ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

Τεχνικές κωδικοποίησης καναλιού σε τηλεπικοινωνιακά συστήματα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Άρτεμις Βάσσιου (ΑΜ: 3121256)

Ελένη Παρλίτση (ΑΜ: 3121154)

Επιβλέπων: ***Κωνσταντίνος Χαϊκάλης ,Αναπληρωτής Καθηγητής***

ΛΑΡΙΣΑ, Μαΐος 2025

UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

Channel coding techniques in telecommunications systems

BACHELOR THESIS

Artemis Vassiou (RN: 3121256)

Eleni Parlitsi (RN: 3121154)

Supervisor: ***Konstantinos Xaikalis , position***

LARISSA, May 2025

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

Ονοματεπώνυμο Παρλίτση Ελένη – Άρτεμις Βάσσιου

Ημερομηνία 10/10/2025

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα	**Κωνσταντίνος *Χαϊκάλης***
	Αναπληρωτής Καθηγητής
	Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	**Όμηρος Ιατρέλης**
	Επίκουρος Καθηγητής,
	Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	**Απόστολος Ξενάκης**
	Επίκουρος Καθηγητής ,
	Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας

Περίληψη

Στην παρούσα πτυχιακή εργασία μελετώνται οι τεχνικές κωδικοποίησης καναλιού σε τηλεπικοινωνιακά συστήματα, με στόχο την κατανόηση της σημασίας τους στη βελτίωση της αξιοπιστίας και της αποδοτικότητας της μετάδοσης δεδομένων σε θορυβώδη κανάλια. Η έρευνα εστιάζει σε θεμελιώδεις έννοιες της θεωρίας πληροφορίας και της διόρθωσης σφαλμάτων , οι οποίες παρουσιάζονται μέσα από τέσσερα βασικά κεφάλαια του συγγράμματος ***Essentials of Error Control Coding***: Εντροπία, Γραμμικοί block κώδικες , κυκλικοί κώδικες και Συνελκτικοί κώδικες.

Η μελέτη ξεκινά με την έννοια της εντροπίας η οποία χρησιμοποιείται ως μέτρο πληροφορίας και στοχαστικής αβεβαιότητας, και συνεχίζει με την ανάλυση block κωδίκων , δίνοντας έμφαση στους πίνακες γεννήτριας και ελέγχου καθώς και σε χαρακτηριστικές κατηγορίες όπως οι κώδικες Hamming. Στη συνέχεια εξετάζονται οι κυκλικοί κώδικες , με έμφαση στην πολυωνυμική θεμελίωση και στις αλγεβρικές τεχνικές κωδικοποίησης και αποκωδικοποίησης , ενώ το θεωρητικό μέρος ολοκληρώνεται με τους συνελκτικούς κώδικες.

Η εργασία δεν περιορίστηκε στη θεωρητική ανάλυση, αλλά εμπλουτίστηκε με ασκήσεις οι οποίες υλοποιήθηκαν σε περιβάλλον MATLAB. Μέσω των προσομοιώσεων έγινε δυνατή η αξιολόγιση της απόδοσης διαφορετικών τεχνικών κωδικοποίησης , με βασικό κριτήριο τον ρυθμό εμφάνισης σφαλμάτων.

Τα αποτελέσματα καταδεικνύουν ότι η χρήση κατάλληλων τεχνικών κωδικοποίησης μπορεί να μειώσει σημαντικά την πιθανότητα σφαλμάτων και να βελτιώσει την αξιοπιστία της επικοινωνίας ακόμη και σε περιβάλλοντα με έντονο θόρυβο. Συνολικά, η εργασία αναδεικνύει τον ρόλο της κωδικοποίησης καναλιού ως ακρογωνιαίο στοιχείο στη σχεδίαση και ανάπτυξη σύγχρονων τηλεπικοινωνιακών συστημάτων, συνδυάζοντας θεωρητική γνώση με πρακτική εφαρμογή.

Abstract

This thesis investigates channel coding techniques in telecommunication systems, aiming to highlight their importance in improving the reliability and efficiency of data transmission over noisy channels. The study focuses on fundamental concepts of information theory and error control coding, presented through four core chapters of the **Essentials of Error Control Coding** textbook: Entropy, Linear Block Codes, Cyclic Codes, and Convolutional Codes.

The work begins with the concept of entropy, used as a measure of information and stochastic uncertainty, and continues with the analysis of linear block codes, emphasizing generator and parity-check matrices as well as characteristic categories such as Hamming codes. It then examines cyclic codes, with emphasis on their polynomial foundation and algebraic encoding/decoding techniques, while the theoretical part concludes with convolutional codes and the Viterbi algorithm as the optimal decoding method.

Beyond the theoretical analysis, the thesis is enriched with exercises and applications implemented in MATLAB. Through these simulations, the performance of different coding techniques was evaluated, with bit error rate (BER) as the main performance criterion.

The results demonstrate that the appropriate choice of coding technique can significantly reduce the probability of errors and improve communication reliability, even in noisy environments. Overall, the thesis highlights the role of channel coding as a cornerstone in the design and development of modern telecommunication systems, combining theoretical foundation with practical application.

-x-

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα κ. Κωνσταντίνο Χαϊκάλη για την καθοδήγηση και υποστήριξή του, καθώς και τις οικογένειές μας για την αδιάκοπη βοήθειά τους.

Ονοματεπώνυμο Παρλίτση Ελένη – Άρτεμις Βάσσιου

Ημερομηνία 10/10/2025

Περιεχόμενα

Εισαγωγή

Η εξέλιξη των τηλεπικοινωνιακών συστημάτων αποτελεί ένα από τα σημαντικότερα επιτεύγματα της σύγχρονης τεχνολογίας, ασκώντας καθοριστική επίδραση σε πλήθος τομέων της ανθρώπινης δραστηριότητας. Η αξιόπιστη και αποδοτική μετάδοση πληροφοριών αποτελεί θεμελιώδη προϋπόθεση για τη λειτουργία εφαρμογών όπως τα ασύρματα δίκτυα, η δορυφορική επικοινωνία, τα δίκτυα κινητής τηλεφωνίας, καθώς και η ανταλλαγή δεδομένων μέσω του διαδικτύου. Παρά τις ραγδαίες τεχνολογικές εξελίξεις, τα κανάλια μετάδοσης εξακολουθούν να υπόκεινται σε περιορισμούς που οφείλονται σε φυσικά φαινόμενα, όπως ο θόρυβος, οι παρεμβολές και οι απώλειες σήματος. Οι παράγοντες αυτοί μπορούν να προκαλέσουν αλλοιώσεις κατά τη διαδικασία μετάδοσης, καθιστώντας αναγκαία την εφαρμογή τεχνικών ανίχνευσης και διόρθωσης σφαλμάτων.

Η παρούσα πτυχιακή εργασία επικεντρώνεται στη μελέτη των τεχνικών κωδικοποίησης καναλιού και στον ρόλο τους στη διασφάλιση της αξιοπιστίας των τηλεπικοινωνιακών συστημάτων. Κύριος στόχος είναι η παρουσίαση των θεμελιωδών εννοιών της θεωρίας πληροφορίας και των μεθόδων διόρθωσης σφαλμάτων, καθώς και η ανάλυση του τρόπου με τον οποίο οι διάφορες κατηγορίες κωδίκων συμβάλλουν στην αποτελεσματική και ασφαλή μετάδοση δεδομένων. Η εργασία απευθύνεται τόσο σε αναγνώστες με τεχνικό υπόβαθρο όσο και σε όσους επιθυμούν να αποκτήσουν μια ολοκληρωμένη εικόνα του αντικειμένου, χωρίς να απαιτείται εξειδικευμένη γνώση.

Η μεθοδολογία που ακολουθήθηκε περιλαμβάνει αρχικά τη θεωρητική ανάλυση βασικών εννοιών της θεωρίας πληροφορίας, όπως η εντροπία και η χωρητικότητα καναλιού, οι οποίες αποτελούν το θεωρητικό υπόβαθρο για τη μελέτη των ορίων της αξιόπιστης επικοινωνίας. Στη συνέχεια, παρουσιάζονται και αναλύονται επιλεγμένες τεχνικές κωδικοποίησης καναλιού, όπως οι γραμμικοί block κώδικες, οι κυκλικοί κώδικες και οι συνελικτικοί κώδικες, με σκοπό την κατανόηση των αρχών λειτουργίας τους, των μαθηματικών εργαλείων που τις υποστηρίζουν, καθώς και των εφαρμογών τους σε πραγματικά τηλεπικοινωνιακά συστήματα.

Τέλος, εξετάζεται η συμβολή της κωδικοποίησης καναλιού στη βελτιστοποίηση της χρήσης των διαθέσιμων πόρων ενός συστήματος, όπως το φάσμα συχνοτήτων και η ισχύς μετάδοσης, αναδεικνύοντας τη σύνδεση της θεωρητικής προσέγγισης με την πρακτική εφαρμογή. Η εργασία συνδυάζει θεωρητική ανάλυση και πειραματική προσομοίωση μέσω MATLAB, με στόχο την εμπεριστατωμένη κατανόηση και επαλήθευση των θεωρητικών αρχών στην πράξη. Μέσω αυτής της συνολικής ανάλυσης, υπογραμμίζεται η σημασία των τεχνικών κωδικοποίησης στην ανάπτυξη αποδοτικών και ανθεκτικών τηλεπικοινωνιακών συστημάτων, ικανών να ανταποκριθούν στις απαιτήσεις της σύγχρονης ψηφιακής εποχή

Κεφάλαιο 1 – Θεωρία Πληροφορίας

1.1 Εισαγωγή στη Θεωρία Πληροφορίας

Η θεωρία πληροφορίας, θεμελιωμένη από τον Claude Shannon το 1948, παρέχει το μαθηματικό πλαίσιο για την ανάλυση, αναπαράσταση και αξιόπιστη μετάδοση της πληροφορίας. Εξετάζει πώς η πληροφορία μετριέται, πώς συμπιέζεται χωρίς απώλειες και ποια είναι τα όρια της μετάδοσης μέσω καναλιών που υπόκεινται σε θόρυβο.

Οι δύο βασικές έννοιες που εισάγει είναι η εντροπία, ως μέτρο της αβεβαιότητας ή της μέσης πληροφορίας που παράγει μία πηγή, και η χωρητικότητα καναλιού, ως μέτρο της μέγιστης ποσότητας πληροφορίας που μπορεί να μεταδοθεί αξιόπιστα μέσω ενός δεδομένου καναλιού επικοινωνίας.

1.2 Έννοια της Εντροπίας

Γενικά, μια πηγή πληροφορίας παράγει ένα από ένα σύνολο M διαφορετικών συμβόλων, τα οποία θεωρούνται ως τιμές μιας διακριτής τυχαίας μεταβλητής X, που μπορεί να πάρει τιμές από το σύνολο A, το οποίο περιλαμβάνει τα σύμβολα x1, x2, έως και xM. Κάθε σύμβολο xi έχει πιθανότητα εμφάνισης Pi και περιέχει πληροφορία Ii. Οι πιθανότητες των συμβόλων πρέπει να πληρούν την προϋπόθεση ότι κάποιο από τα σύμβολα σίγουρα θα εκπέμπεται, επομένως το άθροισμα των πιθανοτήτων τους ισούται με τη μονάδα.

Η κατανομή πιθανοτήτων των συμβόλων της πηγής θεωρείται σταθερή (στατιστικά αμετάβλητη), και τα σύμβολα είναι ανεξάρτητα και μεταδίδονται με ρυθμό r συμβόλων ανά δευτερόλεπτο. Αυτή η περιγραφή αντιστοιχεί σε μία διακριτή πηγή χωρίς μνήμη (Discrete Memoryless Source - DMS).

Καθένα από τα σύμβολα περιέχει πληροφορία Ii, οπότε το σύνολο των τιμών {I1, I2, ..., IM} μπορεί να θεωρηθεί ως μια διακριτή τυχαία μεταβλητή, και ο μέσος όρος των πληροφοριών αυτών καθορίζει το μέτρο εντροπίας της πηγής, το οποίο εκφράζει τη μέση ποσότητα πληροφορίας ανά σύμβολο.

Η συνάρτηση αυτή ονομάζεται εντροπία της πηγής. Όταν ο υπολογισμός γίνεται με λογαρίθμους βάσης 2, η εντροπία μετράται σε bits ανά σύμβολο.

Αν η πιθανότητα P_i ενός συμβόλου είναι 0, τότε η ποσότητα $-P_i \cdot \log_2 P_i$ ορίζεται ίση με 0 (σύμβαση $0 \cdot \log 0 = 0$). Έτσι η εντροπία υπολογίζεται κανονικά.Αντίστοιχα, όταν η πιθανότητα Pi είναι ίση με 1, τότε το αντίστοιχο γινόμενο είναι επίσης μηδέν.

Παράδειγμα 2.1: Έστω μια πηγή χωρίς μνήμη που εκπέμπει τέσσερα σύμβολα x1, x2, x3 και x4, με πιθανότητες εμφάνισης αντίστοιχα 1/2, 1/8, 1/8 και 1/4. Η εντροπία αυτής της πηγής υπολογίζεται ως εξής:

$$\frac{1}{2} log_2(2) + \frac{1}{8} log_2(8)\ δηλαδη\ η\ εντροπια\ ειναι\ 1.75\ bits\ ανά\ σύμβολο.$$

$\frac{1}{4} log_2(4)$ δηλαδή η εντροπία είναι 1,75 bits ανά σύμβολο.

Παράδειγμα 2.2: Μια πηγή που χαρακτηρίζεται στο πεδίο της συχνότητας έχει εύρος ζώνης 4000 Hz και δειγματοληπτείται με το ρυθμό Nyquist, παράγοντας τιμές από το σύνολο {-2, -1, 0, 1, 2}, με αντίστοιχες πιθανότητες 1/2, 1/4, 1/8, 1/16 και 1/16. Η εντροπία της πηγής υπολογίζεται ως 15/8 bits ανά δείγμα. Δεδομένου ότι ο ρυθμός δειγματοληψίας είναι 8000 δείγματα ανά δευτερόλεπτο, ο ρυθμός πληροφορίας είναι 15.000 bits ανά δευτερόλεπτο (15 kbps).

Η εντροπία μπορεί επίσης να υπολογιστεί με χρήση λογαρίθμων διαφορετικής βάσης, και τότε προσαρμόζεται ανάλογα.

Η εντροπία H(X) μπορεί να ερμηνευτεί ως ο μέσος όρος της πληροφορίας ανά σύμβολο που παρέχει η πηγή ή ως η αναμενόμενη πληροφορία πριν την παρατήρηση του αποτελέσματος. Αποτελεί επίσης μέτρο του βαθμού αβεβαιότητας ή τυχαιότητας της πηγής. Με λίγα λόγια, η εντροπία αποτελεί ακριβές ποσοτικό μέτρο της ποσότητας πληροφορίας που εκπέμπει μία πηγή.

Μια ακόμη ερμηνεία της εντροπίας προκύπτει αν θεωρήσουμε ότι όταν εκπέμπονται n συμβολή (με το n να είναι πολύ μεγάλο), τότε το σύνολο της πληροφορίας είναι περίπου ίσο με n επί H(X) bits. Αν η πηγή εκπέμπει με ρυθμό r συμβόλων ανά δευτερόλεπτο, τότε ο συνολικός ρυθμός πληροφορίας είναι ίσος με r επί H(X) bits ανά δευτερόλεπτο.

Το θεώρημα του Shannon αναφέρει ότι η πληροφορία που παράγεται από μία διακριτή πηγή χωρίς μνήμη μπορεί να κωδικοποιηθεί σε δυαδικά ψηφία και να μεταδοθεί μέσω ενός ιδανικού (χωρίς θόρυβο) καναλιού, με ρυθμό τουλάχιστον ίσο με τον ρυθμό πληροφορίας R.

