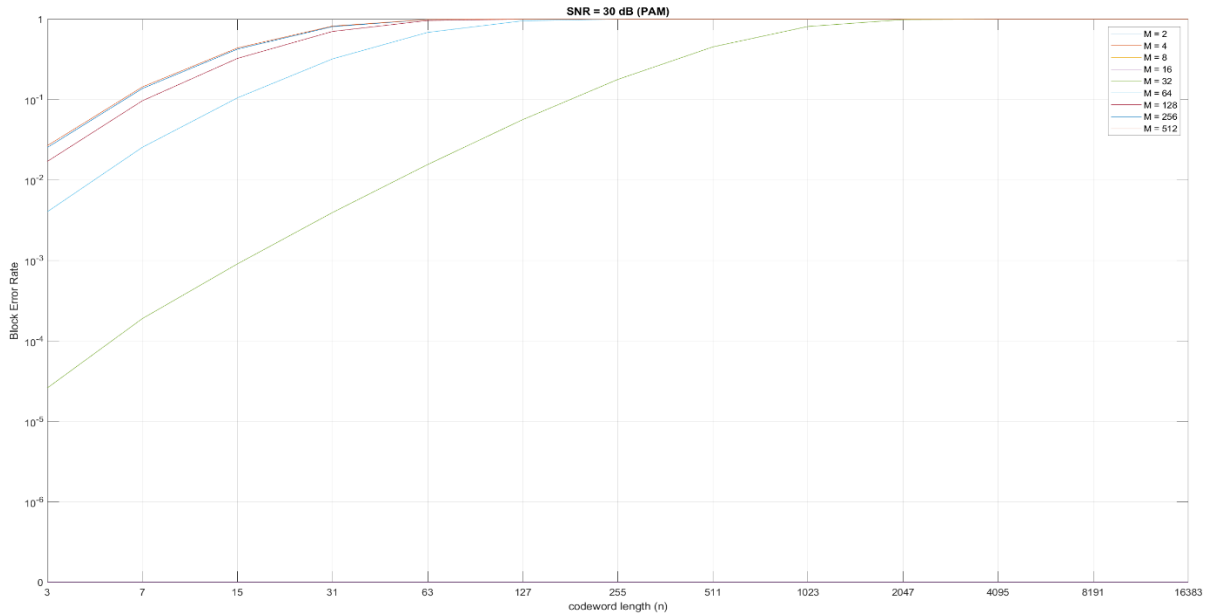
Η επίδραση της τάξης διαμόρφωσης σε συνδυασμό με το μήκος κωδικολέξης στον ρυθμό μετάδοσης γίνεται εμφανής στο παρακάτω διάγραμμα. Είναι φανερό πως η τάξη διαμόρφωσης είναι ανάλογη του ρυθμού μετάδοσης, ενώ το σημείο που αρχίζει να έχει μικρότερη επίδραση στον ρυθμό μετάδοσης το μήκος της κωδικολέξης (elbow του διαγράμματος) είναι το n = 63 που έχει ρυθμό κώδικα 0.9047.

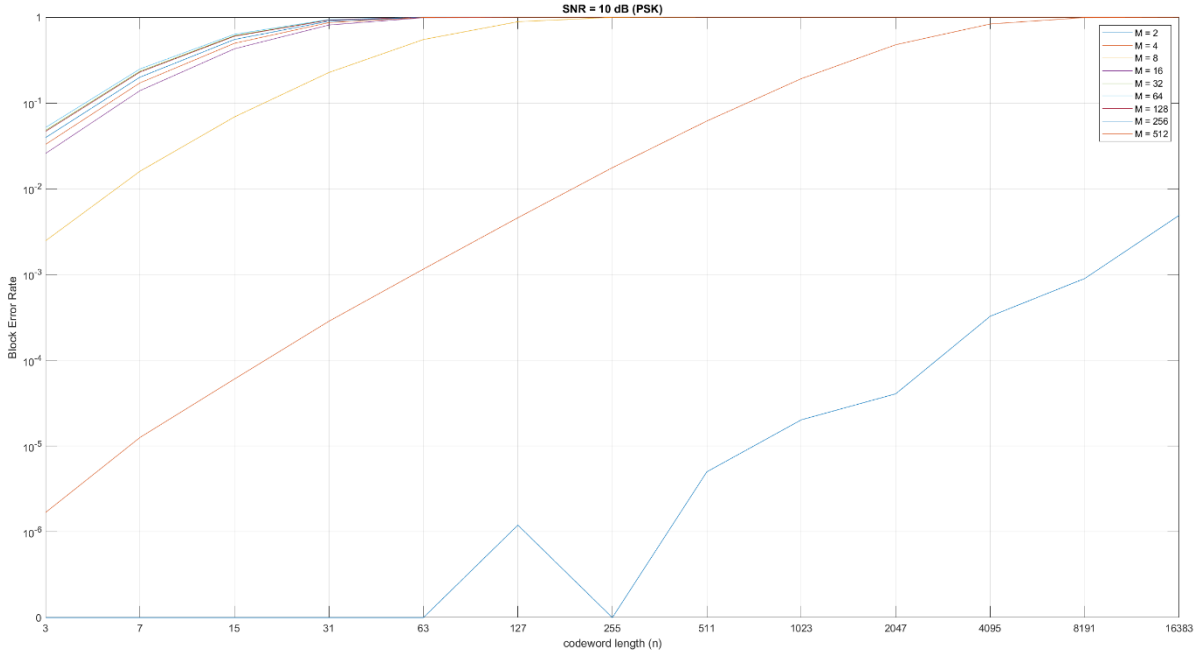


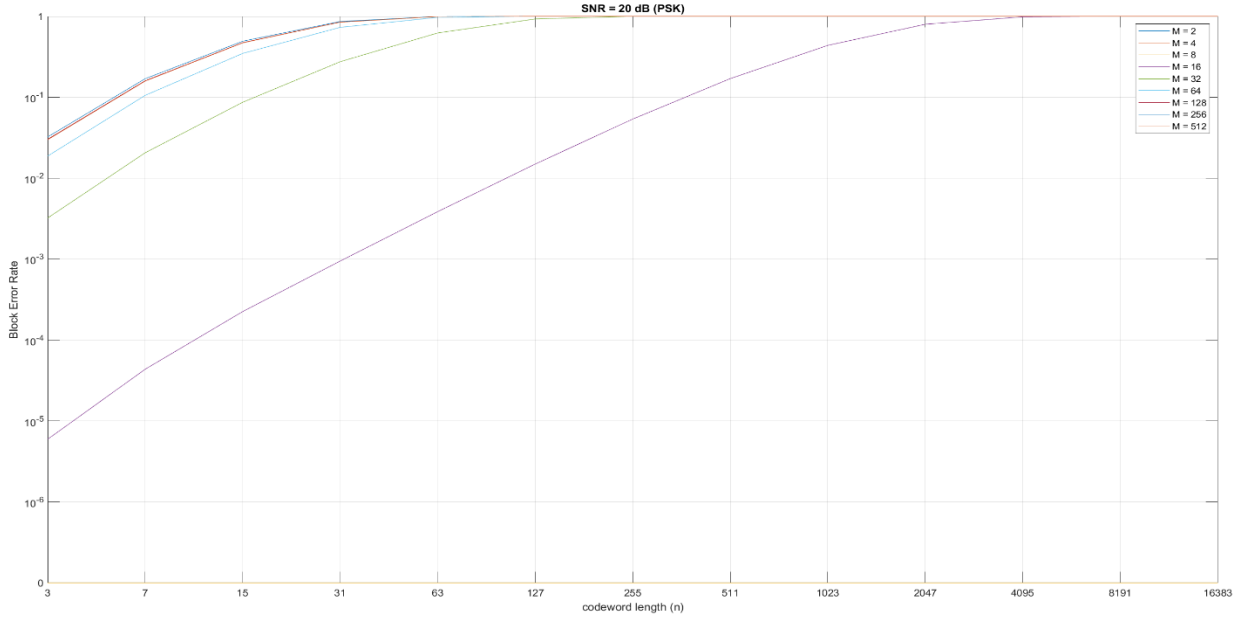
Ωστόσο, πρέπει να ληφθεί υπόψη και η ικανότητα διόρθωσης λαθών, για αυτό παρακάτω παρουσιάζεται και η επίδρασή τους στον ρυθμό σφάλματος μπλοκ, για τρεις διαφορετικές διαμορφώσεις και τρία διαφορετικά SNR (10,20,30 dB). Πρώτα παρουσιάζονται τα διαγράμματα για την MPAM, μετά για την MPSK και τέλος για την MQAM, με αύξον SNR. Οποιαδήποτε τιμή κάτω από 10^{-7} θεωρείται μηδενική.