Επισημαίνεται ξανά ότι το bit αποτελεί μονάδα πληροφορίας, ενώ το δυαδικό ψηφίο (0 ή 1) είναι ένα σύμβολο που χρησιμοποιείται στην αναπαράσταση αυτής της πληροφορίας.

Θεώρημα 2.1: Αν X είναι μια τυχαία μεταβλητή που λαμβάνει τιμές από ένα σύνολο A, τότε ισχύει ότι η εντροπία H(X) είναι πάντα μεγαλύτερη ή ίση με το μηδέν και μικρότερη ή ίση με $log2(M)$, όπου M είναι το πλήθος των διαφορετικών συμβόλων.

Η εντροπία είναι μηδέν όταν η πιθανότητα ενός συμβόλου είναι 1 και όλες οι άλλες είναι μηδέν. Αντίθετα, η εντροπία φτάνει το μέγιστο $log2(M)$ όταν όλα τα σύμβολα έχουν ίσες πιθανότητες, δηλαδή ομοιόμορφη κατανομή.

Στην περίπτωση δυαδικής πηγής (όπου M = 2), αν τα δύο σύμβολα έχουν πιθανότητες $P0 = \alpha$ και $P1 = 1 - \alpha$, τότε η εντροπία H(X) είναι συνάρτηση του α και παίρνει τη μέγιστη τιμή της όταν $\alpha = 1/2$, δηλαδή όταν τα δύο σύμβολα είναι ισοπίθανα. Τότε, η εντροπία είναι 1 bit ανά σύμβολο.

Παράδειγμα 2.3: Μια πηγή εκπέμπει 3000 σύμβολα ανά δευτερόλεπτο από ένα σύνολο τεσσάρων συμβόλων, με πιθανότητες 1/3, 1/3, 1/6 και 1/6. Η αντίστοιχη πληροφορία ανά σύμβολο είναι περίπου 1.5849 bits για τα πρώτα δύο και 2.5849 bits για τα άλλα δύο. Η εντροπία της πηγής είναι περίπου 1.9183 bits ανά σύμβολο, πολύ κοντά στη μέγιστη τιμή $log2(4)$ = 2 bits ανά σύμβολο. Ο ρυθμός πληροφορίας είναι ίσος με 3000 επί 1.9183, δηλαδή περίπου 5754.9 bits ανά δευτερόλεπτο.

Επεκταμένες Διακριτές Μνήμης Πηγές (Extended DMS)

Σε πολλές περιπτώσεις είναι χρήσιμο να αναλύουμε την πληροφορία όχι μόνο σε μεμονωμένα σύμβολα, αλλά σε ομάδες ή ακολουθίες από σύμβολα. Αυτή η προσέγγιση είναι ιδιαίτερα χρήσιμη όταν τα σύμβολα της πηγής μεταδίδονται διαδοχικά. Η

διαδικασία κατά την οποία σχηματίζουμε ακολουθίες από διαδοχικά σύμβολα μιας διακριτής πηγής ονομάζεται επέκταση της πηγής.

Αν έχουμε μια πηγή που εκπέμπει M διαφορετικά σύμβολα, τότε η επέκταση της πηγής κατά n θέσεις δημιουργεί μια νέα πηγή η οποία παράγει ακολουθίες μήκους n. Η νέα πηγή έχει συνολικά M^n δυνατούς συνδυασμούς (ή αλλιώς σύμβολα). Για παράδειγμα, μια πηγή με 4 σύμβολα, όταν επεκταθεί κατά δύο θέσεις, παράγει 16 συνδυασμούς (4x4).

Η πιθανότητα εμφάνισης κάθε συνδυασμού υπολογίζεται ως το γινόμενο των επιμέρους πιθανοτήτων των συμβόλων, εφόσον αυτά είναι ανεξάρτητα μεταξύ τους. Η εντροπία της επεκταμένης πηγής εκφράζει τον μέσο αριθμό bit που απαιτούνται για την κωδικοποίηση κάθε ακολουθίας και δίνεται από το άθροισμα:

$H(x^n) = \sum_i p(y_i) \log_2 \left(\frac{1}{p(y_i)} \right)$ Όπου y_i είναι η κάθε ακολουθία μήκους n, και $p(y_i)$ η αντίστοιχη πιθανότητα.

Σημαντικό συμπέρασμα: Αν η αρχική πηγή έχει εντροπία H(X), τότε η εντροπία της επεκταμένης πηγής είναι $H(x^n) = n * H(X)$, δηλαδή αυξάνεται γραμμικά με το πλήθος των συμβόλων στην ακολουθία.

1.3 Κανάλια και Αμοιβαία Πληροφορία

1.3.1 Μετάδοση Πληροφορίας μέσω Διακριτών Καναλιών

Μέχρι αυτό το σημείο η μελέτη της πληροφορίας περιοριζόταν στην πηγή. Στην πράξη όμως, η πληροφορία μεταδίδεται από έναν πομπό σε έναν δέκτη μέσω ενός καναλιού επικοινωνίας, το οποίο ενδέχεται να εισάγει αλλοιώσεις (θόρυβο) κατά τη μετάδοση.

Ο σκοπός αυτής της ενότητας είναι να μελετηθεί η επίδραση του καναλιού στη μετάδοση της πληροφορίας. Ένα κανάλι θεωρείται αναξιόπιστο όταν υπάρχει πιθανότητα το ληφθέν σύμβολο να είναι διαφορετικό από το μεταδιδόμενο. Ο βαθμός αξιοπιστίας του καναλιού επηρεάζει άμεσα την ικανότητα ανάκτησης της αρχικής πληροφορίας από τον δέκτη.

1.3.2 Μοντελοποίηση Πληροφοριακών Καναλιών

Ένα διακριτό πληροφοριακό κανάλι περιγράφεται από:

Ένα σύνολο εισόδων (π.χ. x1, x2, ..., xU),

Ένα σύνολο εξόδων (π.χ. y1, y2, ..., yV),

Ένα πίνακα πιθανοτήτων $p\left(\frac{y_i}{xi}\right)$ ο οποίος εκφράζει την πιθανότητα να ληφθεί το σύμβολο yj εάν έχει σταλεί το σύμβολο xi.

Ο πίνακας αυτός, γνωστός ως πίνακας μετάβασης καναλιού, έχει τόσες γραμμές όσες είναι οι πιθανές είσοδοι και τόσες στήλες όσες είναι οι δυνατές έξοδοι. Το άθροισμα των πιθανοτήτων κάθε γραμμής ισούται με 1, καθώς μετά από κάθε είσοδο πρέπει υποχρεωτικά να υπάρχει κάποια έξοδος.

Παράδειγμα 1.5: Δυαδικό Συμμετρικό Κανάλι (BSC)

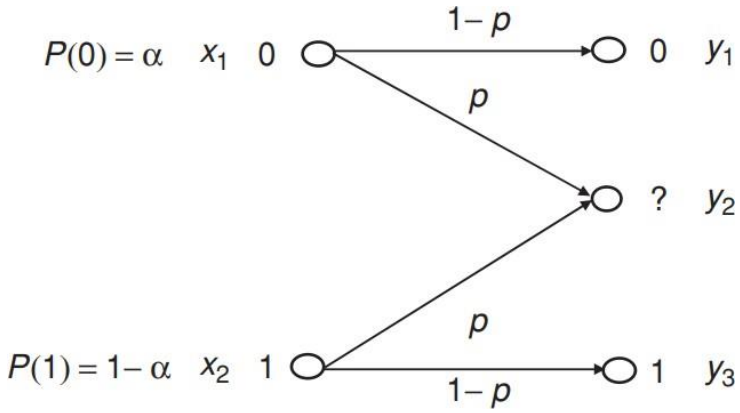
Πρόκειται για ένα κανάλι στο οποίο τα σύμβολα εισόδου είναι 0 και 1, και κάθε σύμβολο έχει πιθανότητα "p" να αλλοιωθεί κατά τη μετάδοση, δηλαδή να μετατραπεί στο αντίθετα του. Η πιθανότητα σωστής μετάδοσης είναι "1 - p".

Ο πίνακας πιθανοτήτων είναι ο εξής:

Λήψη: 0 1

Αποστολή: $\begin{bmatrix} 0 & 1-p & p \\ 1 & p & 1-p \end{bmatrix}$

Παράδειγμα 1.6: Κανάλι Δυαδικής Διαγραφής (BEC)



Στο συγκεκριμένο μοντέλο, όταν η λήψη δεν μπορεί να προσδιορίσει με βεβαιότητα το σταθερό σύμβολο, δηλώνεται η αβεβαιότητα με ένα ειδικό σύμβολο, όπως το "?". Η πιθανότητα διαγραφής είναι p, ενώ οι πιθανότητες λήψης των σωστών τιμών είναι (1 - p).

Ο πίνακας πιθανοτήτων εδώ έχει τρεις εξόδους: 0, 1 και ?. Για παράδειγμα:

Λήψη: 0 ? 1

Αποστολή:

$\begin{bmatrix} 0 & 1-p & p & 0 \\ 1 & 0 & p & 1-p \end{bmatrix}$

1.4 Σχέσεις Πιθανοτήτων Καναλιών

Οι πιθανότητες εξόδου από το κανάλι υπολογίζονται ως συνάρτηση των πιθανοτήτων εισόδου και των πιθανοτήτων μετάβασης.**κ πχ Για κάθε έξοδο yj, η πιθανότητά της δίνεται από το άθροισμα των εξής γινομένων:**

P(yj) = P(yj / x1) * P(x1) + P(yj / x2) * P(x2) + ... + P(yj / xU) * P(xU)

Η παραπάνω σχέση περιγράφει το πώς κάθε πιθανή είσοδος μπορεί να συμβάλει στο να παραχθεί η έξοδος yj.

Για τον υπολογισμό της αντίστροφης πιθανότητας, δηλαδή της πιθανότητας να έχει σταλεί το σύμβολο xi όταν ληφθεί το σύμβολο yj, εφαρμόζεται ο **κανόνας του Bayes:**

P(xi / yj) = [P(yj / xi) * P(xi)] / P(yj)

1.5 A Priori και A Posteriori Εντροπίες

Η πιθανότητα εμφάνισης ενός συμβόλου εξόδου yj συμβολίζεται ως P(yj). Αν γνωρίζουμε το ποιο σύμβολο xi στάλθηκε, τότε μπορούμε να εκφράσουμε την πιθανότητα του yj ως P(yj / xi). Αντιστρόφως, αν παρατηρηθεί το yj, τότε μπορούμε να υπολογίσουμε την πιθανότητα το xi να ήταν το σταλθέν σύμβολο, δηλαδή την P(xi / yj).

Η πιθανότητα P(xi) καλείται a priori, καθώς περιγράφει την πιθανότητα πριν την παρατήρηση του εξόδου. Η πιθανότητα P(xi / yj) καλείται a posteriori, καθώς περιγράφει την πιθανότητα αφού έχει παρατηρηθεί το σύμβολο εξόδου yj.

Η εντροπία πηγής H(X) είναι ο μέσος όρος της πληροφορίας για κάθε σύμβολο εισόδου και δίνεται από το άθροισμα:

$$H(X) = \Sigma P(xi) * \log2(1 / P(xi))$$

Η a posteriori εντροπία H(X / yj), δηλαδή η αβεβαιότητα για την είσοδο μετά την παρατήρηση της εξόδου, δίνεται από:

$$H(X / yj) = \Sigma P(xi / yj) * \log2(1 / P(xi / yj))$$

Παράδειγμα 1.8: Για μια πηγή με P(X=0)=4/5 και P(X=1)=1/5, υπολογίζουμε:

Η a priori εντροπία είναι περίπου 0.7219 bits.

Αν παρατηρηθεί έξοδος '0', η εντροπία H(X / 0) μειώνεται σε περίπου 0.2423 bits.

Αν παρατηρηθεί έξοδος '1', η εντροπία H(X / 1) αυξάνεται σε περίπου 0.9968 bits.

1.6 Αμοιβαία Πληροφορία (Mutual Information)

Η αμοιβαία πληροφορία είναι το μέτρο της πληροφορίας που μεταδίδεται από την είσοδο στην έξοδο. Ορίζεται ως:

$$I(X, Y) = \log2[P(xi / yj) / P(xi)]$$

Σε περίπτωση ιδανικού καναλιού χωρίς θόρυβο, η έξοδος σχετίζεται πλήρως με την είσοδο και η αμοιβαία πληροφορία ισούται με την εντροπία της εισόδου.

Αν το κανάλι είναι θορυβώδες και η έξοδος δεν εξαρτάται από την είσοδο, τότε η αμοιβαία πληροφορία είναι μηδενική.

Μέση τιμή της αμοιβαίας πληροφορίας:

$$I(X, Y) = \Sigma P(xi, yj) * \log2[P(xi / yj) / P(xi)]$$

Η διαφορά μεταξύ της εντροπίας της εισόδου και της αβεβαιότητας που παραμένει μετά την παρατήρηση της εξόδου δίνει την αμοιβαία πληροφορία:

$$I(X, Y) = H(X) - H(X / Y)$$

ή συμμετρικά:

$$I(X, Y) = H(Y) - H(Y / X)$$

1.7 Χωρητικότητα Διακριτού Καναλιού

Η χωρητικότητα καναλιού ορίζεται ως το μέγιστο δυνατό της μέσης αμοιβαίας πληροφορίας:

$$Cs = \max I(X, Y)$$

Η χωρητικότητα δηλώνει τον μέγιστο αριθμό bits ανά σύμβολο που μπορούν να μεταδοθούν αξιόπιστα.

Για το δυαδικό συμμετρικό κανάλι (BSC), ισχύει:

$$Cs = 1 - H(p)$$

Η μέγιστη χωρητικότητα επιτυγχάνεται όταν η πιθανότητα εισόδου είναι συμμετρική, δηλαδή $P(x1) = P(x2) = 0.5$.

Η συνολική ρυθμαπόδοση καναλιού σε bits ανά δευτερόλεπτο είναι:

$$C = s * Cs$$

όπου s είναι ο αριθμός συμβόλων ανά δευτερόλεπτο.

1.8 Θεωρήματα Shannon

1.8.1 Θεώρημα Κωδικοποίησης Πηγής

Η βασική ιδέα είναι ότι μπορούμε να συμπιέσουμε τα δεδομένα που προέρχονται από μια πηγή σε $n*H(X)$ bits, χωρίς να χάνουμε πληροφορία.

Σε μεγάλες ακολουθίες n συμβόλων, η πιθανότητα εμφάνισης μιας συγκεκριμένης ακολουθίας πλησιάζει:

$$P \approx 2^{(-n * H(X))}$$

Μόνο ένα υποσύνολο από τις συνολικές δυνατές ακολουθίες εμφανίζονται συχνά και αυτές είναι οι τυπικές ακολουθίες.

Ο συνολικός αριθμός αυτών των τυπικών ακολουθιών είναι:

$$\approx 2^{(n * H(X))}$$

Αυτό επιτρέπει την συμπίεση του σήματος μέσω τεχνικών απώλειας μηδενικής πληροφορίας.

1.8.2 Χωρητικότητα και Κωδικοποίηση Καναλιού

Η αξιόπιστη μετάδοση σε ένα θορυβώδες κανάλι είναι δυνατή εφόσον ο ρυθμός μετάδοσης δεν ξεπερνά τη χωρητικότητα του καναλιού.

Αν τα μηνύματα μεταδίδονται ως μπλοκ των n bits, η πιθανότητα εμφάνισης λαθών υπολογίζεται ως ο αριθμός των θέσεων στις οποίες η έξοδος διαφέρει από την είσοδο.

Ο μέγιστος αριθμός εισόδων που μπορούν να μεταδοθούν αξιόπιστα είναι:

$$M = 2^{(n * [H(Y) - \Omega(p)])}$$

όπου $\Omega(p)$ είναι η εντροπία θορύβου.

Η μέγιστη ρυθμαπόδοση (χωρητικότητα) ανά bit είναι:

$$Rs = H(Y) - \Omega(p)$$

Για BSC, αν το p είναι μικρό, τότε η ρυθμαπόδοση πλησιάζει την H(X).

1.8.3 Θεώρημα Κωδικοποίησης Καναλιού

Η χωρητικότητα ενός διακριτού καναλιού χωρίς μνήμη ισούται με:

$$Cs = \max I(X, Y)$$

Αν η ταχύτητα μετάδοσης R είναι μικρότερη από τη χωρητικότητα C, τότε υπάρχει τρόπος σχεδίασης κωδίκων που επιτρέπουν αξιόπιστη μετάδοση με πιθανότητα σφάλματος όσο μικρή επιθυμούμε.

Αν όμως R > C, τότε δεν μπορεί να διασφαλιστεί αξιόπιστη επικοινωνία.

1.9 Διανυσματικοί Χώροι και Θεώρημα Κωδικοποίησης Καναλιού

Η αναπαράσταση σημάτων σε διανυσματικούς χώρους επιτρέπει την μελέτη συνεχών καναλιών επικοινωνίας.

Σύμφωνα με το θεώρημα δειγματοληψίας, κάθε σήμα μπορεί να αναπαρασταθεί πλήρως από διακριτά δείγματα. Αν το πλάτος των δειγμάτων έχει μέση ενέργεια P, τότε το διάνυσμα του σήματος έχει νόρμα ρίζα του (n*P), όπου n ο αριθμός δειγμάτων.

Σε Gaussian περιβάλλον με προσθετικό θόρυβο, η έξοδος Y είναι:

$$Y = X + N$$

όπου X είναι το διάνυσμα του σήματος και N το διάνυσμα του θορύβου, με νόρμες ανάλογες προς τις ισχείς P και PN αντίστοιχα. Η συνολική ενέργεια στην έξοδο είναι η άθροιση των δύο, εφόσον ο θόρυβος είναι ασυσχέτιστος με το σήμα.

1.10 Χωρητικότητα Καναλιού Gaussian

Σε κανάλι Gaussian, το λαμβανόμενο σήμα Y είναι το άθροισμα του αρχικού σήματος X και του θορύβου N (δηλαδή Y = X + N). Ο θόρυβος θεωρείται προσθετικός και ανεξάρτητος από το σήμα, με μέση ισχύ PN. Αντίστοιχα, το σήμα έχει ισχύ P.

Το διάνυσμα του σήματος έχει νόρμα ανάλογη με √(nP) και ο θόρυβος √(nPN). Το συνολικό λαμβανόμενο διάνυσμα Y έχει νόρμα το πολύ √(n(P + PN)).

Στον χώρο των σημάτων, αυτό απεικονίζεται με δύο σφαίρες:

Η σφαίρα θορύβου γύρω από το X έχει ακτίνα √(nPN).

Η σφαίρα εξόδου έχει ακτίνα √(n(P + PN)).

Η ερώτηση είναι πόσες τέτοιες σφαίρες (χωρίς επικάλυψη) χωράνε μέσα στη μεγαλύτερη σφαίρα εξόδου. Αυτό προσδιορίζει το πόσα διακριτά μηνύματα μπορούν να μεταδοθούν αξιόπιστα.

Η χωρητικότητα του Gaussian καναλιού προκύπτει ως:

$$Cs = (1/2) * \log2(1 + P / PN) \quad (\text{σε bits ανά δείγμα})$$

Αν το κανάλι δειγματοληπτείται με ρυθμό 2B, τότε η χωρητικότητα ανά δευτερόλεπτο είναι:

$$C = 2B * Cs = 2B * (1/2) * \log2(1 + P / PN) = B * \log2(1 + P / PN)$$

1.11 Κωδικοποίηση για Έλεγχο Σφαλμάτων (Error-Control Coding)

Η πηγή παράγει δεδομένα με ρυθμό R, τα οποία κωδικοποιούνται σε λέξεις μήκους n. Ο κωδικοποιητής μετατρέπει τα δεδομένα σε M διαφορετικούς κωδικούς (κωδικούς λέξεις), καθένας από τους οποίους αντιστοιχεί σε ένα διάνυσμα n δυαδικών ψηφίων.

Η χωρητικότητα του καναλιού είναι:

$$Cs = 1 - \Omega(p)$$

Η μέγιστη πληροφορία ανά σύμβολο εισόδου είναι:

$$R = (s * \log2 M) / n$$

Το θεώρημα του Shannon απαιτεί:

$$R \leq C = s * Cs$$

Αυτό συνεπάγεται:

$$\log2 M / n \leq Cs$$

ή

$$M \leq 2^{(n * Cs)}$$

Ανάλυση Σφαλμάτων Κατά την Αποκωδικοποίηση

Οι λέξεις μεταδίδονται σε διανυσματικό χώρο n διαστάσεων. Κατά την παραλαβή, το διάνυσμα που λαμβάνεται μπορεί να διαφέρει από αυτό που στάλθηκε σε k θέσεις (λόγω σφαλμάτων). Ο αποκωδικοποιητής αποφασίζει ποιος κωδικός στάλθηκε, με βάση σε ποια σφαίρα βρίσκεται το διάνυσμα λήψης.

Αν το διάνυσμα βρίσκεται έξω από τη σφαίρα του σωστού κωδικού ή μέσα στη σφαίρα άλλου κωδικού, προκύπτει σφάλμα.

Η πιθανότητα συνολικού σφάλματος (Pe) είναι:

$$Pe = Ple + Pce$$

Όπου:

- Ple: Πιθανότητα λάθους λόγω του ότι το διάνυσμα λήψης βρίσκεται έξω από τη σωστή σφαίρα.
- Pce: Πιθανότητα ότι δύο ή περισσότεροι κωδικοί έχουν σφαίρες που επικαλύπτονται.

Για να μειωθεί το Ple, η ακτίνα d της σφαίρας λαμβάνεται λίγο μεγαλύτερη από τον αναμενόμενο αριθμό σφαλμάτων (np).

Η πιθανότητα Ple τείνει στο μηδέν όταν $n \rightarrow \infty$, αν ισχύει η σχέση:

$$d = n\beta, \text{ με } \beta > p$$

Η πιθανότητα Pce υπολογίζεται με βάση το πλήθος m των άλλων κωδικών που βρίσκονται μέσα στη σφαίρα ακτίνας d:

$$Pce \approx m / 2^n$$

Το πλήθος αυτό m προσεγγίζεται με βάση συνδυαστικές παραστάσεις και καταλήγει να εξαρτάται εκθετικά από το n.

Τελικά η συνολική πιθανότητα σφάλματος εκφράζεται ως:

$$Pe \leq (K1 / n) + (K2 * 2^{(-nK3)}) + (K4 / \sqrt{n})$$

για κάποιες θετικές σταθερές K1, K2, K3, K4.

Η πιθανότητα τείνει στο μηδέν όσο αυξάνεται το n, εφόσον ο ρυθμός R είναι μικρότερος από την χωρητικότητα C.

1.12 Ιδανικό Σύστημα Επικοινωνίας και Όριο Shannon

1.12.1 Ιδανικό Σύστημα Επικοινωνίας

Ένα ιδανικό σύστημα επικοινωνίας μπορεί να επιτύχει μετάδοση χωρίς σφάλματα όταν ο λόγος σήματος προς θόρυβο (S/N) και το εύρος ζώνης του καναλιού (B) επαρκούν ώστε να ικανοποιείται ο τύπος:

$$R = B * \log2(1 + S/N)$$

Αυτό σημαίνει ότι, με δεδομένη ισχύ σήματος και θόρυβο, υπάρχει ένας μέγιστος ρυθμός R (σε bits ανά δευτερόλεπτο) με τον οποίο μπορούμε να μεταδώσουμε πληροφορία χωρίς λάθη.

Στο ιδανικό αυτό σχήμα:

Η πηγή δημιουργεί μηνύματα από ένα σύνολο M δυνατών σημάτων.

Αυτά μετατρέπονται μέσω ενός πολυτιμητικού κωδικοποιητή (M-ary encoder) σε σήματα.

Το σήμα x(t) περνάει από το κανάλι και επηρεάζεται από προσθετικό Gaussian θόρυβο, παράγοντας y(t).

Το y(t) εντοπίζεται και μετατρέπεται ξανά σε διακριτές τιμές μέσω αποκωδικοποιητή.

Η εξίσωση του Shannon αναφέρει ότι, καθώς η διάρκεια μετάδοσης T μεγαλώνει, ο ρυθμός πλησιάζει τη χωρητικότητα του καναλιού:

$$\lim (R \rightarrow C) \text{ όταν } T \rightarrow \infty \Rightarrow R = B * \log2(1 + S/N)$$

1.13 Ρυθμός και Χωρητικότητα ανά Μονάδα Συχνότητας

Η έκφραση C / B ορίζει τη χωρητικότητα ανά μονάδα συχνότητας (bps/Hz). Με απλή αναδιάταξη, παίρνουμε:

$$C / B = \log2(1 + S / (N0 * B))$$

όπου:

- N0: η φασματική πυκνότητα ισχύος του θορύβου.
- S: ισχύς του σήματος.
- B: εύρος ζώνης.

1.13.1 Σχέση με Ενέργεια ανά Bit

Η ενέργεια ανά bit Eb και η πυκνότητα θορύβου N0 συνδέονται με τον λόγο:

$$Eb / N0 = S / (N0 * R)$$

Αν ο ρυθμός R ισούται με τη χωρητικότητα C, τότε:

$$Eb / N0 = S / (N0 * C)$$

Χρησιμοποιώντας τη σχέση του Shannon για C, λαμβάνουμε:

$$C / B = \log2(1 + (Eb / N0) * (C / B))$$

Από αυτή τη σχέση προκύπτει ότι το ελάχιστο θεωρητικό όριογια το Eb / N0 είναι:

$$Eb / N0 = 1 / \log2(e) \approx 0.693 \text{ (ή } -1.59 \text{ dB)}$$

Αυτό είναι γνωστό ως όριο Shannon και καθορίζει το κατώτατο όριο ενέργειας ανά bit που απαιτείται για αξιόπιστη μετάδοση πληροφορίας.

1.13.2 Ερμηνεία του Ορίου Shannon

Το όριο αυτό ερμηνεύεται ως εξής:

Αν το Eb / N0 είναι μεγαλύτερο από -1.59 dB, τότε είναι θεωρητικά δυνατό να μεταδώσουμε δεδομένα χωρίς σφάλματα, αν χρησιμοποιήσουμε κατάλληλους κωδικοποιητές και μεγάλες ακολουθίες.

Αν είναι μικρότερο, τότε κανένα σύστημα δεν μπορεί να εξασφαλίσει αξιόπιστη μετάδοση.

1.13.3 Περιοχές Πρακτικής και Μη Πρακτικής Λειτουργίας

Σύμφωνα με την εξίσωση:

$$2^{(C/B)} = 1 + (Eb / N0) * (C / B)$$

μπορούμε να κατασκευάσουμε καμπύλες που απεικονίζουν τις περιοχές αποδεκτής λειτουργίας. Αυτές δείχνουν για ποιες τιμές του λόγου Eb / N0 και της αναλογίας R / B μπορούμε να πετύχουμε αξιόπιστη επικοινωνία.

Η περιοχή χωρίζεται σε:

Πρακτική περιοχή λειτουργίας: οι συνδυασμοί (Eb / N0, R/B) για τους οποίους η μετάδοση χωρίς σφάλματα είναι εφικτή.

Μη πρακτική περιοχή: όπου απαιτούνται αδύνατοι φυσικά ή θεωρητικά συνδυασμοί.

Όσο το R / B μικραίνει (δηλαδή η μετάδοση γίνεται πιο «αργή» ανά Hz), τόσο το κατώτατο απαιτούμενο Eb / N0 προσεγγίζει το όριο των -1.59 dB.

Ασκήσεις

Ένα DMS παράγει σύμβολα με τις πιθανότητες που δίνονται στον Πίνακα P.1.1

Table P.1.1 Probabilities of the symbols of a discrete source

A	0.4
B	0.2
C	0.2
D	0.1
E	0.05
F	0.05

(a) Βρείτε την αυτο-πληροφορία που αντιστοιχεί σε κάθε σύμβολο και την εντροπία της πηγής.

(b) Υπολογίστε τη μέγιστη δυνατή εντροπία της πηγής και, συνεπώς, προσδιορίστε την αποδοτικότητα της πηγής.

Λύση:

(a) Υπολογισμός εντροπίας

Για να υπολογίσουμε την εντροπία, πρέπει πρώτα να βρούμε τις αυτοπληροφορίες (self informations):

Ο τύπος είναι:

$$I_i = log_2(\frac{1}{P_i})$$

Υπολογισμοί:

IA = log2(1 / 0.4) = 1.32 bits

IB = log2(1 / 0.2) = 2.32 bits

IC = log2(1 / 0.1) = 3.32 bits

IE = IF = log2(1 / 0.05) = 4.32 bits

Στη συνέχεια, η εντροπία H(X) είναι ο σταθμισμένος μέσος όρος των αυτοπληροφοριών:

H(X) = $\sum_{i=A}^{f} P_i l_i$ = 0.4 * 1.32 + 0.2 * 2.32 + 0.1 * 3.32 + 0.05 * 4.32 + 0.05 * 4.32

= 0.528 + 0.464 + 0.332 + 0.216 + 0.216

= 1.756 ≈ 2.22 bits/symbol

(b) Μέγιστη εντροπία και αποδοτικότητα

Ο μέγιστος δυνατός αριθμός εντροπίας για M διαφορετικά σύμβολα είναι:

$H\,max(x) = \frac{1}{M}\sum \log_2 M = \log_2(M) = \log_2(6) \approx 2.58$ bits/symbol

Η αποδοτικότητα είναι:

eff = 2.22 / 2.58 ≈ 0.86 ή 86%

Συμπέρασμα:

Η πληροφορία που εκπέμπεται από την πηγή αυτή μπορεί να συμπιεστεί.

Υπολογίστε την εντροπία της πηγής, τη διαπληροφόρηση I(X, Y) και τη χωρητικότητα του BSC που ορίζεται στο Σχήμα P.1.1.