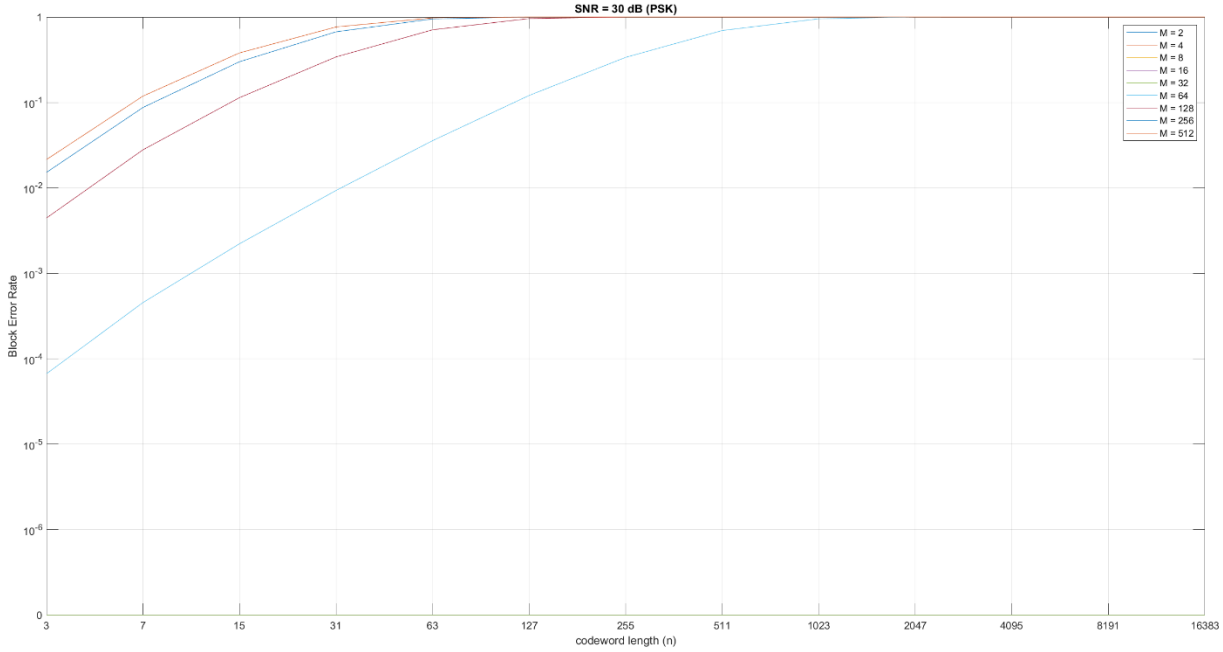

SNR = 10 dB (PAM)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512

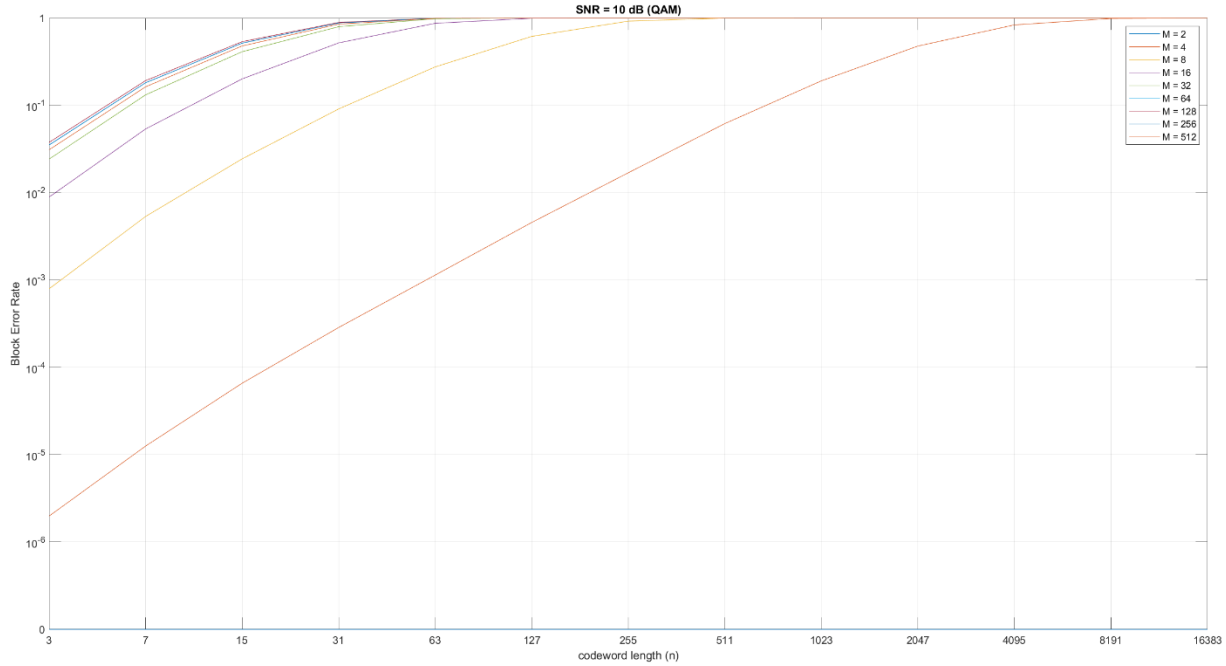

SNR = 20 dB (PAM)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512

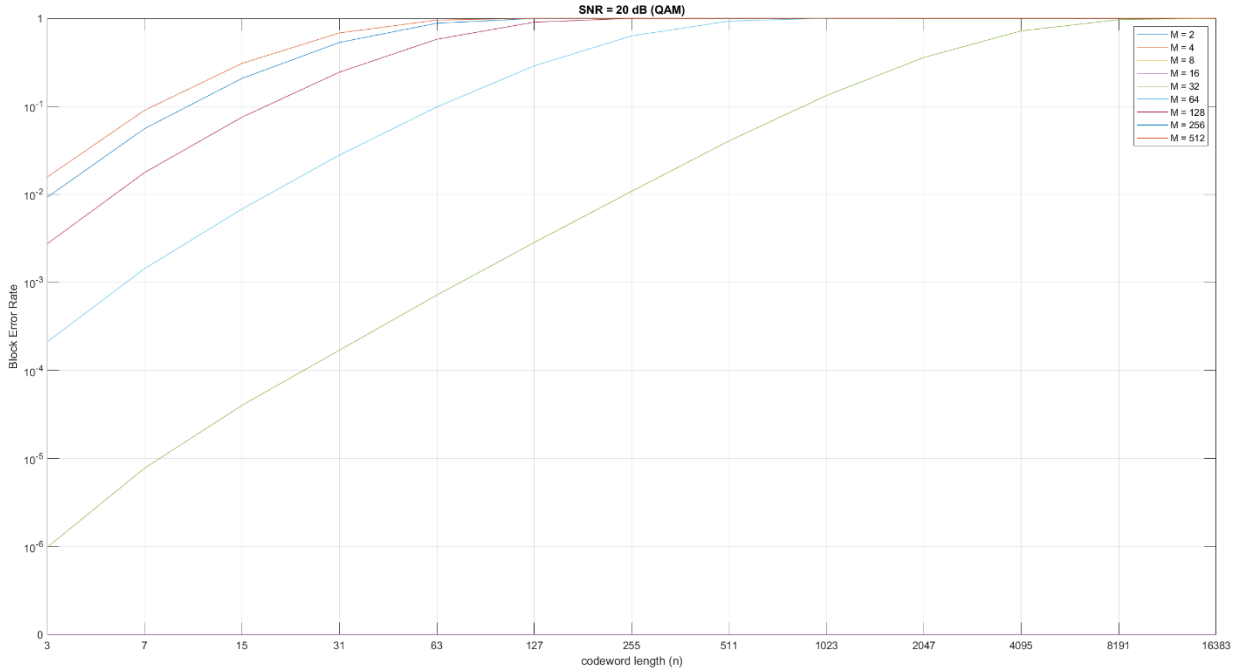

SNR = 30 dB (PAM)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512

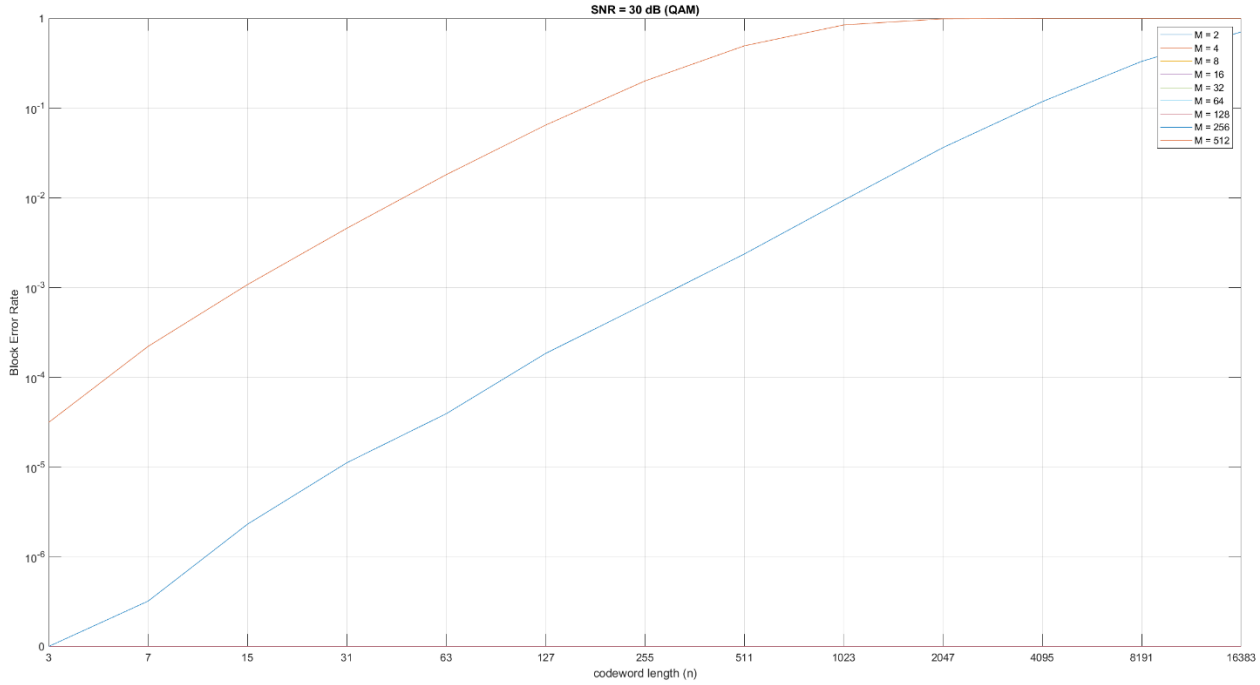

SNR = 10 dB (PSK)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512


SNR = 20 dB (PSK)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512


SNR = 30 dB (PSK)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512


SNR = 10 dB (QAM)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512

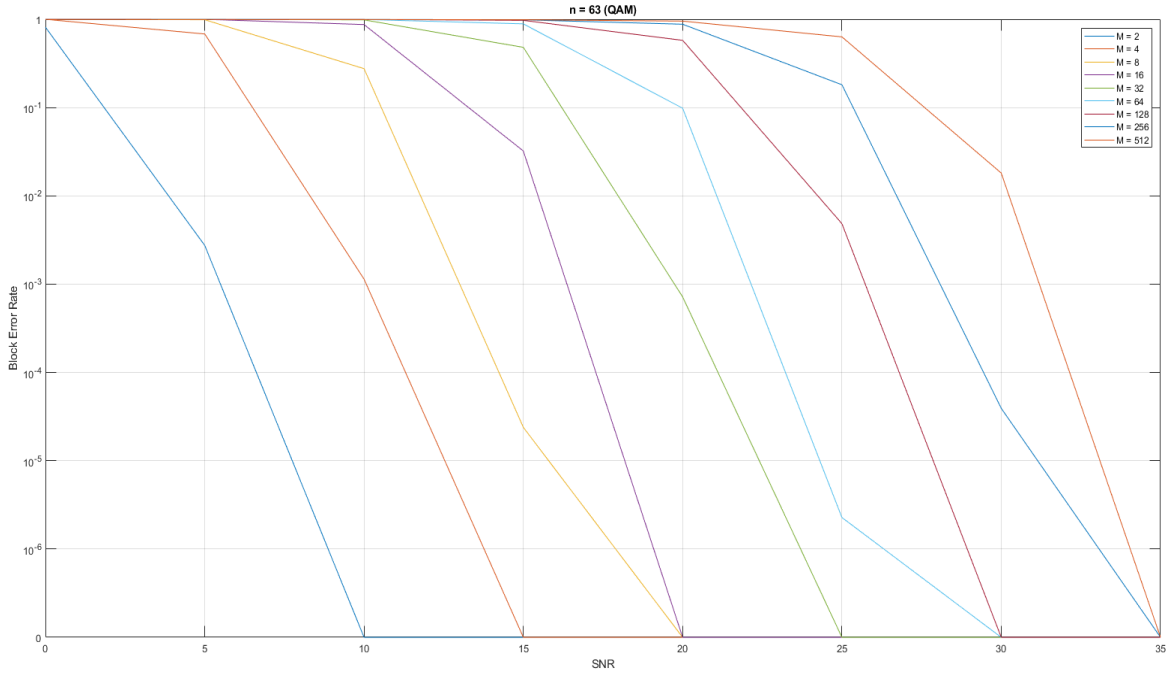

SNR = 20 dB (QAM)
Block Error Rate
codeword length (n)
M = 2
M = 4
M = 8
M = 16
M = 32
M = 64
M = 128
M = 256
M = 512