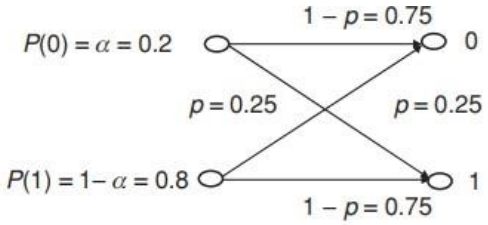


Figure P.1.1 A binary symmetric channel

Υπολογισμός Εντροπίας της Πηγής

Η εντροπία της πηγής δίνεται από:

H(X) = 0.2 * log2(1 / 0.2) + 0.8 * log2(1 / 0.8)

Υπολογίζοντας:

H(X) = 0.2 * log2(5) + 0.8 * log2(1.25) ≈ 0.2 * 2.32 + 0.8 * 0.32≈ 0.464 + 0.256

= 0.7219 bits/symbol

Υπολογισμός Διαπληροφορίας (Transinformation)

Η διαπληροφόρηση υπολογίζεται με τον τύπο:

I(X;Y) = Ω(α + p − 2αp) − Ω(p)

Όπου:

Ω(p) = p * log_2(1 / p) + (1 − p) * $\log_2$(1 / (1 − p))

Βήμα 1: Υπολογισμός Ω(p)

Δίνεται p = 0.25:

Ω(0.25) = 0.25 * $\log_2$(1 / 0.25) + 0.75 * $\log_2$(1 / 0.75) = 0.25 * $\log_2$(4) + 0.75 * $\log_2$(1.333) = 0.25 * 2 + 0.75 * 0.415 = 0.5 + 0.31125 = 0.8112 bits

Βήμα 2: Υπολογισμός α + p − 2αp

Δίνεται α = 0.2 και p = 0.25:

α + p − 2αp = 0.2 + 0.25 − 2 * 0.2 * 0.25 = 0.2 + 0.25 − 0.1 = 0.35

Βήμα 3: Υπολογισμός Ω(0.35)

$\Omega(0.35) = 0.35 * \log_2 (1 / 0.35) + 0.65 * \log_2 (1 / 0.65) = 0.35 * \log_2 (2.857) + 0.65 * \log_2 (1.538) \approx 0.35 * 1.51 + 0.65 * 0.62 \approx 0.5285 + 0.403 = 0.9315 \approx 0.934$ bits

Βήμα 4: Υπολογισμός διαπληροφορίας

I(X;Y) = Ω(0.35) − Ω(0.25) = 0.934 − 0.8112 = 0.1228 ≈ 0.123 bits/symbol

Υπολογισμός Χωρητικότητας (Capacity)

Η χωρητικότητα του καναλιού είναι το μέγιστο της I(X;Y) ως προς την κατανομή εισόδου:

Cs = maxP(xi) I(X;Y) = 1 − Ω(p)

= 1 − 0.8112

= 0.1888 ≈ 0.189 bits/symbol

Ποια είναι η χωρητικότητα της ακολουθίας (cascaded) BSCs όπως δίνεται στο Σχήμα P.1.2;

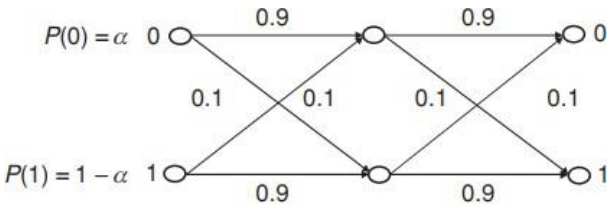


Figure P.1.2 A cascade of BSCs

Βήμα 1 – Υπολογισμός των συνολικών πιθανοτήτων μετάδοσης:

Το σύνθετο κανάλι (η αλυσίδα δύο BSC) μπορεί να αναπαρασταθεί ως ένα νέο κανάλι με συνολικές πιθανότητες:

Πιθανότητα σωστής μετάδοσης:

-

$P(0 \rightarrow 0) = 0.9 * 0.9 + 0.1 * 0.1 = 0.81 + 0.01 = 0.82$

Πιθανότητα λανθασμένης μετάδοσης:

$P(0 \rightarrow 1) = 0.9 * 0.1 + 0.1 * 0.9 = 0.09 + 0.09 = 0.18$

Άρα το ισοδύναμο κανάλι είναι:

$P(Y = X) = 0.82$

$P(Y \neq X) = 0.18$

Βήμα 2 – Υπολογισμός χωρητικότητας καναλιού (channel capacity):

Η χωρητικότητα BSC με πιθανότητα σφάλματος p δίνεται από:

$C = 1 - [p * \log_2 (1 / p) + (1 - p) * \log_2 (1 / (1 - p))]$

Αντικαθιστούμε:

$p = 0.18$

$1 - p = 0.82$

$C = 1 - [0.18 * \log_2 (1 / 0.18) + 0.82 * \log_2 (1 / 0.82)]$

$C = 1 - [0.18 * \log_2 (5.555...) + 0.82 * \log_2 (1.2195)]$

$C = 1 - [0.18 * 2.47 + 0.82 * 0.29]$

$C = 1 - [0.4446 + 0.2378]$

$C = 1 - 0.6824$

$C = 0.3176$ (περίπου)

Στη φωτογραφία χρησιμοποιήθηκαν ακριβέστερες τιμές λογαρίθμων και προκύπτει:

$C = 0.3199$ bits/symbol

Τελική Απάντηση:

Το ισοδύναμο κανάλι είναι ένα BSC με:

P(σωστή μετάδοση) = 0.82

P(σφάλμα) = 0.18

και χωρητικότητα:

$C = 0.3199$ bits/symbol

Θεωρήστε ένα δυαδικό κανάλι με αλφάβητα εισόδου και εξόδου {0, 1} και με πίνακα πιθανοτήτων μετάβασης:

$$\boldsymbol{P}_{\text{ch}} = \begin{bmatrix} 3/5 & 2/5 \\ 1/5 & 4/5 \end{bmatrix}$$

Προσδιορίστε την εκ των προτέρων (a priori) και τις δύο εκ των υστέρων (a posteriori) εντροπίες αυτού του καναλιού.

Λύση:

Δίνεται:

Ένα δυαδικό κανάλι με πίνακα μεταβάσεων:

$$\boldsymbol{P}_{\text{ch}} = \begin{bmatrix} 3/5 & 2/5 \\ 1/5 & 4/5 \end{bmatrix}$$

Δηλαδή:

- P(Y=0 | X=0) = 3/5
- P(Y=1 | X=0) = 2/5
- P(Y=0 | X=1) = 1/5
- P(Y=1 | X=1) = 4/5

Υποθέτουμε ομοιόμορφη κατανομή εισόδου:

P(X=0) = 1/2

P(X=1) = 1/2

Υπολογισμός πιθανοτήτων εξόδου:

P(Y=0):

P(Y=0) = P(Y=0 | X=0) * P(X=0) + P(Y=0 | X=1) * P(X=1)

= (3/5)*(1/2) + (1/5)*(1/2) = (3 + 1)/10 = 4/10 = 2/5

P(Y=1):

P(Y=1) = 1 - P(Y=0) = 1 - 2/5 = 3/5

Υπολογισμός εκ των υστέρων πιθανοτήτων (posterior):

P(X=0 | Y=0):

= [P(Y=0 | X=0) * P(X=0)] / P(Y=0) = [(3/5)*(1/2)] / (2/5) = (3/10) / (2/5) = 3/10 * 5/2 = 3/4

P(X=1 | Y=0):

= [(1/5)*(1/2)] / (2/5) = (1/10) / (2/5) = 1/10 * 5/2 = 1/4

P(X=0 | Y=1):

= [(2/5)*(1/2)] / (3/5) = (2/10) / (3/5) = 1/5 * 5/3 = 1/3

P(X=1 | Y=1):

= [(4/5)*(1/2)] / (3/5) = (4/10) / (3/5) = 2/5 * 5/3 = 2/3

Υπολογισμός εκ των υστέρων εντροπιών:

H(X | Y = 0):

= - [(3/4) * log2(3/4) + (1/4) * log2(1/4)]

= - [(3/4)*log2(0.75) + (1/4)*log2(0.25)]

≈ - [(3/4)*(-0.415) + (1/4)*(-2)]

≈ - [-0.31125 - 0.5] = 0.8112 bit

H(X | Y = 1):

= - [(1/3) * log2(1/3) + (2/3) * log2(2/3)]

≈ - [(1/3)*(-1.585) + (2/3)*(-0.585)]

≈ - [-0.5283 - 0.3903] = 0.9182 bit

Υπολογισμός a priori εντροπίας:

Αφού η είσοδος είναι ομοιόμορφη:

H(X) = - [(1/2) * log2(1/2) + (1/2) * log2(1/2)] = 1 bit

Τελική Απάντηση:

- A priori εντροπία:

H(X) = 1 bit

- A posteriori εντροπία όταν Y = 0:

H(X | Y = 0) ≈ 0.8112 bit

- A posteriori εντροπία όταν Y = 1:

$H(X \mid Y = 1) \approx 0.9182$ bit

Βρείτε τις υπό συνθήκη πιθανότητες P(xi / yj) του BEC με πιθανότητα διαγραφής 0,469, όταν οι πιθανότητες της πηγής είναι 0,25 και 0,75. Στη συνέχεια, βρείτε την αβεβαιότητα (equivocation), τη διαπληροφόρηση (transinformation) και τη χωρητικότητα του καναλιού.

Δίνονται:

- $p(x_1) = \alpha = 0.25$
- $p(x_2) = 1 - \alpha = 0.75$
- $p(y_1 \mid x_1) = 1 - p = 0.531$
- $p(y_2 \mid x_1) = 0$
- $p(y_1 \mid x_2) = 0$
- $p(y_2 \mid x_2) = 0.469$

Υπολογισμοί πιθανοτήτων εξόδου:

P(Y = 0):

$P(Y = 0) = P(Y = 0 \mid X = 0) * P(X = 0) + P(Y = 0 \mid X = 1) * P(X = 1)$

$= \alpha * (1 - p) + 0 = \alpha(1 - p)$

$= 0.25 * 0.531 = 0.13275$

P(Y = 1):

$P(Y = 1) = P(Y = 1 \mid X = 1) * P(X = 1) + P(Y = 1 \mid X = 0) * P(X = 0)$

$= p * (1 - \alpha)$

$= 0.469 * 0.75 = 0.35175$

P(Y = E):

$P(Y = E) = 1 - P(Y = 0) - P(Y = 1) = 1 - 0.13275 - 0.35175 = 0.5155$

Εκ των υστέρων πιθανότητες (Posterior Probabilities):

$P(X = 0 \mid Y = 0)$:

$= [P(Y = 0 \mid X = 0) * P(X = 0)] / P(Y = 0)$

$= [\alpha(1 - p)] / P(Y = 0) = \alpha(1 - p) / [\alpha(1 - p)] = 1$

$P(X = 1 \mid Y = 0)$: $= 0$

P(X = 0 | Y = 1): = 0

P(X = 1 | Y = 1): = 1

P(X = 0 | Y = E):

= [P(Y = E | X = 0) * P(X = 0)] / P(Y = E)

= [(p) * α] / P(Y = E)

= (0.469 * 0.25) / 0.5155 ≈ 0.227 / 0.5155 ≈ 0.44

P(X = 1 | Y = E):

= [P(Y = E | X = 1) * P(X = 1)] / P(Y = E)

= [(1 - p) * (1 - α)] / P(Y = E)

= (0.531 * 0.75) / 0.5155 ≈ 0.398 / 0.5155 ≈ 0.56

Αμοιβαία Πληροφορία I(X;Y):

Ο τύπος που χρησιμοποιείται είναι:

I(X;Y) = (1 - p) * Ω(α) = (1 - p) * [α * log2(1/α) + (1 - α) * log2(1 / (1 - α))]

Με τιμές:

- α = 0.25
- 1 - p = 0.531
- log2(1 / 0.25) = log2(4) = 2
- log2(1 / 0.75) = log2(1.333) ≈ 0.415

Άρα:

I(X;Y) = 0.531 * [0.25 * 2 + 0.75 * 0.415] = 0.531 * [0.5 + 0.31125] = 0.531 * 0.81125 ≈ 0.4307 bit

A Posteriori Εντροπία (Equivocation):

H(X | Y) = H(X) - I(X;Y)

= 0.8112 - 0.4307 ≈ 0.38 bit

Τελική Απάντηση:

- Αμοιβαία Πληροφορία: I(X;Y) ≈ 0.4307 bit
- A Posteriori Εντροπία: H(X | Y) ≈ 0.38 bit

Θεωρήστε μετάδοση μέσω μιας τηλεφωνικής γραμμής με εύρος ζώνης B = 3 kHz. Πρόκειται για ένα αναλογικό κανάλι το οποίο μπορεί να θεωρηθεί ότι επηρεάζεται από AWGN, και για το οποίο ο λόγος σήματος προς θόρυβο ισχύος είναι τουλάχιστον,30dB.

(a) Ποια είναι η χωρητικότητα αυτού του καναλιού, υπό τις παραπάνω συνθήκες;

(b) Ποιος είναι ο απαιτούμενος λόγος σήματος προς θόρυβο για να μεταδοθεί ένα σήμα M-ary ικανό να μεταφέρει 19.200 bps;

(α) Αν C = 3000 bps, και S/N = 1000, τότε:

C = B * log2(1 + S/N)

= 3000 * log2(1 + 1000)

= 3000 * log2(1001)

≈ 3000 * 9.967 = 29902 bps

(β) Αν C = 19200 bps και B = 3000 Hz:

19200 = 3000 * log2(1 + S/N)

log2(1 + S/N) = 19200 / 3000 = 6.4

1 + S/N = 2^6.4 ≈ 83.44

S/N ≈ 82.44

S/N (σε dB) = 10 * log10(83.44) ≈ 19.21 dB

2 Κώδικες Μπλοκ (Block Codes)

Οι κώδικες μπλοκ χρησιμοποιούνται για ανίχνευση και διόρθωση σφαλμάτων.Η παρουσίαση βασίζεται σε πίνακες G, H και στη μέθοδο του συνδρόμου.

2.1 Κωδικοποίηση Ελέγχου Σφαλμάτων

Σύμφωνα με το θεώρημα του Shannon, είναι δυνατή η μετατροπή ενός θορυβώδους καναλιού σε αξιόπιστο μέσω κατάλληλης κωδικοποίησης. Η ιδέα αυτή υλοποιείται με την προσθήκη πλεονάζουσας πληροφορίας στα αρχικά δεδομένα, ώστε να εντοπίζονται και να διορθώνονται σφάλματα.

Τα μηνύματα οργανώνονται σε ομάδες των k bit και μετατρέπονται σε κωδικές λέξεις n bit (όπου n > k), μέσω μιας αμφιμονοσήμαντης αντιστοίχισης. Η διαδικασία αυτή αποτελεί έναν κώδικα μπλοκ. Υπάρχουν δύο βασικές κατηγορίες τεχνικών προσθήκης πλεονασμού: οι κώδικες μπλοκ και οι συνελικτικοί κώδικες. Το παρόν κεφάλαιο εστιάζει στους πρώτους.

Η κωδικοποίηση μπορεί είτε να εντοπίζει είτε να διορθώνει σφάλματα. Ο εντοπισμός απαιτεί λιγότερες πληροφορίες από τη διόρθωση, που προϋποθέτει γνώση της θέσης και της φύσης του σφάλματος.

2.2 Εντοπισμός και Διόρθωση Σφαλμάτων

Στην πράξη, ο εντοπισμός σφαλμάτων είναι ευκολότερος από τη διόρθωση. Ανάλογα με την εφαρμογή, μπορεί να εφαρμοστεί είτε μηχανισμός επαναμετάδοσης (ARQ – Automatic Repeat reQuest) είτε προληπτική διόρθωση (FEC – Forward Error Correction).

Η επαναμετάδοση εφαρμόζεται όταν είναι δυνατή η αμφίδρομη επικοινωνία. Σε περιπτώσεις όπως η αποστολή SMS ή το paging, όπου δεν είναι εφικτή η επαναποστολή, εφαρμόζεται FEC: ο δέκτης αναγνωρίζει και διορθώνει μόνος του τα σφάλματα.

2.2.1 Απλοί Κώδικες: Ο Κώδικας Επανάληψης

Μια απλή τεχνική είναι η επανάληψη του ίδιου bit πολλές φορές. Π.χ. για n = 3, το bit 1 μεταδίδεται ως 111 και το bit 0 ως 000. Αν ένα από τα τρία bit αλλοιωθεί (π.χ. 110), τότε ο δέκτης αποφασίζει πως το αρχικό bit ήταν 1.

Ο κώδικας επανάληψης μπορεί να εντοπίσει ένα ή δύο σφάλματα, αλλά όχι τρία. Επίσης, μπορεί να διορθώσει ένα σφάλμα. Η πιθανότητα σφάλματος εξαρτάται από το p (πιθανότητα σφάλματος ανά bit).