Αυτό που συμπεραίνουμε, αν αυθαίρετα θέσουμε ως όριο επίτευξης επικοινωνίας ρυθμό σφάλματος μπλοκ κάτω από 0.001, είναι πως για χαμηλά SNR δεν μπορούμε να χρησιμοποιήσουμε υψηλή τάξη διαμόρφωσης σε συνδυασμό με μεγάλη κωδικολέξη. Αυτό που πρέπει, να γίνει είναι ένας συμβιβασμός που θα δώσει τον βέλτιστο ρυθμό μετάδοσης ικανοποιώντας αυτό το αίτημα. Από τα δεδομένα μας υπολογίζονται οι παρακάτω συνδυασμοί για τις διάφορες τιμές SNR (για MQAM):

- SNR = 0 dB: δεν μπορεί να επιτευχθεί το όριο για την επικοινωνία με αυτόν τον κώδικα.
- SNR = 5 dB: M = 2, n = 31, rate = 37.74 Mbps
- SNR = 10 dB: M = 4, n = 31, rate = 75.48 Mbps
- SNR = 15 dB: M = 8, n = 255, rate = 130.77 Mbps
- SNR = 20 dB: M = 32, n = 63, rate = 203.57 Mbps
- SNR = 25 dB: M = 64, n = 1023 rate = 267.36 Mbps
- SNR = 30 dB: M = 256, n = 255 rate = 348.71 Mbps
- SNR = 35 dB: M = 512. n = 16383, rate = 404.66Mbps

Παρατηρούμε επίσης πως ο κώδικας Hamming είναι εξαιρετικά χρήσιμος σε συνθήκες υψηλού SNR, που τα σφάλματα είναι σχετικά σπάνια, γιατί ο κωδικός του ρυθμός πλησιάζει τη μονάδα. Επίσης βλέπουμε την αναμενόμενη επίδραση που έχει ο σηματοθορυβικός λόγος στον ρυθμό μετάδοσης, δεδομένου κατωφλίου επικοινωνίας. Τέλος ενδεικτικά βλέπουμε την επίδραση του SNR στο παρακάτω διάγραμμα.



Μετάδοση πληροφορίας μέσω καναλιών με διαφορετική πιθανότητα λάθους

Γνωρίζουμε από την θεωρία πληροφορίας ότι το άνω όριο όλων των επιτεύξιμων κωδικών ρυθμών σε ένα κανάλι είναι το Operational Capacity του καναλιού. Επίσης, από το Θεώρημα κωδικοποίησης θορυβώδους καναλιού του Shannon (Shannon Noisy Channel Coding theorem), το Operational Capacity του καναλιού είναι ίσο με το Information Capacity, $\max_{p(x)} I(X;Y)$. Για ευκολία, από ‘δω και πέρα θα αναφερόμαστε στις παραπάνω έννοιές απλώς ως (Channel) Capacity ή στα ελληνικά χωρητικότητα καναλιού με μονάδα μέτρησης bits/μετάδοση (ή bits/trans). Υποθέτουμε ότι δεν υπάρχουν overhead bits και συνεπώς το Gross Bit Rate (R_b), χωρίς τα FEC bits, μπορεί να θεωρηθεί ίσο με το Net Bit Rate (R). Το Net Bit Rate του καναλιού επηρεάζεται από το Code Rate (r) του κώδικα που χρησιμοποιείται για Channel Coding. Συγκεκριμένα, ο ρυθμός του κώδικα μας δείχνει τι ποσοστό από τα bits που φέρουν πληροφορία μπορούν να μεταδοθούν στο κανάλι ανά κωδικολέξη. Έτσι ρυθμός κώδικα r θα πρέπει να είναι μεγαλύτερος από τον ρυθμό μετάδοσης του καναλιού, ώστε τα bit streams που στέλνονται να ‘προλάβουν’ να κωδικοποιηθούν με τον επιθυμητό τρόπο πριν αναμεταδοθούν, αλλά ταυτόχρονα μικρότερος από το Channel Capacity για να είναι εφικτή η επικοινωνία με αυθαίρετα μικρή πιθανότητα σφάλματος. Έτσι, για δεδομένη διάρκεια συμβόλου T_s, άρα και δεδομένο ρυθμό μετάδοσης R (πάντα μικρότερο του C), μπορούμε διαλέξουμε έναν ρυθμό κώδικα r από τους διαθέσιμους ώστε να η επικοινωνία να είναι επιτυχής με αυθαίρετα μεγάλη πιθανότητα.

Έστω ότι έχουμε διαθέσιμα δύο δυαδικά συμμετρικά κανάλια με διαφορετική πιθανότητα λάθους και θέλουμε να μεταφέρουμε πληροφορία ταυτόχρονα και από τα δύο αυτά κανάλια. Θέλουμε να εξασφαλίσουμε ταυτόχρονη μετάδοση και από τα δύο κανάλια (υποθέτουμε πως είναι τέλεια συγχρονισμένα). Έστω $C_a = 1 - H_b(p_a)$ η χωρητικότητα του πρώτου καναλιού και $C_\beta = 1 - H_b(p_\beta)$ η χωρητικότητα του δεύτερου καναλιού και έστω $C_a > C_\beta$ χωρίς βλάβη της γενικότητας. Έστω r_α, r_β η κωδικοί ρυθμοί του κώδικα που χρησιμοποιείται για μετάδοση πληροφορίας σε κάθε κανάλι αντίστοιχα. Όπως δείξαμε παραπάνω πρέπει να ισχύει $r < C$ ώστε να έχουμε επιτυχημένη μετάδοση πληροφορίας σε ένα κανάλι. Στον παρακάτω πίνακα παραθέτουμε του κωδικούς ρυθμούς των κωδίκων Hamming τάξης m από 2 έως 10 και τα διαστήματα απόφασης όποτε αυτά θα εξηγηθούν παρακάτω.

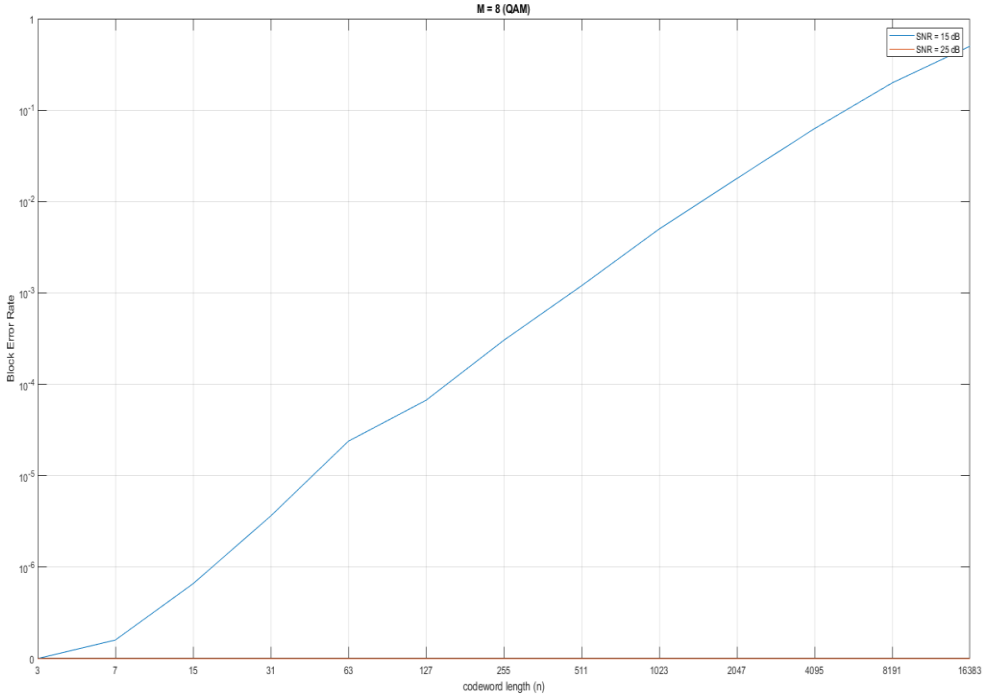
(n,k)	r	Δ
(3,1)	0.3333	$\Delta_1 = (0.3333,0.5714)$
(7,4)	0.5714	$\Delta_2 = (0.5714,0.0.7333)$
(15,11)	0.7333	$\Delta_3 = (0.7333,0.8387)$
(31,26)	0.8387	$\Delta_4 = (0.8387,0.9047)$
(63,57)	0.9047	$\Delta_5 = (0.9047,0.9448)$
(127,120)	0.9448	$\Delta_6 = (0.9448,0.9686)$
(255,247)	0.9686	$\Delta_7 = (0.9686,0.9823)$
(511,502)	0.9823	$\Delta_8 = (0.98230.9902)$
(1023,1013)	0.9902	$\Delta_9 = (0.9902,1)$

Η χαμηλότερη τιμή του r παρατηρείται για crossover probability περίπου ίση με 0.1741, που είναι μια πάρα πολύ μεγάλη τιμή για πιθανότητα σφάλματος σε AWGN κανάλι, αντιστοιχεί σε κάκιστες συνθήκες SNR. Συνεπώς δεν έχει νόημα να συμπεριλάβουμε χαμηλότερες περιπτώσεις του Channel Capacity. Μια πιθανότητά σφάλματος της τάξεως του 10^{-3} αντιστοιχεί Capacity στο BSC εντός του διαστήματος (0.9192,0.9886) οπότε βλέπουμε ότι ακόμα και σε περιπτώσεις πιθανότητας σφάλματος που προκύπτουν από μέτριο προς κακό SNR περιμένουμε η αντίστοιχη χωρητικότητα του BSC να μην πέσει κάτω από το 0.9 . Τα διαστήματα απόφαση Δ_i μας δείχνουν το πλήθος των τιμών της χωρητικότητάς καναλιών που υποστηρίζουν ρυθμό κώδικα r_i χωρίς να υποστηρίζουν ρυθμό κώδικα r_{i+1}. Σταματάμε στο Hamming (1023,1013) για λόγους memory management. Έτσι αν για παράδειγμα έχουμε $C_\alpha = 0.98$ και $C_\beta = 0.9$ βλέπουμε ότι $C_\alpha \in \Delta_7$ ενώ $C_\beta \in \Delta_4$ οπότε η βέλτιστες επιλογές μήκος κωδικολέξης είναι $l_\alpha = 255$ με ρυθμό κώδικα $r_\alpha = 0.9686$ και $l_\beta = 31$ με ρυθμό κώδικα $r_\beta = 0.8387$. Η επιλογή αποστολής της ίδιας κωδικολέξης κρίνεται βέλτιστη μόνο όταν οι χωρητικότητες και των δύο καναλιών ανήκουν εντός του ίδιου διαστήματος απόφασης Δ_i. Τονίζεται ότι παρόλο που ένας ρυθμός κώδικα μπορεί να δείχνει ότι αγγίζει θεωρητικά το channel capacity πρακτικά υπάρχουν διάφορα εμπόδια και οι κώδικες Hamming δεν είναι κώδικες που προσεγγίζουν ασυμπτωτικά την χωρητικότητα του καναλιού.