Για παράδειγμα:

Πιθανότητα σφάλματος λέξης (μόνο εντοπισμός): $Pwe = p^3 \cdot$ Πιθανότητα σφάλματος λέξης (με διόρθωση): $Pwe = 3p^2(1-p) + p^3 = 3p^2 - 2p^3$

Αν και ο κώδικας επανάληψης προσφέρει καλή ανθεκτικότητα σε σφάλματα, έχει πολύ μικρό ρυθμό μετάδοσης, επειδή μεταδίδονται πολλά περιττά bits.

Ο ρυθμός του κώδικα ορίζεται ως:

$$Rc = k / n$$

2.3 Κώδικες Μπλοκ: Εισαγωγή και Παράμετροι

Στους κώδικες μπλοκ, το αρχικό μήνυμα χωρίζεται σε μπλοκ k bits. Αυτά κωδικοποιούνται σε μπλοκ n bits, προσθέτοντας n − k πλεονάζοντα bits. Η κωδικοποίηση γίνεται έτσι ώστε τα αρχικά δεδομένα να μπορούν να ανακτηθούν μέσω αντίστροφης διαδικασίας στον δέκτη.

Οι κώδικες μπλοκ συμβολίζονται συνήθως ως C(n, k). Ο ρυθμός τους επηρεάζει άμεσα το εύρος ζώνης: όσο περισσότερα πλεονάζοντα bits, τόσο περισσότερο bandwidth απαιτείται.

Παράδειγμα: Αν ο ρυθμός είναι 2/3, σε δεδομένο χρόνο Τ, ο μη κωδικοποιημένος δέκτης λαμβάνει 2 bits, ενώ ο κωδικοποιημένος 3 bits. Άρα η διάρκεια του κάθε σήματος αλλάζει, και η φασματική πυκνότητα αυξάνεται.

2.4 Διανυσματικός Χώρος πάνω στο Δυαδικό Πεδίο

Ο διανυσματικός χώρος αποτελείται από σύνολα διανυσμάτων στα οποία ισχύουν συγκεκριμένοι αλγεβρικοί κανόνες. Στην περίπτωση των δυαδικών κωδίκων, χρησιμοποιείται το δυαδικό πεδίο GF(2), όπου τα στοιχεία παίρνουν τιμές 0 ή 1.

Ο διανυσματικός χώρος Vn είναι το σύνολο όλων των διανυσμάτων μήκους n. Περιλαμβάνει συνολικά 2n διανύσματα. Οι πράξεις πάνω σε αυτά είναι:

- Πρόσθεση Modulo-2 (0+0=0, 1+0=1, 1+1=0)
- Πολλαπλασιασμός Modulo-2 (1×1=1, τα υπόλοιπα 0)

Ο μηδενικός διάνυσμα (0,0,…,0) είναι το ουδέτερο στοιχείο της πρόσθεσης. Κάθε διάνυσμα έχει το δικό του αρνητικό, που είναι το ίδιο.

2.4.1 Υποχώροι Διανυσματικού Χώρου (Vector Subspaces)

Ένας υποχώρος είναι υποσύνολο του διανυσματικού χώρου που πληροί τους ίδιους κανόνες. Π.χ. το σύνολο S = {(0000), (1001), (0100), (1101)} είναι υποχώρος του V4.

Κάθε διάνυσμα του υποχώρου μπορεί να γραφτεί ως γραμμικός συνδυασμός άλλων διανυσμάτων (π.χ.

$a_1v_1 + a_2v_2 + \ldots + a_k*v_k$).

Αυτό το πλαίσιο μάς επιτρέπει να αναλύσουμε τους κώδικες μπλοκ με βάση την αλγεβρική δομή τους, κάτι που είναι θεμελιώδες για την κατανόηση της κωδικοποίησης και της αποκωδικοποίησης. Αν ένας υποχώρος S έχει διάσταση k στον διανυσματικό χώρο V_n, τότε υπάρχει ένας δυϊκός υποχώρος S_d, ώστε για κάθε u \in S και κάθε v \in S_d να ισχύει ότι το εσωτερικό γινόμενο u \cdot v = 0. Αυτό σημαίνει ότι οι διανυσματικοί χώροι είναι ορθογώνιοι (orthogonal).

Ισχύει η σχέση:

$$\dim(S) + \dim(Sd) = n$$

2.4.2 Μορφή Πίνακα (Matrix Form)

Οι γραμμικά ανεξάρτητοι διανύσματα που παράγουν έναν υποχώρο μπορούν να οργανωθούν ως γραμμές ενός πίνακα G. Αυτός ο πίνακας έχει διαστάσεις k × n και λέγεται πίνακας γεννήτορας (generator matrix). Οι γραμμές του αποτελούν βάση για την παραγωγή των 2^k δυνατών κωδικών λέξεων.

Επιτρέπονται γραμμικές πράξεις στον πίνακα G, χωρίς να αλλάζει ο υποχώρος που παράγεται.

2.4.3 Πίνακας Δυϊκού Υποχώρου

Ο πίνακας H δημιουργείται από τους γραμμικά ανεξάρτητους διανύσματα που ορίζουν τον δυϊκό υποχώρο Sd. Αν ο G έχει k γραμμικά ανεξάρτητες γραμμές, τότε ο H έχει διαστάσεις (n − k) × n, και κάθε γραμμή του H είναι ορθογώνια ως προς κάθε γραμμή του G (δηλαδή: gi · hj = 0).

Ο H χρησιμοποιείται για ελέγχους ισοτιμίας και ανίχνευση σφαλμάτων.

2.5 Γραμμικοί Κώδικες Μπλοκ (Linear Block Codes)

Οι παραπάνω έννοιες χρησιμοποιούνται για τον ορισμό των γραμμικών κωδίκων μπλοκ. Ένα μήνυμα m = (m_0, m_1, …, m_{k-1}) αποτελείται από k bits και αντιστοιχίζεται σε μια κωδική λέξη c = (c_0, c_1, …, c_{n-1}) μέσω προσθήκης πλεονάζουσας πληροφορίας.

Οι 2^k δυνατοί κωδικοί σχηματίζουν έναν υποχώρο διάστασης k στον διανυσματικό χώρο Vn. Η αντιστοίχιση αυτή είναι αμφιμονοσήμαντη και επιλέγεται ώστε να ελαχιστοποιείται η πλεοναζουσα πληροφορία και να μεγιστοποιείται η ικανότητα διόρθωσης.

2.5.1 Πίνακας Γεννήτορας G

Ο πίνακας G αποτελείται από τις k γραμμικά ανεξάρτητες γραμμές (g_0, g_1, …, g_{k-1}) που σχηματίζουν τους κωδικούς. Η παραγωγή μιας κωδικής λέξης από το μήνυμα m γίνεται μέσω πολλαπλασιασμού:

$$c = m \cdot G$$

Η διαδικασία αυτή υλοποιείται μέσω εσωτερικών γινομένων ανάμεσα στο διάνυσμα m και τις γραμμές του πίνακα G.

2.5.2 Κώδικες σε Συστηματική Μορφή (Systematic Form)

Στην παρούσα περιγραφή οι συστηματικοί κώδικες έχουν τη μορφή όπου τα k bits του αρχικού μηνύματος τοποθετούνται στο τέλος της κωδικής λέξης, ενώ τα (n − k) bits ελέγχου βρίσκονται στην αρχή. Έτσι, η κωδική λέξη αποτελείται από:

- (n − k) bits ισοτιμίας (parity)
- k bits πληροφορίας

Αυτό επιτρέπει εύκολη αναγνώριση των δεδομένων μέσα στην κωδική λέξη.

2.5.3 Πίνακας Ελέγχου Ισοτιμίας H (Parity Check Matrix)

Ο πίνακας ελέγχου ισοτιμίας H κατασκευάζεται έτσι ώστε κάθε γραμμή του να είναι ορθογώνια ως προς κάθε γραμμή του πίνακα G. Δηλαδή, για κάθε γραμμή g του G και h του H ισχύει:

$$g \cdot h = 0$$

Η συστηματική μορφή του πίνακα H είναι:

$$H = [I \mid P^t]$$

όπου το P^t είναι ο ανάστροφος (transpose) του υποπίνακα P του G. Η σχέση:

$$G \cdot H^t = 0$$

εξασφαλίζει ότι οι παραγόμενες κωδικές λέξεις πληρούν τις εξισώσεις ελέγχου ισοτιμίας.

2.6 Ανίχνευση Σφαλμάτων μέσω Συνδρόμου

Κατά τη μετάδοση, η κωδική λέξη c μπορεί να μετατραπεί λόγω θορύβου σε ένα διάνυσμα r. Η διαφορά $e = r \oplus c$ είναι το διάνυσμα σφάλματος.Για να εντοπιστεί η ύπαρξη σφαλμάτων, υπολογίζεται το σύνδρομο:

$$S = r \cdot H^t$$

Αν $S = 0$, τότε θεωρείται ότι δεν υπάρχουν σφάλματα. Αν $S \neq 0$, τότε εντοπίζεται σφάλμα. Σε κάποιες περιπτώσεις το σύνδρομο μπορεί να είναι μηδέν ακόμη κι αν υπάρχει σφάλμα (μη ανιχνεύσιμο σφάλμα). Τα μη ανιχνεύσιμα σφάλματα ισχύουν όταν:

$$e \cdot H^t = 0 \Rightarrow e \text{ είναι κωδική λέξη}$$

Ο αριθμός των τέτοιων μοτίβων είναι $2^k - 1$ (μη μηδενικά κωδικά διανύσματα).

2.7 Ελάχιστη Απόσταση Κώδικα (Minimum Distance)

Η ελάχιστη απόσταση dmin ενός κώδικα είναι το ελάχιστο πλήθος διαφορετικών θέσεων μεταξύ δύο διαφορετικών κωδικών λέξεων. Ορίζεται ως:

$$dmin = \min d(ci, cj), \text{ για όλα τα } ci \neq cj$$

Η απόσταση Hamming μεταξύ δύο λέξεων είναι ίση με το βάρος (αριθμός 1) του αθροίσματός τους $ci \oplus cj$. Άρα:

$$d(ci, cj) = w(ci \oplus cj)$$

Η dmin καθορίζει πόσα σφάλματα μπορεί να εντοπίσει ή να διορθώσει ο κώδικας.

Ελάχιστη Απόσταση και Πίνακας Η

> Η ελάχιστη απόσταση μπορεί να υπολογιστεί και από τον πίνακα Η. Αν το άθροισμαπ.χ. τριών στηλών του Η δίνει μηδενικό διάνυσμα, τότε υπάρχει λέξη με βάρος 3, άρα dmin = 3

2.8 Ικανότητα Διόρθωσης Σφαλμάτων

Αν dmin είναι η ελάχιστη απόσταση του κώδικα, τότε ο μέγιστος αριθμός σφαλμάτων t που μπορούν να διορθωθούν ικανοποιεί:

$$2t + 1 \leq dmin$$

Για παράδειγμα, αν dmin = 7, τότε t = 3, δηλαδή διορθώνονται μέχρι 3 σφάλματα. Η πιθανότητα αποτυχίας διόρθωσης εξαρτάται από το πλήθος και τη θέση των σφαλμάτων.

2.9 Ανίχνευση Συνδρόμου και Πρότυπη Διάταξη (Standard Array)

Η τεχνική standard array είναι ένας τρόπος αποκωδικοποίησης. Χωρίζει τον διανυσματικό χώρο σε ομάδες (cosets) των 2^k κωδικών λέξεων. Κάθε ομάδα ξεκινά από έναν ηγέτη σφάλματος (error leader), ο οποίος είναι το πιο πιθανό μοτίβο σφάλματος.

Η πρώτη σειρά περιλαμβάνει τις 2^k κανονικές κωδικές λέξεις, ξεκινώντας από το (0…0). Κάτω από κάθε λέξη γράφεται το άθροισμα με κάθε ηγέτη σφάλματος. Το πλήθος των ομάδων είναι 2^{n-k}.

Η αποκωδικοποίηση γίνεται αντιστρέφοντας το σφάλμα του ηγέτη και βρίσκοντας την κωδική λέξη που αντιστοιχεί στο ληφθέν διάνυσμα.

Ένας κώδικας C(n,k)έχει 2^(n-k) δυνατά σύνδρομα,άρα μπορεί να αντιστοιχίσει μέχρι 2^(n-k) διαφορετικά μοτίβα σφαλμάτων σε μοναδικά σύνδρομα (δηλαδή να διορθώσει το πολύ 2^(n-k)-1 μη μηδενικά μοτίβα).

Ένας γραμμικός μπλοκ κώδικας με ελάχιστη απόσταση d_min μπορεί να χρησιμοποιεί όλους τους διανύσματα σφαλμάτων βάρους μικρότερου ή ίσου από $t = \lfloor (d_{min} - 1)/2 \rfloor$ ως ηγέτες coset (δηλαδή ως πρότυπα σφαλμάτων). Αυτό διευκολύνει τη διαδικασία απλής διόρθωσης σφαλμάτων με βάση τα σύνδρομα.

Το σύνδρομο κάθε λαμβανόμενου διανύσματος υπολογίζεται ως το γινόμενο r * H^T, όπου r είναι το λαμβανόμενο διάνυσμα και H^T είναι ο πίνακας ελέγχου ισοτιμίας (μεταθετικός). Με αυτό τον τρόπο, το σύστημα μπορεί να εντοπίσει το είδος του σφάλματος και να το διορθώσει.

Παράδειγμα 2.16:

Για τον γραμμικό μπλοκ κώδικα C(7,4), ο πίνακας ελέγχου ισοτιμίας είναι:

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

Αυτός ο κώδικας έχει 16 διαφορετικούς κωδικούς και 8 πιθανά μοτίβα σφαλμάτων που μπορούν να διορθωθούν (cosets). Το ελάχιστο βάθος Hamming είναι 3, που σημαίνει ότι μπορεί να διορθώσει οποιοδήποτε σφάλμα ενός bit.

Στο πίνακα 2.4 παρουσιάζεται η αντιστοίχιση ανάμεσα στα μοτίβα σφαλμάτων και τα σύνδρομά τους.

Table 2.4 Error patterns and their corresponding syndrome vectors, Example 2.16

Error patterns							Syndromes		
1	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	1	0
0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	1	0	1	1	1
0	0	0	0	0	0	1	1	0	1

2.10 Κώδικες Hamming

Οι κώδικες Hamming είναι μία ειδική κατηγορία γραμμικών μπλοκ κωδίκων που μπορούν να διορθώσουν ένα σφάλμα ανά λέξη. Για κάθε m ≥ 3, ισχύει:

- Μήκος κώδικα: n = 2^m − 1
- Πλήθος bit πληροφορίας: k = 2^m − m − 1
- Πλήθος bit ισοτιμίας: n − k = m
- Ικανότητα διόρθωσης: t = 1 (ελάχιστη απόσταση d_min = 3)

Ο πίνακας H μπορεί να γραφεί σε συστηματική μορφή ως H = [Im Q], όπου Im είναι η μοναδιαία υπομήτρα και Q η υπόλοιπη.

-

Για m = 3, προκύπτει ο κώδικας C(7,4), που είναι ο ίδιος με το προηγούμενο παράδειγμα.

2.11 Διόρθωση Σφαλμάτων με Προώθηση και Αυτόματη Επανάληψη

2.11.1 Διόρθωση Σφαλμάτων με Προώθηση (FEC)

Στα συστήματα FEC, δεν γίνεται επανάληψη της μετάδοσης. Όλη η διόρθωση γίνεται από τον παραλήπτη, με βάση τα πρόσθετα bit ελέγχου. Ο ρυθμός μετάδοσης r πρέπει να είναι μεγαλύτερος από τον ρυθμό πληροφορίας rb. Ο λόγος Eb/N0 παίζει σημαντικό ρόλο στην αξιολόγηση της απόδοσης του συστήματος.