Αποτελέσματα

Σε αυτό το μέρος επιλέχθηκε αυθαίρετα διαμόρφωση 8-QAM και για τα δύο κανάλια. Το πρώτο «κακό» κανάλι έχει SNR ίσο με 15 dB, το δεύτερο «καλό», έχει SNR ίσο με 25 dB. Θεωρώντας και στα 2 κανάλια προϋπόθεση για επικοινωνία block error rate < 0.001, ο καλύτερος ρυθμός μετάδοσης επιτυγχάνεται αν στο «κακό» κανάλι χρησιμοποιηθεί μήκος κωδικολέξης 255 και στο «καλό» 16383, και είναι ίσος με 265.65 Mbps (με τις ίδιες υποθέσεις για το Ts με το πρώτο ερώτημα). Αν ωστόσο, δεν υπάρχει κατώφλι και μας ενδιαφέρει το throughput (με βάση το block error rate), ο καλύτερος συνδυασμός είναι 1023 στο κακό και 16383 στο καλό με 267.9 Mbps. Γενικά με αυτήν την ερμηνεία του ερωτήματος, όπως αναπτύχθηκε και θεωρητικά, σχεδόν πάντα θα είναι προτιμότερο να χρησιμοποιείται διαφορετικό μήκος κωδικολέξης για κάθε κανάλι. Ωστόσο αν τα δύο κανάλια είχαν ίδιο SNR, τότε η βέλτιστη λύση θα ήταν να στέλνουμε την ίδια κωδικολέξη.

Το παρακάτω διάγραμμα απεικονίζει της δύο καμπύλες ενδιαφέροντος για τα δύο κανάλια. Με μπλε το «κακό» και με κόκκινο το «καλό», που έχει μηδενική πιθανότητα σφάλματος μπλοκ για κάθε n που δοκιμάσθηκε.

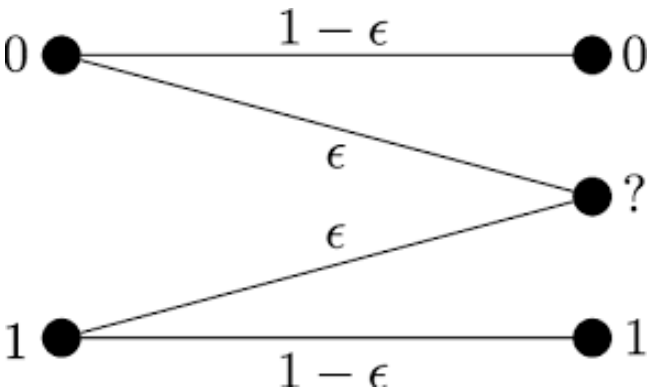


Κωδικοποίηση στο δυαδικό κανάλι με διαγραφές

Στο παρόν κεφάλαιο εξετάζουμε το πρόβλημα κωδικοποίησης για μετάδοση πληροφορίας μέσω του δυαδικού καναλιού με διαγραφές με χρήση LDPC κωδίκων. Στην αρχή παρουσιάζεται με σύντομη θεωρητική επισκόπηση και στην συνέχεια γίνεται πρακτική υλοποίηση του προβλήματος και μελέτη επίδοσης regular και irregular LDPC κωδίκων με χρήση λογισμικού, κώδικα Matlab.

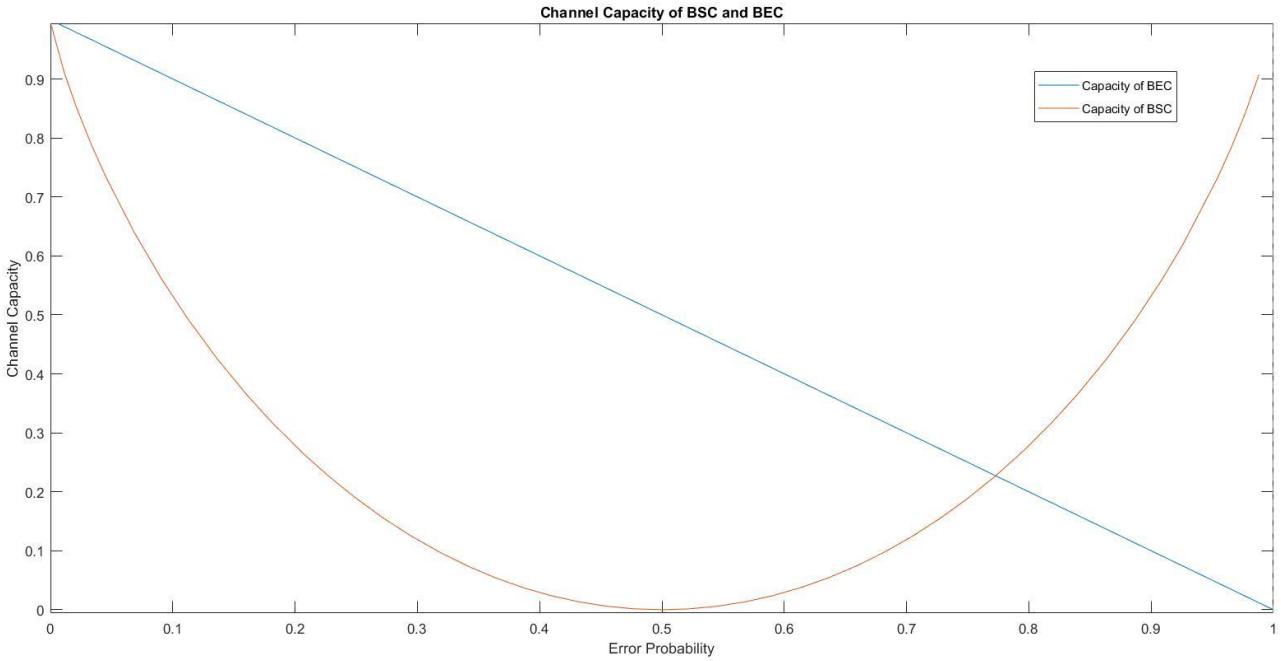
Στοιχεία από την Θεωρία Πληροφορίας

Το δυαδικό κανάλι διαγραφής (Binary Erasure Channel ή συνοπτικά BEC) είναι ένα θεωρητικό κανάλι της θεωρίας πληροφορίας και ορίζεται όπως παρακάτω.



Η παράμετρος που το χαρακτηρίζει είναι η πιθανότητα διαγραφής ε (erasure probability). Οι δεσμευμένες πιθανότητες $P(Y|X)$, όπου X το σήμα που στέλνει ο πομπός και Y το σήμα που λαμβάνει ο δέκτης είναι $P(Y=0|X=0)=P(Y=1|X=1)=1-\varepsilon$ και $P(Y=?|X=0)=P(Y=?|X=1)=\varepsilon$. Επιπλέον, παρατηρούμε ότι υπάρχει απόλυτη βεβαιότητα για το σύμβολο που στάλθηκε αν ο δέκτης λάβει 0 ή 1 ενώ διαγραφή μπορεί να προκύψει με ίδια πιθανότητα για οποιοδήποτε σήμα σταλεί από τον πομπό. Αυτό μεταφράζεται στην γλώσσα των πιθανοτήτων ως $P(X=0|Y=0)=P(X=1|Y=1)=1$ και $P(X=0|Y=?)=P(X=1|Y=?)=\frac{1}{2}$. Η χωρητικότητα κατά Shannon του BEC είναι

ίση με $C_{BEC} = 1 - \varepsilon$. Παρατηρούμε ότι σε αντίθεση με το BSC, η χωρητικότητα του BEC είναι γνησίως μονότονη συνάρτηση που φθίνει γραμμικά. Παρακάτω φαίνονται σε κοινό διάγραμμα οι χωρητικότητες των BSC και BEC.



Παρατηρούμε ότι τομή των δύο διαγραμμάτων παρατηρείται για $p = e \approx 0.773$ ενώ γενικά το BEC παρουσιάζει υψηλότερη χωρητικότητα από το BSC μικρομεσαίες τιμές της πιθανότητας σφάλματος. Η αύξηση της C_{BSC} για crossover probability πάνω από 0.5 είναι αναμενόμενη καθώς αν ξέρουμε ότι τα περισσότερα bits θα φτάσουν λάθος στον δέκτη τότε εφαρμόζουμε bit flipping πριν από την αποκωδικοποίηση και οδηγούμαστε στην συμμετρική περίπτωση, κάτι δεν μπορεί να εφαρμοστεί στο BEC.

LDPC κώδικες

Οι LDPC κώδικες είναι μια ειδική μορφή γραμμικών κωδίκων. Πρωτοπαρουσιάστηκαν από τον Robert Gallagher (ένας από τους σημαντικότερους επιστήμονες στον κλάδο της θεωρίας πληροφορίας και κάτοχο βραβείου Shannon) στην διδακτορική διατριβή του το 1960. Παρόλα αυτά, δεν χρησιμοποιήθηκαν για τα υπόλοιπα 30 χρόνια λόγο της υψηλής πολυπλοκότητας της επαναληπτικής αποκωδικοποίησης (iterative decoding) που απαιτούσαν για την εποχή εκείνη. Το ενδιαφέρον ως προς τους LDPC κώδικες αναστήθηκε ύστερα από την εφεύρεση των Turbo κωδίκων οι οποίοι με χρήση iterative decoding ήταν οι πρώτοι κώδικες που κατάφεραν να προσεγγίσουν πρακτικά το channel capacity του Shannon. Τα πρώτα βήματα επαναχρησιμοποίησης αυτών έγιναν από τον David MacKay. Έκτοτε, οι LPDC κώδικες έχουν γνωρίσει ραγδαία ύφεση και έχουν πλέον καθιερωθεί ως η κύρια μορφή κωδικοποίησης σε πρότυπα ενσύρματης και ασύρματης επικοινωνίας. Μαζί με τους Turbo κώδικες και τους Polar κώδικες αποτελούν τους μοναδικούς ως σήμερα γνωστούς capacity-approaching κώδικες, υπερτερούν όμως όλων σε απόδοση σε υψηλούς ρυθμούς κώδικα και μεγάλα μήκη κωδικολέξεων. Είναι επίσης οι μοναδικοί γραμμικοί κώδικες που ταυτόχρονα προσεγγίζουν το capacity του Shannon και

ικανοποιούν το όριο Gilbert-Varshamov για γραμμικούς κώδικες. Πάμε να δούμε όμως τι είναι αυτοί οι περιβόητοι κώδικες.

LDPC = Low Density Parity Check. Τι μας λέει η διατύπωση αυτού το ακρωνυμίου; Πολύ απλά ότι αναμένουμε ο parity check matrix $\boldsymbol{H}$ αυτών να περιέχει πάρα πολλά μηδενικά και λίγους άσσους (sparse binary matrix). Πρώτος ο Gallagher όρισε τους LDPC κώδικες με τον εξής τρόπο:
$C_{LDPC}(n, s, v)$ όπου

- n είναι το μήκος της κωδικολέξης
- s είναι ο αριθμός των άσσων ανά στήλη του $\boldsymbol{H}$
- v είναι ο αριθμός των άσσων ανά γραμμή του $\boldsymbol{H}$

Ο ρυθμός του παραπάνω κώδικα είναι $r = \frac{v-s}{v}$ με την προϋπόθεση ότι όλες οι γραμμές του $\boldsymbol{H}$ είναι ανεξάρτητες. Ειδάλλως, ο ρυθμός κώδικα υπολογίζεται ως $r = \frac{n-s'}{n}$ όπου s' είναι η διάσταση του χώρου των γραμμών του $\boldsymbol{H}$. Είναι προφανές ότι ο παραπάνω ορισμός παραπέμπει σε LDPC κώδικες με σταθερό αριθμό άσσων στις γραμμές και στις στήλες του $\boldsymbol{H}$. Η παραπάνω μορφή ενός Parity Check Matrix λέγεται regular μορφή και οι αντίστοιχοι κώδικες regular LDPC (θυμόμαστε ότι ο πίνακας $\boldsymbol{H}$ περιγράφει πλήρως έναν κώδικα). Σε περίπτωση που ο αριθμός των μηδενικών και των άσσων ποικίλει ανά γραμμή και ανά στήλη τότε γίνεται λόγος για irregular LDPC κώδικες τους οποίους θα αναλύσουμε αργότερα. Όσον αφορά τους regular LDPC κώδικες, στο εξής θα τους συμβολίζουμε με (l, r) όπου l είναι ο αριθμός των άσσων ανά στήλη του $\boldsymbol{H}$ και r είναι ο αριθμός των άσσων ανά γραμμή του $\boldsymbol{H}$. Έτσι ο κωδικός ρυθμός δίνεται ως $r = \frac{k}{n} = 1 - \frac{n-k}{n} = 1 - \frac{l}{\mathrm{r}}$.

Η πλέον κατεστημένη απεικόνιση ενός LDPC κώδικα γίνεται μέσω του Tanner Graph (bibartite graph) που αντιστοιχεί στον πίνακα $\boldsymbol{H}_{(\boldsymbol{n-k})\times\boldsymbol{n}}$ αυτού. Συγκεκριμένα, το Tanner Graph ενός κώδικα αντιστοιχεί σε έναν γράφο με n κόμβους (εν ονόματι κόμβοι μεταβλητών) που συνδέονται σε $n-k$ κόμβους (εν ονόματι κόμβοι ελέγχου). Με την παραπάνω αντιστοιχία, μπορούμε να αναφερόμαστε πλέον ως l στον βαθμό των κόμβων μεταβλητών και ως r στον βαθμό των κόμβων ελέγχου. Έτσι για παράδειγμα ένας regular $(3,6)$ LDPC κώδικας με έναν πίνακα ελέγχου ισοτιμίας διάστασης 100×200 αντιστοιχίζεται σε ένα Tanner Graph με 100 κόμβους ελέγχου βαθμού 6 και 200 κόμβους μεταβλητών βαθμού 3. Η αναπαράσταση κωδίκων μέσω γράφων διευκολύνει πολύ στην κατανόηση και υλοποίηση αυτών. Αρχικά αναφέρουμε ότι οι ακμές που καταλήγουν σε έναν κόμβο ελέγχου μας δίνουν πληροφορία για το ποια bits της κωδικολέξης εμπεριέχονται σε εκείνη την συγκεκριμένη εξίσωση ελέγχου ισοτιμίας (parity check equation). Ανάποδα, οι ακμές που καταλήγουν σε έναν κόμβο μεταβλητών από διάφορους κόμβους ελέγχου μας δείχνουν σε πόσες parity check equations συμπεριλαμβάνεται το συγκεκριμένο bit της κωδικολέξης. Επιπροσθέτως, ο τρόπος με τον οποίο μεγαλώνει ο αριθμός των ακμών ενός γράφου σε σχέση με το μήκος κωδικολέξης n μας πληροφορεί για το density evolution του κώδικα στον οποίο αντιστοιχεί ο γράφος. Ένα πολύ ισχυρό χαρακτηριστικό των LDPC κωδίκων είναι ότι ο αριθμός των ακμών των Tanner Graphs τους αυξάνεται γραμμικά με το n ενώ στην πλειοψηφία των κωδίκων διόρθωσης σφαλμάτων ο αριθμός αυτός αυξάνεται ανάλογα με το τετράγωνο του n. Αυτό σύμφωνα με το γλωσσάρι της θεωρίας κωδίκων ονομάζεται low density evolution. Η μορφή του Tanner Graph ενός regular LDPC κώδικα είναι πολύ απλή όπως βλέπουμε έως τώρα. Τι συμβαίνει όμως στην περίπτωση των irregular LDPC κωδίκων;

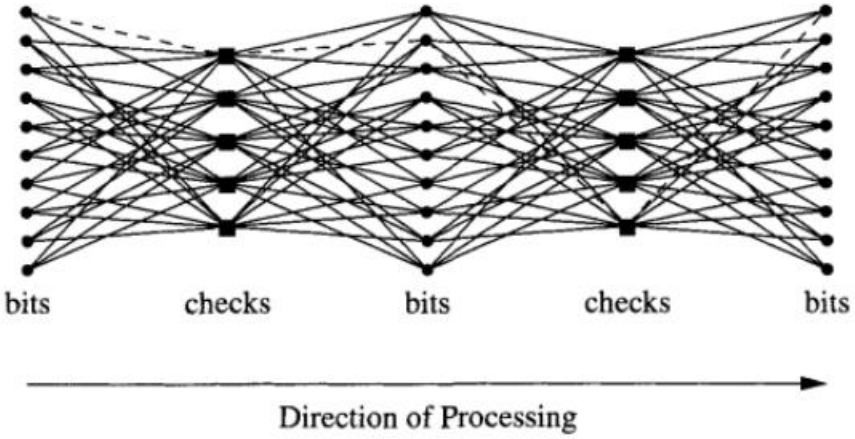
Σκεφτείτε ένα Tanner Graph με δεδομένο αριθμό κόμβων μεταβλητών και ελέγχου στο οποίο από οποιονδήποτε κόμβο μεταβλητών θα μπορούσαμε να καταλήξουμε σε οποιονδήποτε κόμβο ελέγχου χωρίς κανέναν περιορισμό. Αν τραβούσαμε απλώς τυχαία ακμές και αντιστοιχούσαμε τους κόμβους χωρίς να μεσολαβήσει καμία απολύτως αιτιοκρατική διαδικασία τότε με βεβαιότητα θα καταλήγαμε σε μια απεικόνιση ενός irregular

LDPC κώδικα. Προφανώς όμως δεν θα μπορούσε αυτή η τυχαία αναπαράσταση να οδηγεί σε καλούς κώδικες, σωστά; Η απάντηση αποκλίνει μερικώς εκ του προφανούς. Σαφώς μια τελείως τυχαία αναπαράσταση οδηγεί σε 'κακούς' κώδικες. Για αυτό ας ορίσουμε ως αφετηρία ότι θέλουμε ο κώδικας μας να έχει έναν συγκεκριμένο ρυθμό και να μην αποκλίνει πολύ από αυτόν. Τότε αποδεικνύεται ότι βελτιστοποιώντας τις παραμέτρους που τελικά θα μας δώσουν την κατανομή των βαθμών των κόμβων, μπορούμε να παράγουμε πολύ καλούς LDPC κώδικες, για μεγάλα μήκη κωδικολέξης, με τυχαίες αντιστοιχίσεις μεταξύ των κόμβων μεταβλητών και ελέγχου! Αυτό το πρόβλημα βελτιστοποίησης θα το ορίσουμε παρακάτω, αλλά στην παρούσα φάση σκοπός μας είναι να δείξουμε πως ακόμα και με μια αρκετά 'ελαστική' κατασκευή των Tanner Graphs (και συνεπώς των πινάκων ελέγχου ισοτιμίας $\boldsymbol{H}$) μπορούμε να λάβουμε LDPC κώδικες με πολύ καλές επιδόσεις. Γενικά έχοντας λύσει τα προβλήματα βελτιστοποιήσεων που θα ορίσουμε σε παρακάτω παράγραφο, έχουμε δύο επιλογές. Είτε να κατασκευάσουμε με τυχαίο τρόπο των parity check matrix (random LDPC) και να αναμένουμε πολύ μεγάλο μήκος κωδικολέξης για να έχουμε επιδόσεις 'κοντά' στο channel capacity ή να κατασκευάσουμε τον parity check matrix με μια (μερικώς ή ολοκλήρως) αιτιοκρατική διαδικασία (structured LDPC) η οποία θα είναι χρονικά πιο ακριβή αλλά θα παράγει κώδικες καλύτερης απόδοσης για μικρότερα μήκη κωδικής λέξης. Έχει αποδειχθεί ότι ψευδό-τυχαίες (pseudo-random) διαδικασίες κατασκευής του $\boldsymbol{H}$ matrix οδηγούν στην βέλτιστη απόδοση. Μία τέτοια διαδικασία επιλέξαμε και εμείς για να κατασκευάσουμε τους LDPC κώδικες μας.