Η πιθανότητα σφάλματος εκφράζεται με τον τύπο:

$$P_{be} \approx [(n-1)/t] * [Q(\sqrt{(2 * Rc * Eb/N0)})]^{(t+1)}, \text{ κατά προσέγγιση}$$

Όπου Q είναι η συνάρτηση Q που αντιστοιχεί στην πιθανότητα λάθους ενός bit λόγω θορύβου.

2.11.2 Αυτόματη Επανάληψη Κατόπιν Αιτήματος (ARQ)

Το ARQ βασίζεται στην ανίχνευση σφαλμάτων και την επανάληψη μετάδοσης. Ο δέκτης στέλνει θετική απάντηση (ACK) όταν δεν εντοπίζει σφάλμα και αρνητική (NAK) όταν εντοπίζει. Το πακέτο με σφάλμα μεταδίδεται ξανά.

2.11.3 Σχήματα ARQ

- Stop and Wait: Ο πομπός στέλνει μία λέξη και περιμένει ACK ή NAK πριν στείλει την επόμενη. Είναι απλό, αλλά έχει μεγάλη καθυστέρηση.
- Go-back-N: Ο πομπός στέλνει διαδοχικά πολλές λέξεις. Αν υπάρξει σφάλμα, ξαναστέλνει όλες από το σημείο του σφάλματος και μετά.
- Selective Repeat: Ο πομπός ξαναστέλνει μόνο τα πακέτα με σφάλματα. Είναι πιο αποδοτικό, αλλά χρειάζεται περισσότερη μνήμη.

2.11.4 Αποδοτικότητα ARQ

Για να αξιολογηθεί η αποδοτικότητα ενός συστήματος ARQ, εξετάζεται ο μέσος αριθμός μεταδόσεων ανά λέξη και υπολογίζεται η αποδοτικότητα Rc'.

Για το Stop and Wait, η αποδοτικότητα μειώνεται λόγω των καθυστερήσεων και της ανάγκης για επιβεβαίωση μετάδοσης. Δίνεται από τη σχέση:

$$Rc' \leq (k/n) * (1 - P\_ret) / (1 + 2t\_d * rb / k)$$

2.12 Διόρθωση Σφαλμάτων Προώθησης (FEC) και Αυτόματο Αίτημα Επανάληψης (ARQ)

Τα συστήματα FEC δεν απαιτούν επαναμετάδοση. Ο πομπός προσθέτει bit ελέγχου και στέλνει το κωδικοποιημένο μήνυμα.

Ο ρυθμός μετάδοσης είναι:

$$r = rb / Rc, \text{ όπου } Rc = k / n$$

Η ενέργεια ανά bit στο κωδικοποιημένο σήμα είναι:

$$(Eb / N0)c = Rc * (Eb / N0)$$

Πιθανότητες λάθους

Πιθανότητα λάθους λέξης:

$$Pwe \approx (n / (t + 1)) * p^{(t + 1)}$$

Πιθανότητα λάθους bit:

$$Pbe \approx ((n - 1) / t) * p^{(t + 1)}$$

Αντίστοιχα, η θεωρητική πιθανότητα σφάλματος bit σε FEC σύστημα είναι:

$$Pe = Q(sqrt(2 * Eb / N0))$$

Και για κωδικοποιημένο bit:

$$Pbe \approx ((n - 1) / t) * (Q(sqrt(2 * Rc * Eb / N0)))^{(t + 1)}$$

2.12.1 Απόδοση Stop-and-Wait ARQ

Η καθυστέρηση πακέτου είναι:

$$Tw = n / r = k / rb$$

Η απόδοση είναι:

$$Rc' = (k / n) * (1 - Pret) / (1 + D / Tw)$$

ή ισοδύναμα:

$$Rc' = (k / n) * (1 - Pret) / (1 + (2 * td * rb) / k)$$

όπου:

- D είναι η καθυστέρηση καναλιού,
- td είναι η καθυστέρηση διάδοσης.

2.12.2 Απόδοση Go-Back-N

Μέσος αριθμός μεταδόσεων:

$$\bar{m} = 1 + (N * Pret) / (1 - Pret)$$

Απόδοση:

$$Rc' = (k / n) * (1 - Pret) / (1 - Pret + N * Pret)$$

2.12.3 Υβριδικά Συστήματα ARQ

Η FEC εφαρμόζει διόρθωση. Αν αποτύχει, ενεργοποιείται μηχανισμός ARQ.

- Για type-1 hybrid ARQ:

  αν η FEC δεν καταφέρει να διορθώσει το μήνυμα, ζητείται επαναμετάδοση του ίδιου μηνύματος.

- Για type-2 hybrid ARQ:

αποστέλλεται πρώτα μόνο το μήνυμα. Αν χρειαστεί, στέλνονται πρόσθετα bits ισοτιμίας για διόρθωση, χωρίς να στείλουμε ξανά όλο το μήνυμα.

2.12.4 Αποδοτικότητα Σχημάτων ARQ

Το σχήμα Go-back-N ARQ δεν παρουσιάζει απώλειες χρόνου από άσκοπες παύσεις του πομπού, αλλά πρέπει να επαναλάβει τη μετάδοση όταν εντοπιστεί κάποιο σφάλμα. Σε αυτή την περίπτωση, ο μέσος αριθμός μεταδόσεων ανά λέξη είναι μεγαλύτερος, και εξαρτάται από την πιθανότητα επαναμετάδοσης. Έτσι, η αποδοτικότητα της μετάδοσης υπολογίζεται μειώνοντας τον αρχικό ρυθμό με έναν διορθωτικό συντελεστή.

Η τελική αποδοτικότητα εξαρτάται από την τιμή του N (αριθμός λέξεων στο buffer) και τον λόγο της καθυστέρησης του καναλιού προς τον χρόνο μετάδοσης της λέξης. Εάν η καθυστέρηση του καναλιού είναι μικρή (π.χ. το διπλάσιο της καθυστέρησης είναι μικρότερο από το μήκος του μηνύματος), τότε η αποδοτικότητα είναι ικανοποιητική. Σε αντίθετη περίπτωση, η απόδοση μειώνεται σημαντικά.

Στο διάγραμμα της Εικόνας 2.8 παρουσιάζεται η αποδοτικότητα (ως λόγος k προς n) για τα τρία βασικά σχήματα ARQ: stop-and-wait, go-back-N και selective repeat. Το selective repeat εμφανίζει την καλύτερη απόδοση, ειδικά σε συνθήκες μεγάλων καθυστερήσεων στο δίκτυο, ενώ το stop-and-wait παρουσιάζει σημαντικές απώλειες όταν ο χρόνος καθυστέρησης είναι μεγάλος.

Τελικά, το go-back-N προσφέρει καλή απόδοση όταν η καθυστέρηση είναι μικρή και η πιθανότητα σφάλματος είναι χαμηλή. Το selective repeat είναι η καλύτερη επιλογή σε δίκτυα με υψηλό ρυθμό μετάδοσης και μεγάλες καθυστερήσεις.

ΑΣΚΗΣΕΙΣ

2.1 Για τον τριπλό επαναληπτικό κώδικα μπλοκ Cb(3, 1), που παράγεται με τη χρήση του υποπίνακα ελέγχου ισοτιμίας P = [1 1], κατασκευάστε τον πίνακα όλων των πιθανών ληφθέντων διανυσμάτων και υπολογίστε τα αντίστοιχα διανύσματα συνδρόμου S = r • HT, ώστε να προσδιορίσετε τα μοτίβα σφαλμάτων που μπορούν να διορθωθούν και να εντοπιστούν, σύμφωνα με την ικανότητα διόρθωσης σφαλμάτων του κώδικα.

$$\boldsymbol{P} = [1 \quad 1], \qquad \boldsymbol{G} = [1 \quad 1 \quad 1], \qquad \boldsymbol{H} = [\boldsymbol{I}_{n-k} \quad \boldsymbol{P}^T] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{H}^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

code table

m	C
0	000
1	111

Syndrome table

r	s
000	00
001	11
010	01
011	10
100	10
101	01
110	11
111	00

Πιθανά Μοτίβα Σφαλμάτων (8 στο σύνολο), τα πιο πιθανά είναι:

e	e•H^T
000	00
001	11
010	01
100	10

Αυτά είναι τα μοτίβα μεμονωμένων σφαλμάτων (1 bit). Τα συνδρόματά τους είναι όλα διαφορετικά, άρα ο κώδικας μπορεί να διορθώσει οποιοδήποτε μοτίβο 1 σφάλματος (t = 1).

Επιπλέον, ο κώδικας μπορεί να ανιχνεύσει οποιοδήποτε μοτίβο 2 σφαλμάτων (l = 2).

Πλήρης Πίνακας Ανάλυσης:

r	$r \cdot H^t$	e	$r \oplus e$	m
000	00	000	000	0
001	11	001	000	0
010	01	010	000	0
011	10	100	111	1
100	10	100	000	0
101	01	010	111	1
110	11	001	111	1
111	00	000	111	1

Σε MATLAB:

```
% Άσκηση 2.1 – Διόρθωση διαστάσεων

G = [1 1];                      % Γεννήτρια G = [I | P]
H = [1 0; 1 1];                 % Πίνακας ελέγχου

m = [0; 1];                     % Πληροφοριακά bits
c = mod(m * G, 2);              % Κωδικοί

r_all = de2bi(0:3, 2, 'left-msb');   % ΟΛΕΣ οι πιθανές λέξεις 2 bit
syndromes = mod(r_all * H', 2);      % Συνδρόματα

fprintf('r     S     m_hat\n');
for i = 1:size(r_all, 1)
    r = r_all(i,:);
    S = mod(r * H', 2);
    found = false;
    for j = 1:size(c,1)
        if isequal(mod(c(j,:) * H', 2), S)
            m_hat = m(j);
            found = true;
            break;
        end
    end
    if ~found
        m_hat = NaN;
    end
    fprintf('%s   %s    %s\n', num2str(r), num2str(S), num2str(m_hat));
end
```

```
Command Window
>> ask2_1
r     S     m_hat
0  0   0  0    0
0  1   0  1    NaN
1  0   1  1    NaN
1  1   1  0    1
```

2.2 Η ελάχιστη απόσταση Hamming ενός κώδικα μπλοκ είναι dmin = 11. Προσδιορίστε την ικανότητα διόρθωσης και την ικανότητα ανίχνευσης σφαλμάτων αυτού του κώδικα.
Ελάχιστη απόσταση: d_min = 11

Για ανίχνευση έως l σφαλμάτων:

$d\_min \geq l + 1 \Rightarrow 11 \geq l + 1 \Rightarrow l = 10$

Για διόρθωση έως t σφαλμάτων:

$d\_min \geq 2t + 1 \Rightarrow 11 \geq 2t + 1 \Rightarrow t = 5$

Σε MATLAB :

```
dmin = 11;

% Ανίχνευση έως l σφαλμάτων
l = dmin - 1;

% Διόρθωση έως t σφαλμάτων
t = floor((dmin - 1) / 2);

fprintf('Μέγιστος αριθμός ανιχνευόμενων σφαλμάτων: l = %d\n', l);
fprintf('Μέγιστος αριθμός διορθώσιμων σφαλμάτων: t = %d\n', t);
```

```
Command Window
>> ask2_2
Μέγιστος αριθμός ανιχνευόμενων σφαλμάτων: l = 10
Μέγιστος αριθμός διορθώσιμων σφαλμάτων: t = 5
```

2.3 (a) Προσδιορίστε την ελάχιστη απόσταση Hamming ενός κώδικα μήκους n που πρέπει να ανιχνεύει έως έξι σφάλματα και να διορθώνει έως τέσσερα σφάλματα ανά διανύσμα κώδικα, σε μετάδοση πάνω από ένα BSC.

(b) Αν ο ίδιος κώδικας μπλοκ χρησιμοποιηθεί σε ένα δυαδικό κανάλι διαγραφών (binary erasure channel), πόσες διαγραφές σε ένα μπλοκ μπορεί να ανιχνεύσει και πόσες μπορεί να διορθώσει;

(c) Ποιο είναι το ελάχιστο μήκος μπλοκ που μπορεί να έχει ένας κώδικας με αυτή την ελάχιστη απόσταση Hamming;

Λύση:

(a)

Δίνεται ότι:

$l + t + 1 \geq d_{min}$

Αν $l = 6$ και $t = 4$, τότε:

$6 + 4 + 1 = 11 \geq d_{min} \Rightarrow$ Ισχύει.

Επίσης, ισχύει:

$d_{min} = n - k + 1$

Αν $k = 1$ και $n = 11$, τότε:

$d_{min} = 11 - 1 + 1 = 11$

(b)

Ο κώδικας μπορεί να:

- ανιχνεύσει n απώλειες (erasures)
- διορθώσει 10 απώλειες

(c)

Και εδώ επαληθεύεται:

$d_{min} = n - k + 1 = 11$, με $k = 1$ και $n = 11$

Σε MATLAB :

```
l = 6; t = 4;
dmin = 11; n = 11; k = 1;

check = l + t + 1 >= dmin;
dmin_calc = n - k + 1;

fprintf('Ισχύει l + t + 1 >= d_min: %d\n', check);
fprintf('Υπολογισμένο d_min από Singleton: %d\n', dmin_calc);
```

```
Command Window
>> ask2_3
Ισχύει l + t + 1 >= d_min: 1
Υπολογισμένο d_min από Singleton: 11
```

2.4 Ένας δυαδικός κώδικας μπλοκ έχει διανύσματα κώδικα σε συστηματική μορφή όπως δίνονται στον Πίνακα P.2.1

Table P.2.1 A block code table

0	0	0	0	0	0
0	1	1	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	0	0	0	1
1	0	1	1	0	1
0	1	1	0	1	1
0	0	0	1	1	1

(a) Ποιος είναι ο ρυθμός του κώδικα;

(b) Γράψτε τον πίνακα γεννήτορα και τον πίνακα ελέγχου ισοτιμίας αυτού του κώδικα σε συστηματική μορφή.

(c) Ποια είναι η ελάχιστη απόσταση Hamming του κώδικα;

(d) Πόσα σφάλματα μπορεί να διορθώσει και πόσα μπορεί να ανιχνεύσει;

(e) Υπολογίστε το διάνυσμα συνδρόμου για το ληφθέν διάνυσμα r = (101011) και στη συνέχεια βρείτε τη θέση τυχόν σφάλματος.

Λύση:

(a)

Από τον πίνακα κατανομής βαρών προκύπτει:

Αριθμός πληροφοριακών bit: k = 3

Αριθμός συνολικών bit: n = 6

Ο ρυθμός του κώδικα είναι:

R = k / n = 3 / 6 = 0.5

(b)

Επιλέγοντας 3 γραμμικά ανεξάρτητα διανύσματα κώδικα, σχηματίζουμε τον πίνακα γεννήτριας G, με μορφή [P | I], ώστε το δεξιό τμήμα να είναι η μοναδιαία υπομήτρα.

Ο πίνακας G είναι:

001	100
101	010
110	001

Ο πίνακας ελέγχου ισοτιμίας H^t είναι:

100
010
001
110
011
101

(c)

Αφού πρόκειται για γραμμικό μπλοκ κώδικα, η ελάχιστη απόσταση είναι το μικρότερο βάρος (αριθμός άσων) σε μη μηδενικό κωδικό.