Πριν περάσουμε στην πρακτική υλοποίηση των κωδίκων κρίνεται σκόπιμο να αφιερώσουμε μία ακόμη παράγραφο για να αναλύσουμε ορισμένα χαρακτηριστικά των Tanner Graphs από τα οποία κατασκευάζουμε LDPC κώδικες. Η μορφή που έχει ένα Tanner Graph σχετίζεται άμεσα με την πολυπλοκότητα της αποκωδικοποίησης του κώδικα για τον οποίο έχει σχεδιαστεί. Συγκεκριμένα οι LDPC κώδικες δουλεύουν βέλτιστα με αποκωδικοποίηση μέσω message passing algorithm. Ο Michael Tanner (εφευρέτης των Tanner Graphs) πρότεινε την αναπαράσταση των κωδικών ως bipartite graphs και την οπτικοποίηση της επαναληπτικής αποκωδικοποίησης ως έναν message passing αλγόριθμο σε ένα τέτοιο γράφημα. Θα εξηγήσουμε αργότερα πως δουλεύει αυτή. Η ουσία είναι ότι 'απλώνουμε' τα bits της κωδικολέξης στους κόμβους μεταβλητών (όπως αυτά αντιστοιχούν) και δια μέσου των ακμών που συνδέουν τους κόμβους μεταβλητών με τους κόμβους ελέγχου παίρνονται επαναληπτικά αποφάσεις για τα λάθος λαμβανόμενα bits (διαγραφές στο BEC). Αδυναμία αποκωδικοποίησης υπάρχει όταν μία διαγραφή που φτάνει σε έναν κόμβο ελέγχου δεν μπορεί να διορθωθεί. Αυτό είναι πολύ πιθανόν να συμβεί αν ένας κόμβος μεταβλητών και ένας κόμβος ελέγχου συνδέονται μεταξύ με μία ακμή και δεν υπάρχει καμία άλλη ακμή σε κανέναν από αυτούς τους κόμβους. Τότε λέμε ότι σχηματίζεται ένας κύκλος μήκους 2. Ανάλογες αδιέξοδοι μπορούν να συμβούν και μεταξύ περισσοτέρων κόμβων μεταβλητών και ελέγχου όπου προκύπτουν κύκλοι μεγαλύτερου μήκους. Γενικά, ένας εκλαϊκευμένος ορισμένος ενός κύκλου στο Tanner Graph είναι μια αλληλουχία ακμών που καταλήγει εκεί που αρχίζει. Ως εκ τούτου, θα ήταν εύλογο να υποθέσουμε ότι ένας καλός κώδικας δεν θα είχε κύκλους στο Tanner Graph. Παρόλα αυτά, έχει αποδειχθεί ότι Tanner Graphs χωρίς κύκλους (αναγόμαστε στην περίπτωση που ο γράφος αντιστοιχεί σε δένδρο) δεν έχουν βέλτιστη απόδοση. Έχει αποδειχθεί ότι τα λεγόμενα αυτά Cycle-Free Tanner Graphs αντιστοιχούν σε κώδικες με απόσταση $d \leq 2$ το οποίο συνεπάγεται κακό error correction capability. Συνεπώς η ύπαρξη κύκλων σε ένα Tanner Graph αποτελεί προϋπόθεση για να είναι εφικτή η αποκωδικοποίηση χωρίς σφάλματα. Μια παράμετρος που συνδέεται στενά με όσα συζητάμε σε αυτήν την παράγραφο είναι το girth του Tanner Graph. Ως girth ορίζεται το μήκους του μικρότερου κύκλου που υπάρχει στο Tanner Graph. Έχουν προταθεί διάφοροι αλγόριθμοι υπολογισμού του girth οι οποίοι όμως παρουσιάζουν υψηλή πολυπλοκότητα και είναι δύσκολο να υλοποιηθούν. Είναι σημαντικό όμως να διαλευκανθεί ότι ο έλεγχος του girth παίζει κομβικό ρόλο στην κατασκευή καλών κωδίκων. Τέλος αναφέρουμε απλά ότι μια μορφή LDPC κωδίκων που εμφανίζει πολύ υψηλές επιδόσεις, κοντά στο Shannon Capacity, σε πρακτικές εφαρμογές είναι οι Quasi-Cyclic LDPC κώδικες στους οποίους ο πίνακας

ελέγχου ισοτιμίας τους μπορεί να έχει τη μορφή ενός πίνακα μπλοκ που αποτελείται είτε από υποπίνακες αντιμετάθεσης κυκλικών μεταθέσεων (circulant permutation sub-matrices) είτε από τον μηδενικό υποπίνακα (zero submatrix).

Έχει ήδη αναφερθεί πολλαπλές φορές ότι οι LDPC κώδικες λειτουργούν βέλτιστα με χρήση επαναληπτικής αποκωδικοποίησης και συγκεκριμένα με χρήση message passing algorithm. Ο αλγόριθμος message passing που χρησιμοποιείται για αποκωδικοποίηση όταν χρησιμοποιούνται LDPC κώδικες είναι γνωστός ως belief propagation. Ένας belief propagation decoder δουλεύει ως εξής: προχωράμε σε γύρους επαναλήψεων (rounds of iterations) και κάθε γύρος ξεκινά με την επεξεργασία των εισερχόμενων μηνυμάτων στους κόμβους ελέγχου και στην συνέχεια τα προκύπτοντα εξερχόμενα μηνύματα στέλνονται στους κόμβους μεταβλητών κατά μήκος όλων των ακμών. Τα εισερχόμενα μηνύματα επεξεργάζονται στη συνέχεια στους κόμβους μεταβλητών και τα εξερχόμενα μηνύματα αποστέλλονται πίσω κατά μήκος όλων των ακμών στους κόμβους ελέγχου. Έτσι ολοκληρώνεται ένας γύρος της μεταβίβασης μηνυμάτων. Γενικά, η αποκωδικοποίηση αποτελείται από πολλούς τέτοιους γύρους. Θυμόμαστε ότι τα bits ενός εισερχόμενου μηνύματος αρχικά 'απλώνονται' στους κόμβους μεταβλητών που αντιστοιχούν βάσει του πίνακα $\boldsymbol{H}$. Αυτό οδηγεί συνηθώς στην τακτοποίηση των bits 'από κάτω μέχρι πάνω στους κόμβους μεταβλητών για συστηματικούς κώδικες. Ποιοι είναι όμως αυτοί οι κανόνες επεξεργασίας που τόσο αμυδρά αναφέραμε προηγουμένως; Περιγραφικά, σε έναν κόμβο μεταβλητών το εξερχόμενο μήνυμα είναι διαγραφή αν όλα τ α εισερχόμενα μηνύματα είναι διαγραφές. Ειδάλλως, αφού στο κανάλι ποτέ δεν λαμβάνονται λάθος σύμβολα, αλλά μόνο διαγραφές, όλα τα μηνύματα τα οποία δεν είναι διαγραφές πρέπει να είναι 0 ή 1. Σε έναν κόμβο ελέγχου, το εξερχόμενο μήνυμα είναι διαγραφή αν έστω ένα από τα εισερχόμενα μηνύματα είναι διαγραφές. Αλλιώς, αν όλα τα εισερχόμενα μηνύματα είναι 0 ή 1, τότε το εξερχόμενο μήνυμα είναι το $mod2$ άθροισμα όλων των εισερχόμενων μηνυμάτων. Έτσι δύναται η δυνατότητα σε κάθε γύρος του iterative decoding να διορθωθεί μια διαγραφή αν επιλεγεί σωστά ο κόμβος από τον οποίο θα ξεκινήσει η λήψη αποφάσεων. Παρακάτω φαίνεται σχηματικά η παραπάνω περιγραφική αναπαράσταση. Με διακεκομμένη γραμμή κάθε φορά φαίνεται ο κόμβος για τον οποίο παίρνεται η απόφαση με βάση τα υπόλοιπα μηνύματά που φτάνουν στον κόμβο τα οποία παρουσιάζονται με συνεχείς γραμμές στον κόμβο αυτό.



Περισσότερα για την αποκωδικοποίηση θα αναφέρουμε στο σημείο που θα μιλήσουμε για την πρακτική υλοποίηση αυτής μέσο Matlab

Σε προηγούμενη παράγραφο αναφέραμε ότι μπορούμε να κατασκευάσουμε καλούς τυχαίους κώδικες αν έχει προηγηθεί βελτιστοποίηση των βαθμών των κόμβων. Ας ξεκινήσουμε με κάποιους βασικούς ορισμούς και ορισμένες βασικές σχέσεις για irregular LDPC κώδικες πριν παρουσιάσουμε το τελικό πρόβλημα βελτιστοποίησης.

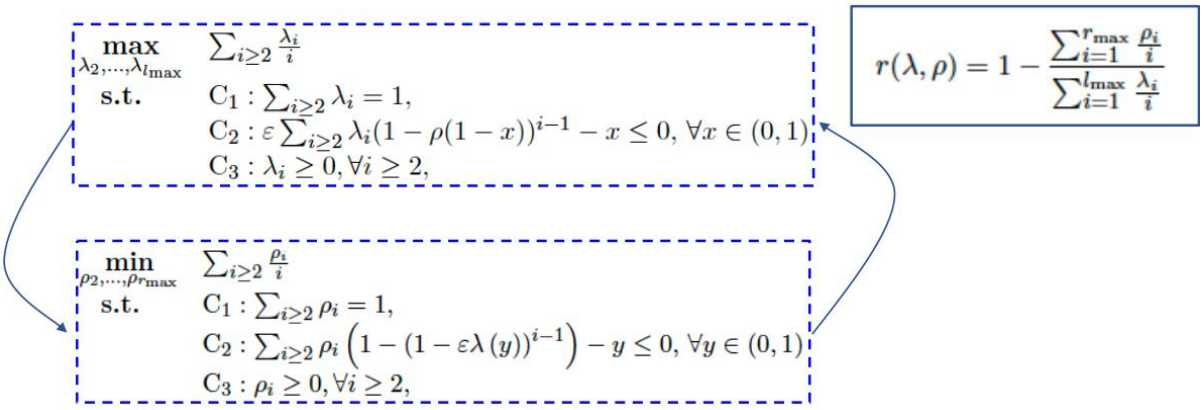
- Λ_i: αριθμός κόμβων μεταβλητών με βαθμό i
- P_i: αριθμός κόμβων ελέγχου με βαθμό i