Από τον πίνακα, το ελάχιστο βάρος είναι 3, άρα:

$d_{min} = 3$

$l = 2, t = 1$

(d)

Δίνεται το διάνυσμα:

$r = (101011)$

Υπολογίζουμε το σύνδρομο:

$S = r \cdot H^t = (101011) \cdot H^t = (1\ 1\ 0)$

Το σύνδρομο αντιστοιχεί σε μοτίβο σφάλματος:

$e = (000001)$

Άρα:

$c = r \oplus e = (101010)$

Πίνακας Συνδρόμου - Σφάλματος:

e	$e \cdot H^t$
100000	100
010000	010
001000	001
000100	011
000010	101
000001	110

Σε MATLAB :

```
G = [0 1 1 1 0 0;
     1 0 1 0 1 0;
     1 1 0 0 0 1];

HT = [1 0 0;
      0 1 0;
      0 0 1;
      1 1 0;
      0 1 1;
      1 0 1];

% Υπολογισμός d_min
codewords = mod(de2bi(0:7, 3, 'left-msb') * G, 2);
weights = sum(codewords, 2);
dmin = min(weights(weights > 0));

% Υπολογισμός συνδρόμου για r
r = [1 0 1 0 1 1];
S = mod(r * HT, 2);

% Πίνακας μονοψήφιων λαθών και συνδρόμων
fprintf('e          S\n');
for i = 1:6
    e = zeros(1,6); e(i) = 1;
    syn = mod(e * HT, 2);
    fprintf('%s   %s\n', num2str(e), num2str(syn));
end

% Εύρεση λάθους
syndrome_table = [1 0 0;
                  0 1 0;
                  0 0 1;
                  0 1 1;
                  1 0 1;
                  1 1 0];

e_table = eye(6);
idx = find(ismember(syndrome_table, S, 'rows'), 1);
e = e_table(idx, :);
c = mod(r + e, 2);

fprintf('Διορθωμένος κωδικός: %s\n', num2str(c));
```

```
Command Window
>> ask2_4
e          S
1  0  0  0  0  0   1  0  0
0  1  0  0  0  0   0  1  0
0  0  1  0  0  0   0  0  1
0  0  0  1  0  0   1  1  0
0  0  0  0  1  0   0  1  1
0  0  0  0  0  1   1  0  1
Διορθωμένος κωδικός: 1  0  1  1  1  1
```

2.5

(a) Κατασκευάστε έναν γραμμικό κώδικα μπλοκ Cb(5, 2), μεγιστοποιώντας την ελάχιστη απόστασή του Hamming.

(b) Προσδιορίστε τους πίνακες γεννήτορα και ελέγχου ισοτιμίας αυτού του κώδικα.

ΛΥΣΗ

Σε έναν γραμμικό μπλοκ κώδικα C_b(5, 2), με k = 2 και n = 5, υπάρχουν $2^2 = 4$ κωδικοί μήκους 5. Ο πίνακας γεννήτριας έχει τη μορφή:

$$\boldsymbol{G} = [\boldsymbol{P} \quad \boldsymbol{I}_k] = \begin{bmatrix} \boldsymbol{P} & \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \end{bmatrix}$$

Από τους πιθανούς 32 κωδικούς μήκους 5, μπορούμε να επιλέξουμε c_1 = (11110) και c_2 = (01011), ώστε να σχηματίσουμε τον πίνακα G ως:

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

Ο κώδικας έχει τον εξής πίνακα κωδικών:

m	c	w
00	00000	0
01	10101	3
10	11110	4

-

11	01011	3

Η ελάχιστη απόσταση του κώδικα είναι d_min = w_min = 3.

Ο πίνακας H^Tέχει διαστάσεις 3×5, άρα τα σύνδρομα έχουν μέγεθος 3 bits. Ο αριθμός των μη μηδενικών συνδρόμων είναι 2^3- 1 = 7. Εφόσον το μήκος του κώδικα είναι 5, δεν μπορούμε να περιμένουμε διόρθωση 2 σφαλμάτων, αφού για αυτό θα χρειαζόμασταν τουλάχιστον:

C(5,1) + C(5,2) = 5 + 10 = 15 σύνδρομα.

Ο πίνακας ελέγχου ισοτιμίας H^Tτου κώδικα είναι:

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

Σε MATLAB: `clear; clc;`

```
% (a) Κατασκευή Cb(5,2) με d_min μέγιστο
G = [1 1 1 1 0;
     1 0 1 0 1];        % 2x5

% (b) Πίνακας ελέγχου ισοτιμίας (H) ώστε GH' = 0
H = [1 0 1 0 0;
     0 1 0 1 0;
     1 1 0 0 1];        % 3x5
HT = H.';

% Έλεγχος ορθότητας GH' = 0 (mod 2)
ok = all(mod(G*HT,2)==0,'all');

% Παραγωγή όλων των κωδικών (m ∈ {00,01,10,11})
msgs = [0 0; 0 1; 1 0; 1 1];
C = mod(msgs*G,2);                  % 4x5
w = sum(C,2);
dmin = min(w(w>0));
t = floor((dmin-1)/2);
l = dmin-1;

% Εκτύπωση αποτελεσμάτων
fprintf('G =\n'); disp(G);
fprintf('H =\n'); disp(H);
fprintf('Έλεγχος GH''==0 (mod 2): %d\n', ok);
T = table(msgs(:,1),msgs(:,2),C(:,1),C(:,2),C(:,3),C(:,4),C(:,5),w, ...
    'VariableNames', {'m1','m2','c1','c2','c3','c4','c5','w'});
disp(T);
fprintf('d_min = %d,  t = %d,  l = %d\n', dmin, t, l);

% Παράδειγμα συνδρόμου και διόρθωσης 1-σφάλματος
```

```
m = [1 0];                % παράδειγμα μηνύματος
c = mod(m*G,2);           % κωδική λέξη 1x5
e = [0 0 0 1 0];          % σφάλμα στη θέση 4
r = mod(c + e, 2);        % ληφθέν
S = mod(r*HT,2);          % σύνδρομο 1x3
err_pos = find(all(H == S.',1));    % θέση σφάλματος από H

c_hat = r;
if ~isempty(err_pos), c_hat(err_pos)=mod(c_hat(err_pos)+1,2); end

fprintf('m = '); disp(m);
fprintf('c = '); disp(c);
fprintf('r = '); disp(r);
fprintf('S = '); disp(S);
fprintf('Θέση σφάλματος: %d\n', isempty(err_pos)*0 + ~is-
empty(err_pos)*err_pos);
fprintf('c_hat = '); disp(c_hat);
```

Command Window

```
G =
     1     1     1     1     0
     1     0     1     0     1

H =
     1     0     1     0     0
     0     1     0     1     0
     1     1     0     0     1

Έλεγχος GH'==0 (mod 2): 1
    m1    m2    c1    c2    c3    c4    c5    w
    --    --    --    --    --    --    --    -

     0     0     0     0     0     0     0     0
     0     1     1     0     1     0     1     3
     1     0     1     1     1     1     0     4
     1     1     0     1     0     1     1     3

d min = 3,  t = 1,  l = 2
```

-

```
m =      1     0

c =      1     1     1     1     0

r =      1     1     1     0     0

S =      0     1     0

Θέση σφάλματος: 4
c_hat =      1     1     1     1     0
```

2.6 Τυχαία σφάλματα με πιθανότητα $p = 10^{-3}$ σε ένα BSC πρόκειται να διορθωθούν από έναν block code ελέγχου τυχαίων σφαλμάτων με n = 15, k = 11 και dmin = 3. Ποιος είναι ο συνολικός ρυθμός διέλευσης (throughput rate) και η πιθανότητα σφάλματος μπλοκ μετά την αποκωδικοποίηση, όταν ο κώδικας χρησιμοποιείται

(a) σε λειτουργία FEC και

(b) σε λειτουργία διόρθωσης σφαλμάτων με επαναμετάδοση (ARQ mode);

ΛΥΣΗ:

(a) Ο κώδικας είναι μία συντομευμένη εκδοχή του κώδικα Hamming $C_h(15,11)$, με υποπίνακα ισοτιμίας:

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

$2^4 - 1 \geq n \Rightarrow 2^4 \geq 9 \Rightarrow n - k = 4 \Rightarrow k = 6$

(b) n - k = 4

(c) R_c = k / n = 6 / 10 = 3 / 5

(d) Ο πίνακας H^t κατασκευάζεται ως:

$$H^T = \begin{bmatrix} I_{n-k} \\ P \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Ο πίνακας γεννήτριας είναι:

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

(e) c = (1111111) * G = (00100111111)

(f) Ένα σφάλμα στη θέση 7 αντιστοιχεί σε e = (0000001000), το οποίο δίνει σύνδρομο (0 1 0), που είναι η 7η γραμμή του πίνακα H^t.

ΣΕ MATLAB: `clear; clc;`

```
P = [0 0 1 1;
     0 1 0 1;
     1 0 0 1;
     0 1 0 0;
     1 0 1 0;
     0 1 1 0];

G = [eye(6) P];
H = [P' eye(4)];

k = 6; n = 10; R = k/n;
fprintf('(a) R = %d/%d = %.3f\n', k, n, R);

disp('(b) G ='); disp(G);
disp('H ='); disp(H);

dmin = 3;
t = floor((dmin-1)/2);
l = dmin - 1;
```

-

```
fprintf('(c) d_min = %d\n', dmin);
fprintf('(d) t = %d, l = %d\n', t, l);

m = [1 0 1 0 1 1];
c = mod(m * G, 2);
fprintf('(e) m = '); disp(m);
fprintf('c = '); disp(c);

e = zeros(1,10); e(7) = 1;
r = mod(c + e, 2);
fprintf('r = '); disp(r);

S = mod(r * H', 2);
fprintf('S = '); disp(S);

err_pos = find(all(H == S.', 1));
if isempty(err_pos)
    if all(S == 0)
        fprintf('No error\n');
        err_pos = 0;
    else
        fprintf('Syndrome not single-error\n');
        err_pos = NaN;
    end
else
    fprintf('Error position: %d\n', err_pos);
end

c_hat = r;
if ~isnan(err_pos) && err_pos > 0
    c_hat(err_pos) = mod(c_hat(err_pos) + 1, 2);
end
fprintf('c_hat = '); disp(c_hat);

if ~isnan(err_pos)
    ok = isequal(c, c_hat);
    fprintf('Corrected: %s\n', string(ok));
end
```

```
(a) R = 6/10 = 0.600
(b) G =
     1     0     0     0     0     0     0     0     1     1
     0     1     0     0     0     0     0     1     0     1
     0     0     1     0     0     0     1     0     0     1
     0     0     0     1     0     0     0     1     0     0
     0     0     0     0     1     0     1     0     1     0
     0     0     0     0     0     1     0     1     1     0

H =
     0     0     1     0     1     0     1     0     0     0
     0     1     0     1     0     1     0     1     0     0
     1     0     0     0     1     1     0     0     1     0
     1     1     1     0     0     0     0     0     0     1

(c) d_min = 3
(d) t = 1, l = 2
(e) m =      1     0     1     0     1     1


c =      1     0     1     0     1     1     0     1     1     0

r =      1     0     1     0     1     1     1     1     1     0

S =      1     0     0     0

Error position: 7
c_hat =      1     0     1     0     1     1     0     1     1     0

Corrected: true
```

2.7 Ένα σχήμα FEC λειτουργεί σε κανάλι AWGN και θα πρέπει να επιτυγχάνει ρυθμό σφάλματος bit Pbe < 10^{-4}, χρησιμοποιώντας την ελάχιστη μεταδιδόμενη ισχύ. Οι επιλογές για αυτό το σχήμα είναι οι κώδικες μπλοκ όπως δίνονται στον Πίνακα P.2.2.

Table P.2.2 Options for Problem 2.11

n	k	d_{min}
31	26	3
31	21	5
31	16	7

(a) Προσδιορίστε την καλύτερη επιλογή αν απαιτείται η ελάχιστη μεταδιδόμενη ισχύς.

(b) Υπολογίστε το κέρδος κωδικοποίησης (coding gain) στο επιθυμητό Pbe σε σχέση με τη μετάδοση χωρίς κωδικοποίηση.

ΛΥΣΗ:

Δίνεται:

p = 10^{-3} (πιθανότητα σφάλματος ανά bit)

n = 15 (μήκος κωδικού)

k = 11 (πληροφοριακά bits)

d_min = 3 (ελάχιστη απόσταση)

Άρα:

R_c = k / n = 11 / 15 ≈ 0.733

(a) FEC mode (Forward Error Correction):

$$P_{we} \approx \binom{n}{t+1} p^{t+1} = \binom{15}{2} (10^{-3})^2 = 1.04x10^{-4}$$

(a) ARQ mode (Automatic Repeat reQuest):

$$R_c = k / n(1-p)^n = 11/15(1-0-001)^{15} = 0.722$$

$$P_{we} \approx \binom{n}{l+1} p^{l+1} = \binom{15}{3} (10^{-3})^3 = 4.05x10^{-7}$$

Σε MATLAB: %% Άσκηση 2.11 – FEC και ARQ

```
p = 1e-3;      % πιθανότητα σφάλματος bit
n = 15;
k = 11;
dmin = 3;
t = floor((dmin - 1) / 2);   % t = 1

% Rate χωρίς ARQ
Rc = k / n;

%% (a) FEC mode
Pwe_FEC = nchoosek(n, t+1) * (p)^(t+1);
fprintf('(a) FEC mode:\n');
fprintf('R = %.3f\n', Rc);
fprintf('P_we ≈ %.2e\n\n', Pwe_FEC);

%% (b) ARQ mode
Rc_ARQ = (k / n) / ( (1 - p)^n );  % ρυθμός με ARQ
Pwe_ARQ = nchoosek(n, t+2) * (p)^(t+2);

fprintf('(b) ARQ mode:\n');
fprintf('R_c = %.3f\n', Rc_ARQ);
fprintf('P_we ≈ %.2e\n', Pwe_ARQ);
```

Command Window

```
>> ask2_11
(a) FEC mode:
R = 0.733
P_we ≈ 1.05e-04

(b) ARQ mode:
R_c = 0.744
P_we ≈ 4.55e-07
```

3 Κυκλικοί κώδικες

Οι κυκλικοί κώδικες είναι μια σημαντική κατηγορία γραμμικών block κωδικών , οι οποίοι διακρίνονται από την εύκολη υλοποίηση με την χρήση κυκλωμάτων μετατόπισης.

-

3.1 Εισαγωγή

Για ένα διάνυσμα n συνιστωσών c=(c0,c1,…,cn−1) η δεξιά κυκλική μετατόπιση των στοιχείων του κατά i θέσεις οδηγεί σε ένα νέο διάνυσμα, το οποίο ορίζεται ως:

$$c^{(i)}= C_{n-i},C_{n-i+1},\ldots,c_{n-1},c_0,c_1, \ldots , C_{n-i-1}$$

Ένας γραμμικός block κώδικας χαρακτηρίζεται ως κυκλικός, εάν για κάθε κωδικό διάνυσμα του κώδικα, η i-οστή κυκλική μετατόπιση αποτελεί επίσης έγκυρο κωδικό διάνυσμα του ίδιου κώδικα. Επιπλέον, λόγω της γραμμικότητας, το άθροισμα δύο οποιωνδήποτε κωδικών διανυσμάτων ανήκει και αυτό στον ίδιο κώδικα.

3.2 Πολυωνυμική Αναπαράσταση Κωδικών Λέξεων

Στους κυκλικούς κώδικες, κάθε κωδική λέξη μήκους n μπορεί να αναπαρασταθεί ως πολυώνυμο βαθμού το πολύ n-1 με συντελεστές που ανήκουν στο σώμα GF(2) , δηλαδή στο δυαδικό πεδίο {0,1}. Έτσι, για μια κωδική λέξη:

$$\text{C}= (C_0, C_2,\ldots,C_{n-1}) , c_i \in GF(2)$$

η αντίστοιχη Πολυωνυμική μορφή είναι:

$$c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$$

Στο δυαδικό πεδίο, οι πράξεις μεταξύ πολυωνύμων ορίζονται modulo 2, κάτι που σημαίνει ότι η πρόσθεση αντιστοιχεί στη λογική πράξη XOR, ενώ ο πολλαπλασιασμός αντιστοιχεί στη λογική πράξη AND, σε συνδυασμό με πρόσθεση mod 2.