- $\Lambda(x) = \sum_{i=1}^{l_{max}} \Lambda_i x_i$: το πολυώνυμο των βαθμών των κόμβων μεταβλητών
- $P(x) = \sum_{i=1}^{r_{max}} P_i x_i$: το πολυώνυμο των βαθμών των κόμβων ελέγχου
- Προφανώς $\sum_{i=1}^{l_{max}} \Lambda_i = n$ και $\sum_{\iota}^{r_{max}} P_i = n - \kappa$
- $\sum_{i=1}^{l_{max}} i\Lambda_i = \sum_{i=1}^{r_{max}} iP_i$
- $\lambda_i = \frac{i\Lambda_i}{\sum_j j\Lambda_j}$: Πιθανότητα μια τυχαία επιλεγμένη ακμή να συνδέεται σε έναν κόμβο μεταβλητών βαθμού i
- $\rho_i = \frac{iP_i}{\sum_j jP_j}$: Πιθανότητα μια τυχαία επιλεγμένη ακμή να συνδέεται σε έναν κόμβο ελέγχου βαθμού i
- $\lambda(x) = \sum_i \ \lambda_i x^{i-1} = \frac{\Lambda'(x)}{\Lambda'(1)}$: το πολυώνυμο πιθανότητας των κόμβων μεταβλητών
- $\rho(x) = \sum_i \ \rho_i x^{i-1} = \frac{P'(x)}{P'(1)}$: το πολυώνυμο πιθανότητας των κόμβων ελέγχου
- Προφανώς $\sum_i \ \lambda_i = \sum_i \rho_i = 1$
- $r(\lambda, \rho) = 1 - \frac{\sum_i P_i}{\sum_i \Lambda_i} = 1 - \frac{\frac{\sum_i P_i}{\sum_i iP_i}}{\frac{\sum_i \Lambda_i}{\sum_i i\Lambda_i}} = 1 - \frac{\sum_i \frac{\rho_i}{i}}{\sum_i \frac{\lambda_i}{i}}$: το design rate του irregular LDPC κώδικα
- x_l: πιθανότητα στην $l - o\sigma\tau\Psi$ επανάληψη το μήνυμα που φεύγει από έναν κόμβο μεταβλητών να είναι διαγραφή
- y_l: πιθανότητα στην $l - o\sigma\tau\Psi$ επανάληψη το μήνυμα που φεύγει από έναν κόμβο ελέγχου να είναι διαγραφή
- ε : πιθανότητα διαγραφής συμβόλου στο BEC

Με τους παραπάνω ορισμούς μπορούμε να με αναλυτικό τρόπο να υπολογίσουμε τις συνθήκες που πρέπει να ικανοποιούνται έτσι ώστε με το πέρας της κάθε επανάληψης να έχει μειωθεί η πιθανότητα να υφίσταται διαγραφή στα υπολειπόμενα σύμβολα έτσι ώστε να διασφαλίζεται η πρόοδος του decoding (σε αντίθετη περίπτωση ο αλγόριθμος θα εκτελεί επαναλήψεις χωρίς να διορθώνει διαγραφές). Συγκεκριμένα, καταλήγουμε ότι πρέπει να ισχύει:

$$\varepsilon\lambda\big(1 - \rho(1 - x_l)\big) < x_l \quad \kappa\alpha\iota \quad \sum_i \ \rho_i \left(1 - \left(1 - \varepsilon\lambda(y_l)\right)^{i-1}\right) < y_l$$

Με τους ορισμούς και περιορισμούς που έχουν τεθεί παραπάνω αποδεικνύεται ότι κωδικός ρυθμός μπορεί να βελτιστοποιηθεί λύνοντας τα παρακάτω 2 προβλήματα:



Αντί για το δεύτερο πρόβλημα μπορεί να γίνει επιλογή των ρ_i παραμέτρων βέλτιστα με βάση την εξίσωση: $\rho(x) = \frac{r(r+1-r_{avg})}{r_{avg}} x^{r-1} + \frac{r_{avg} - r(r+1-r_{avg})}{r_{avg}} x^r, r = \lfloor r_{avg} \rfloor$

Έτσι αρκεί να επιλυθεί μόνο το πρόβλημα 1 (βελτιστοποίηση των λ_i παραμέτρων) για διάφορες τιμές του r_{avg} (μέσος βαθμός κόμβων ελέγχου). Για την επίλυση του προβλήματος χρησιμοποιείται διακριτοποίηση του x ενώ ενδείκνυται και η χρήση του επιπλέον

περιορισμού $\lambda_2 \leq \frac{1}{\varepsilon\rho'(1)}$. Επίσης αναφέρουμε ότι ο λόγος που ξεκινάει ο ορισμός των παραμέτρων των κόμβων μεταβλητών/ελέγχου
από λ_2/ρ_2 συνάγει με όσα είπαμε σε προηγούμενη παράγραφο περί αποφυγής κόμβων με βαθμό 1. Η λύση του προβλήματος βελτιστοποίησης γίνεται με χρήση γραμμικού προγραμματισμού.
Έχοντας λύση το πρόβλημα αυτό έχουμε στην κατοχή μας τις βέλτιστες τιμές των πιθανοτήτων λ_i, ρ_i και του ρυθμού κώδικα. Το επόμενο πρόβλημα που πρέπει να λυθεί για να υλοποιηθεί ο κώδικας είναι να βρεθούν οι βέλτιστες ακέραιες τιμές Λ_i, P_i ώστε το Tanner Graph που θα προκύψει να αντιστοιχεί σε LDPC κώδικα. Επειδή η εύρεση των Λ_i, P_i με αναλυτικό τρόπο κατά πάσα πιθανότητα θα οδηγήσει σε κώδικα με ανεπιθύμητα μεγάλο μήκος κωδικολέξης n, ενδείκνυται μια προσεγγιστική μέθοδος έτσι ώστε για δεδομένο μήκος κωδικολέξης n να έχουμε LDPC κώδικα με όσο το δυνατόν καλύτερη προσέγγιση του κωδικού ρυθμού που επιθυμούμε. Η προσεγγιστική μέθοδος καταλήγει στο παρακάτω πρόβλημα βελτιστοποίησης:

$$\min_{\hat{x}_2^{\Lambda},\ldots,\hat{x}_{l_{\max}}^{\Lambda},\hat{x}_2^{P},\ldots,\hat{x}_{r_{\max}}^{P}} \left(\sum_{i=2}^{r_{\max}} \hat{x}_i^{P} - A\right)^2$$
$$\text{s.t.}$$
$$\mathrm{C}_1 : \sum_{i=2}^{l_{\max}} \hat{x}_i^{\Lambda} = n - \sum_{i=2}^{l_{\max}} \hat{\Lambda}_i,$$
$$\mathrm{C}_2 : \sum_{i=2}^{l_{\max}} i\hat{x}_i^{\Lambda} - \sum_{i=2}^{r_{\max}} i\hat{x}_i^{P} = \sum_{i=2}^{r_{\max}} i\hat{P}_i - \sum_{i=2}^{l_{\max}} i\hat{\Lambda}_i,$$
$$\mathrm{C}_3 : \hat{x}_i^{\Lambda} \in \{0,1\}, \hat{x}_i^{P} \in \{0,1\}$$

όπου $A = \sum_{i=2}^{r_{max}} \left(P_i - \hat{P}_i\right)$, $\hat{\Lambda}_i = \Lambda_i - {x_i}^{\Lambda} = \left\lfloor \frac{\frac{\lambda_i}{i}}{\sum_j \frac{\lambda_j}{j}} n \right\rfloor$ και $\hat{P}_i = P_i - {x_i}^{P} = \left\lfloor \frac{\frac{\rho_i}{i}}{\sum_j \frac{\rho_j}{j}} n \right\rfloor$.

Μπορούμε να διευκολύνουμε την επίλυση του παραπάνω προβλήματος σπάζοντας το σε δύο επιμέρους προβλήματα:

$$\boxed{\sum_{i=2}^{r_{\max}} \hat{x}_i^{P} - A \geq 0} \quad \min_{\hat{x}_2^{\Lambda},\ldots,\hat{x}_{l_{\max}}^{\Lambda},\hat{x}_2^{P},\ldots,\hat{x}_{r_{\max}}^{P}} \sum_{i=2}^{r_{\max}} \hat{x}_i^{P}$$
$$\text{s.t.}$$
$$\mathrm{C}_1 : \sum_{i=2}^{l_{\max}} \hat{x}_i^{\Lambda} = n - \sum_{i=2}^{l_{\max}} \hat{\Lambda}_i,$$
$$\mathrm{C}_2 : \sum_{i=2}^{l_{\max}} i\hat{x}_i^{\Lambda} - \sum_{i=2}^{r_{\max}} i\hat{x}_i^{P} = \sum_{i=2}^{r_{\max}} i\hat{P}_i - \sum_{i=2}^{l_{\max}} i\hat{\Lambda}_i,$$
$$\mathrm{C}_3 : \hat{x}_i^{\Lambda} \in \{0,1\}, \hat{x}_i^{P} \in \{0,1\},$$
$$\mathrm{C}_4 : \sum_{i=2}^{r_{\max}} \hat{x}_i^{P} \geq \lceil A \rceil.$$

$$\boxed{\sum_{i=2}^{r_{\max}} \hat{x}_i^{P} - A \leq 0} \quad \max_{\hat{x}_2^{\Lambda},\ldots,\hat{x}_{l_{\max}}^{\Lambda},\hat{x}_2^{P},\ldots,\hat{x}_{r_{\max}}^{P}} \sum_{i=2}^{r_{\max}} \hat{x}_i^{P}$$
$$\text{s.t.}$$
$$\mathrm{C}_1 : \sum_{i=2}^{l_{\max}} \hat{x}_i^{\Lambda} = n - \sum_{i=2}^{l_{\max}} \hat{\Lambda}_i,$$
$$\mathrm{C}_2 : \sum_{i=2}^{l_{\max}} i\hat{x}_i^{\Lambda} - \sum_{i=2}^{r_{\max}} i\hat{x}_i^{P} = \sum_{i=2}^{r_{\max}} i\hat{P}_i - \sum_{i=2}^{l_{\max}} i\hat{\Lambda}_i,$$
$$\mathrm{C}_3 : \hat{x}_i^{\Lambda} \in \{0,1\}, \hat{x}_i^{P} \in \{0,1\},$$
$$\mathrm{C}_4 : \sum_{i=2}^{r_{\max}} \hat{x}_i^{P} \leq \lfloor A \rfloor.$$

και κρατώντας την βέλτιστη εκ των δύο λύσεων. Ύστερα, έχοντας τις εκτιμήσεις των ${x_i}^{\Lambda}$ και ${x_i}^{P}$, υπολογίζουμε τα διορθωμένα Λ_i και P_i. Η λύση επιτυγχάνεται με χρήση γραμμικού προγραμματισμού ακεραίων.

Κώδικας Matlab

Για την προσομοίωση των LDPC, δημιουργήθηκε ένα σύνολο από συναρτήσεις, που η κάθε μία υλοποιεί ένα βήμα της θεωρίας. Συνολικά 4 συναρτήσεις και 2 script στα οποίο παρουσιάζεται η χρήση τους και με το οποίο λήφθηκαν τα αποτελέσματα που θα ακολουθήσουν στην αντίστοιχη ενότητα.

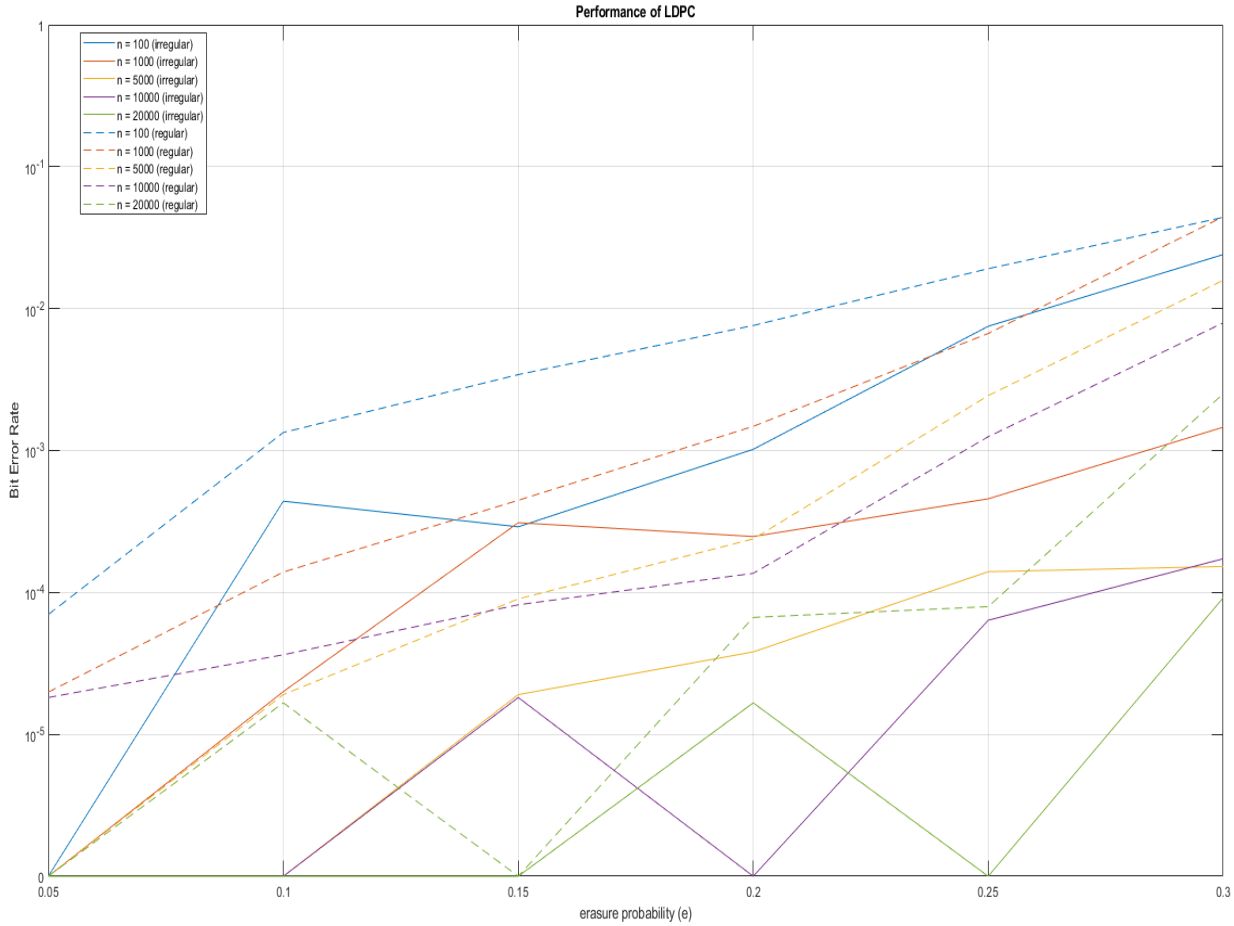
Η πρώτη συνάρτηση (*bp_dec*) υλοποιεί έναν belief propagation decoder. Στην είσοδο δέχεται τα κωδικοποιημένα δεδομένα, με κάθε λέξη σε μία σειρά ενός δισδιάστατου πίνακα, καθώς και τον πίνακα ελέγχου ισοτιμίας **H** του κώδικα. Ως μέγιστος αριθμός επαναλήψεων υπολογίζεται ο μέγιστος αριθμός των διαγραφών που αναμένεται να έχει η κάθε λέξη. Αυτό έχει ως αποτέλεσμα ναι μεν να γίνεται αρκετά αργός ο αλγόριθμος για σχετικά μεγάλα e και n, αλλά οδηγεί σε καλή εκτίμηση της ικανότητας διόρθωσης λαθών του κώδικα. Για μείωση των επαναλήψεων μπορεί να οριστεί e = 0, στην είσοδο της συνάρτησης. Επίσης, σε περίπτωση που κατά την διάρκεια μιας επανάληψης δεν διορθωθεί κάποια διαγραφή, η συνάρτηση σταματάει εκεί για την δεδομένη λέξη. Αυτό μπορεί να συμβεί π.χ. σε περίπτωση που υπάρχει ένας μικρός κύκλος (girth = 4) στον H και οι δύο κόμβοι μεταβλητών είναι και οι 2 διαγραφές. Τέλος, επιλογή μας ήταν, η συνάρτηση να επιστρέφει τις διορθωμένες κωδικολέξεις μαζί με τις διαγραφές, αν και θεωρητικά θα μπορούσε να μαντεύει και να ρίξει το ber στο μισό.

Η δεύτερη συνάρτηση (*poly2mat*) ουσιαστικά δημιουργεί τον πίνακα ελέγχου ισοτιμίας ενός LDPC κώδικα (είτε regular είτε irregular) με δεδομένα τα πολυώνυμα του και το μέγεθος κωδικολέξης n. Ανακατεύει τους κόμβους και τις ακμές του γράφου, κι έπειτα τα αντιστοιχίζει. Ουσιαστικά χρησιμοποιεί ομοιόμορφη κατανομή. Σε γενικές γραμμές δεν είναι αποδοτική, διότι δημιουργεί πολλούς κύκλους μήκους 4 ακόμα και σε irregular. Ωστόσο, λειτουργεί ικανοποιητικά.

Η τρίτη συνάρτηση (*li_ri_opt*) υλοποιεί το πρώτο βήμα της βελτιστοποίησης του πολυωνύμου ενός irregular LDPC. Ενώ η τέταρτη (*Li_Ri_approx_opt_v1_2*) το δεύτερο. Ο κώδικας είναι επαρκώς σχολιασμένος.

Αποτελέσματα

Για το πείραμα, δημιουργήθηκαν τα scripts LDPCs.m και LDPCs_r.m, που χρησιμοποιούν τις παραπάνω συναρτήσεις. Για να εξάγουμε αποτελέσματα για n = 100,1000,5000,10000,20000 για έναν βέλτιστο irregular κι έναν ισάξιο (από άποψη ρυθμού κώδικα) regular. Επιλέξαμε τον βέλτιστο κώδικα με lmax = 6 και rmax = 8 και ρυθμό κώδικα 0.604, έτσι ώστε να είναι εύκολο να έχουμε τον ίδιο ρυθμό κώδικα με έναν regular-LDPC(2,5). Θα μπορούσαμε να διαλέξουμε πιο μεγάλα rmax και lmax, αλλά με αυτόν τον γίνεται πολύ γρήγορη η διαδικασία της βελτιστοποίησης. Ο κώδικας είχε design rate 0.66, επομένως η προσομοίωση έγινε για e < 0.33. Σε κάθε περίπτωση χρησιμοποιήθηκαν 100000 bit πληροφορίας, επομένως η στάθμη 0 τίθεται για 10^{-6}.



Στο διάγραμμα φαίνεται οι η μεταβολή του BER συναρτήσει της πιθανότητας διαγραφής e για διάφορες τιμές του μήκους της κωδικής λέξης n. Πρωτίστως, είναι εμφανής η αναμενόμενη μείωση του BER όσο αυξάνει το μήκος κωδικολέξης. Όπως έχει αναφερθεί και παραπάνω, οι LDPC κώδικες δουλεύουν βέλτιστα (κοντά στο capacity του συστήματος) για πολύ μεγάλα μήκη κωδικής λέξης. Περαιτέρω, βλέπουμε ότι οι συναρτήσεις για σταθερό n είναι γενικά αύξουσες (αύξηση του e συνεπάγεται αύξηση του BER). Παρατηρούνται διαστήματα που φαίνεται να μειώνεται ελάχιστα ο ρυθμός σφάλματος καθώς επίσης και τυχαίοι μηδενισμοί του BER για μικρές πιθανότητες σφάλματος οι οποίες οφείλονται στο γεγονός ότι δεν αφιερώσαμε επαρκώς μεγάλο στατιστικό δείγμα στο simulation που εφαρμόσαμε. Με περισσότερες (και ιδιαίτερα χρονοβόρες) επαναλήψεις των αλγορίθμων αναμένεται εξομάλυνση των συναρτήσεων, να γίνουν αύξουσες. Συγκρίνοντας τους irregular κώδικες (συνεχείς γραμμές) με τους regular (διακεκομμένες) για ίδιο μήκος κωδικής λέξης παρατηρούμε ότι οι irregular λειτουργούν καλύτερα ως προς το error correction capability από τους regular. Η διαφορά μεταξύ των BER των regular και irregular ldpc κωδίκων μεγαλώνει με αύξηση της πιθανότητας σφάλματος για τις τιμές που έχουμε διαθέσιμες. Με βελτιστοποίηση των παραμέτρων αυτών για πιο υψηλές τιμές των επιλεγόμενων r_{max}, l_{max} θα μπορούσαμε να παράγουμε κώδικες πιο υψηλής αποδοτικότητας. Κάτι τέτοιο θα απαιτούσε την αύξηση των επαναλήψεων στο πρόβλημα βελτιστοποίησης παραμέτρων που ορίσαμε και συνεπώς εκθετική αύξηση του χρόνου εκπλήρωσης του προγράμματος. Εν κατακλείδι, στην υλοποίηση μας οι irregular ldpc κώδικες είναι πιο αποδοτικοί από τους regular, όπως θα περιμέναμε και από την θεωρία.

Βιβλιογραφία

1. A Mathematical Theory of Communication, Claude E. Shannon
2. Essentials of Error-Control Coding, Jorge C. Moreira, Patrick G. Farrell
3. Communication Systems Engineering, John G. Proakis, Masoud Salehi
4. Wireless Communications, Andrea Goldsmith
5. Modern Coding Theory, Thomas J. Richardson, Rudiger L. Urbanke
6. Error Correction Coding, Todd K. Moon

7. Information Theory, Inference, and Learning Algorithms, David J.C. McKay
8. MIT lectures on Linear Block Codes: Encoding and Syndrome Decoding
9. The capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding, Thomas J. Richardson, Rudiger L. Urbanke (paper)
10. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes, Thomas J. Richardson, Rudiger L. Urbanke (paper)
11. On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit, Thomas J. Richardson, Rudiger L. Urbanke (paper)
12. Probabilistic Analysis of Cycles in Random Tanner Graphs, Xiaopeng Jiao, Jianjun Mu (paper)
13. Which Codes Have Cycle-Free Tanner Graphs?, Tuvi Etzion, Ari Trachtenberg, Alexander Vardy (paper)
14. Generating Random Tanner Graphs with Large Girth, Mohsen Bayati, Raghunandan Keshavan, Andrea Montanari, Sewoong Oh, Amin Saberi (paper)
15. Σημειώσεις μαθήματος