Για παράδειγμα, έστω οι δύο κωδικές λέξεις:

$c_1(x) = 1 + x^2 + x^3$, $c_2(x) = x + x^3$

Η πρόσθεση τους προκύπτει ως:

$$C_1(x) + C_2(x) = 1 + (x^2) + (x^3 + x^3) + x = 1 + x + x^2$$

επειδή στο GF(2) ισχύει 1+1=0.

Η αναπαράσταση αυτή είναι ιδιαίτερα χρήσιμη, καθώς η έννοια του **πολυωνύμου γεννήτριας** g(x) ορίζει τον ίδιο τον κώδικα: κάθε έγκυρη κωδική λέξη είναι πολλαπλάσιο του g(x). Έτσι, για κάθε c(x) ισχύει:

$$C(x) = m(x) \cdot g(x)$$

όπου m(x) είναι το πολυώνυμο πληροφορίας. Επιπλέον, ο έλεγχος εγκυρότητας μιας λέξης μπορεί να γίνει με διαίρεση πολυωνύμων:

$$C(x) = q(x) \cdot g(x) \cdot r(x)$$

Αν το υπόλοιπο r(x)=0, τότε η λέξη ανήκει στον κώδικα.

Παράδειγμα:

Έστω κώδικας με γεννήτρια g(x)=1+x+x3. Η λέξη c=(1,0,1,1) που αντιστοιχεί στο πολυώνυμο c(x)=1+x2+x3, όταν διαιρεθεί με το g(x), δίνει μη μηδενικό υπόλοιπο. Επομένως, η λέξη αυτή **δεν** είναι έγκυρη κωδική λέξη.

3.3 Γεννήτρια Πολυωνύμων

Η κυκλική μετατόπιση κατά i θέσεις ενός κωδικού διανύσματος c μπορεί να εκφραστεί με Πολυωνυμική μορφή. Αν το πολυώνυμο που αντιστοιχεί στο διάνυσμα είναι c(X), τότε η μετατόπιση προκύπτει από τον πολλαπλασιασμό του με x^i και τη λήψη του υπολοίπου στη διαίρεση με το πολυώνυμο x^{n+1}. Συνεπώς:

$$C^{(i)}(x) = x^i C(x)\, mod(x^n + 1)$$

Ανάμεσα σε όλα τα πολυώνυμα ενός κυκλικού κώδικα $C_{CyC}\,(n, k)$, υπάρχει πάντα ένα πολυώνυμο μικρότερου βαθμού, το οποίο και ονομάζεται **πολυώνυμο γεννήτορας**. Το πολυώνυμο αυτό, που συνήθως συμβολίζεται με g(X), είναι διαιρέτης του x^{n+1} . Κάθε άλλο πολυώνυμο του κώδικα είναι πολλαπλάσιο του g(X). Έτσι, το g(X) λειτουργεί ως βάση για τη δημιουργία όλων των κωδικών πολυωνύμων.

Το πολυώνυμο γεννήτορας έχει βαθμό r = n−k, όπου n είναι το μήκος του κώδικα και k το μήκος του μηνύματος. Ο αριθμός r εκφράζει και το πλήθος των **πλεονασματικών bit** (redundancy) που εισάγει ο κώδικας, ενώ παράλληλα δείχνει πόσοι είναι οι βαθμοί ελευθερίας μεταξύ των πολυωνύμων του κώδικα.

Ένα κωδικό πολυώνυμο του κυκλικού κώδικα μπορεί να γραφεί ως:

$$C(x) = m(x) \cdot g(x)$$

όπου m(X) είναι το πολυώνυμο μηνύματος βαθμού το πολύ k−1. Η σχέση αυτή δείχνει ότι κάθε κωδικό πολυώνυμο προκύπτει από τον πολλαπλασιασμό του πολυωνύμου γεννήτορα με το πολυώνυμο μηνύματος.

Παράδειγμα:

Στον κυκλικό κώδικα $c_{cyc}(7,4)$, το πολυώνυμο γεννήτορας είναι:

g(X)=1+X+X3

Αν κωδικοποιήσουμε τα μηνύματα

- m_0=(0000)
- m_1=(1000)
- m_2=(0100)
- m_3=(1100)

τότε τα αντίστοιχα κώδικά πολυώνυμα προκύπτουν από τον πολλαπλασιασμό κάθε πολυωνύμου μηνύματος με το g(X).

Για παράδειγμα, για το μήνυμα m_1 :

$$C(x) = x^3 \cdot g(x) = x^3(1 + x + x^3) = x^3 + x^4 + x^6$$

Έτσι προκύπτει ο πλήρης πίνακας των κωδικών πολυωνύμων του κώδικα.

Τέλος, αξίζει να σημειωθεί πως η σχέση ανάμεσα στο πολυώνυμο γεννήτορα g(X) και στο x^{n+1} είναι θεμελιώδης:

- Αν το g(X) είναι πολυώνυμο βαθμού r και αποτελεί διαιρέτη του x^{n+1}, τότε παράγει έναν γραμμικό κυκλικό κώδικα $C_{CyC}(n, k)$.
- Αντίστροφα, κάθε κυκλικός κώδικας ορίζεται μοναδικά από το πολυώνυμο γεννήτορά του.

3.4 Κωδικοποίηση με Κυκλικούς Κώδικες

Η συστηματική μορφή ενός κώδικα διατηρεί τα bit του αρχικού μηνύματος ως μέρος της κωδικής λέξης.

Για ένα μήνυμα m(X), υπολογίζουμε:

1. Το $X^{n-k} * m(X)$
2. Διαιρούμε με g(X), βρίσκουμε το υπόλοιπο p(X)
3. Η τελική κωδική λέξη είναι:

$$c(X) = X^{n-k} * m(X) + p(X)$$

Η κωδική λέξη τότε έχει τη μορφή:

$$c = (p_0, p_1, ..., p_{n-k-1}, m_0, ..., m_{k-1})$$

Τα πρώτα bits είναι τα πλεονάζοντα (redundancy), και τα τελευταία τα bit του μηνύματος.

Παράδειγμα:

Για το m = 1010 στο C(7,4) με $g(X) = 1 + X + X^3$:

$m(X) = 1 + X^2$

$X^3 * m(X) = X^3 + X^5$

Διαιρούμε με g(X), παίρνουμε υπόλοιπο $p(X) = X^2$

Άρα:

$c(X) = X^3 * m(X) + p(X) = X^2 + X^3 + X^5$

c = (0 0 1 1 0 1 0)

3.5 Πίνακας Κωδικών (Πλήρες Παράδειγμα)

Ο πίνακας του κώδικα C(7,4) δείχνει όλους τους συνδυασμούς 4-bit μηνυμάτων και τις αντίστοιχες κωδικές λέξεις 7-bit.

Μήνυμα (m)	**Κωδική λέξη (c)**
0000	0000000
0001	1011001
0010	0110100
0011	1101101
...	...
1111	0100111

3.6 Ιδιότητες και Σχέσεις

- Αν g(X) διαιρεί το $X^n + 1$, τότε είναι έγκυρο γεννήτριο πολυώνυμο.
- Το πολυώνυμο $X^k * g(X)$ είναι και αυτό πολυώνυμο του κώδικα.
- Υπάρχει μοναδικό πολυώνυμο ελάχιστου βαθμού που ορίζει πλήρως τον κώδικα.
- Ο αριθμός των δυνατών κωδικών είναι 2^k (με k: αριθμός bit του μηνύματος).

3.7 Πίνακας Γεννήτριας Κυκλικού Κώδικα

Ο πίνακας γεννήτριας ενός κυκλικού κώδικα C_{cyc}(n,k) προκύπτει από την κυκλική μετατόπιση του πολυωνύμου γεννήτριας g(X). Ο πίνακας αυτός περιέχει τις μετατοπίσεις των πολυωνύμων g(X), xg(X) ,x^2g(X),…x^{k-1}g(X) και έχει διαστάσεις $k \times n$. Δεν είναι αναγκαστικά σε συστηματική μορφή, αλλά μπορεί να μετατραπεί με στοιχειώδεις γραμμικές πράξεις.

Παράδειγμα 3.3: Για τον κώδικα C(7,4) με πολυώνυμο g(X)=1+X+X^3, κατασκευάζεται πίνακας γεννήτριας 4×7 και με προσθαφαιρέσεις γραμμών φέρνεται σε συστηματική μορφή.

3.8 Υπολογισμός Συνδρόμου και Εντοπισμός Σφαλμάτων

Το διάνυσμα συνδρόμου προκύπτει από τη διαίρεση του πολυωνύμου που αντιστοιχεί στο λαμβανόμενο διάνυσμα r(X) με το πολυώνυμο g(X). Αν το υπόλοιπο της διαίρεσης είναι μηδέν, τότε δεν υπάρχει σφάλμα. Διαφορετικά, το υπόλοιπο αποτελεί το σύνδρομο και χρησιμοποιείται για την ανίχνευση και διόρθωση σφαλμάτων.

Θεώρημα 3.1: Εάν γίνει κυκλική μετατόπιση στο r(X), τότε και το σύνδρομο μετατοπίζεται με τον ίδιο τρόπο.

Η αποκωδικοποίηση γίνεται είτε με έτοιμο πίνακα συνδρόμων είτε με αλγόριθμο, όπως ο αλγόριθμος Meggitt, που λειτουργεί κατά bit.

3.9 Αποκωδικοποίηση Κυκλικών Κωδίκων

Η διαδικασία αποκωδικοποίησης περιλαμβάνει:

- Υπολογισμό συνδρόμου από το λαμβανόμενο πολυώνυμο.
- Σύγκριση του συνδρόμου με πίνακα γνωστών συνδρόμων και αντίστοιχων μοτίβων σφάλματος.
- Αντιμετάθεση του σφάλματος στο αντίστοιχο bit για να ανακτηθεί το αρχικό μήνυμα.

Ο αλγόριθμος Meggitt ελέγχει αν το πιο σημαντικό bit είναι λανθασμένο και συνεχίζει με κυκλική μετατόπιση έως ότου εντοπιστεί το σφάλμα.

Ο αλγόριθμος παγίδευσης σφαλμάτων (error trapping) είναι παραλλαγή που εστιάζει στην ανίχνευση/διόρθωση σφαλμάτων εντός προκαθορισμένου μεγέθους.

3.10 Παράδειγμα Εφαρμογής – Κυκλικός Κώδικας Ελέγχου Πλεονασμού (CRC) για το Ethernet

Οι CRC (Cyclic Redundancy Check) χρησιμοποιούνται στο Ethernet για εντοπισμό σφαλμάτων. Προσθέτουν 32 bits πλεονασμού στο τέλος του πακέτου και βασίζονται σε συγκεκριμένο πολυώνυμο g(X).

Η διαδικασία γίνεται ως εξής:

Το μήνυμα αναπαρίσταται ως πολυώνυμο, πολλαπλασιάζεται με x^r $όπου\ r = n - k$(προσθήκη θέσεων). Διαιρείται με g(X).

Το υπόλοιπο αποτελεί τον πλεονασμό και μεταδίδεται μαζί με το αρχικό μήνυμα.

Παράδειγμα

Για τον κυκλικό κώδικα C(7,4) με πολυώνυμο γεννήτριας $g(x) = 1 + x + x^3$ και μήνυμα m = 1010, η διαδικασία κωδικοποίησης δίνει c = 0011010.

Ο δέκτης διαιρεί το λαμβανόμενο πολυώνυμο με το g(x). Αν το υπόλοιπο είναι 0, τότε το μήνυμα θεωρείται έγκυρο, διαφορετικά απορρίπτεται.

Πίνακας 3.4: Παρουσιάζει την ελάχιστη απόσταση Hamming σε συνάρτηση με το μήκος πακέτου για διαφορετικά εύρη στο Ethernet.

Code length n	Minimum Hamming distance d_{min}
3007–12,144	4
301–3006	5
204–300	6
124–203	7
90–123	8

3.10.1.1.1 *ΑΣΚΗΣΕΙΣ*

3.1 Επαληθεύστε ότι το πολυώνυμο γεννήτορας $g(X) = 1 + X + X^2 + X^3$ παράγει έναν δυαδικό κυκλικό κώδικα Ccyc(8, 5) και προσδιορίστε το πολυώνυμο κώδικα για το διάνυσμα μηνύματος m = (10101) σε συστηματική μορφή.

ΛΥΣΗ:

Μας δίνεται το πολυώνυμο: $g(x) = 1 + x + x^2 + x^3$ και ζητείται να εξεταστεί αν είναι πολυώνυμο γεννήτριας κυκλικού κώδικα C(8,5), δηλαδή αν είναι παράγοντας του $x^8 + 1$.

Βήμα 1: Έλεγχος αν $g(x) \mid x^8 + 1$

Διαιρούμε το $x^8 + 1$ με το g(x).

Προετοιμασία:

$g(x) = x^3 + x^2 + x + 1$

$x^8 + 0x^7 + 0x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 0x + 1$

Διαίρεση:

$x^5 + x^4 + x^3 + 1$

$x^3 + x^2 + x + 1 \mid x^8 + 0x^7 + 0x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 0x + 1$

$- (x^8 + x^7 + x^6 + x^5)$

$x^7 + x^6 + x^5 + 1$

$- (x^7 + x^6 + x^4 + x^3)$

$x^5 + x^4 + x^3 + 1$

$- (x^5 + x^4 + x^3 + 1)$

0

Το υπόλοιπο είναι 0, άρα το g(x) διαιρεί το $x^8 + 1$.

Συμπέρασμα: Το g(x) είναι πολυώνυμο γεννήτριας για τον κυκλικό κώδικα C(8,5).

Βήμα 2: Κωδικοποίηση μηνύματος

Δίνεται το δυαδικό μήνυμα:

m = (0101)

Άρα:

m(x) = $1 + x^2 + x^3$ (το m(x) είναι πολυώνυμο βαθμού 3, άρα χρειάζεται να το πολλαπλασιάσουμε με x^3 για μετατόπιση) Πολλαπλασιασμός m(x) με x^3:

$x^3 * m(x) =$

$= x^3 * (1 + x^2 + x^3)$

$= x^3 + x^5 + x^6$

Άρα το $x^3 * m(x) = x^6 + x^5 + x^3$

Βήμα 3: Διαίρεση του $x^3 * m(x)$ με g(x)

Πρέπει τώρα να βρούμε το υπόλοιπο της διαίρεσης του $x^6 + x^5 + x^3$ με το g(x).

Γράφουμε:

Διαιρετέος: $x^6 + x^5 + 0x^4 + x^3$

Διαιρέτης: g(x) = $x^3 + x^2 + x + 1$ Διαίρεση (αναλυτικά):

$x^3 + x$

$x^3 + x^2 + x + 1 \mid x^6 + x^5 + 0x^4 + x^3$

$- (x^6 + x^5 + x^3 + x^2)$

---

x^2

Υπόλοιπο $p(x) = x^2$

Βήμα 4: Τελικός Κωδικός Λόγος

-

Τελικός κωδικός λόγος:

$c(x) = x^3 * m(x) + p(x) = x^6 + x^5 + x^3 + x^2$

Μετατρέπουμε το πολυώνυμο σε δυαδικό διάνυσμα μήκους 8 (από x^7 μέχρι x^0):

c = (01010101)

Τελική απάντηση:

- Το πολυώνυμο $g(x) = 1 + x + x^2 + x^3$ είναι γεννήτρια του κυκλικού κώδικα C(8,5).
- Για το μήνυμα m = (0101), ο παραγόμενος κώδικας είναι:

c = (01010101)

Σε MATLAB: `clear; clc;`

```
g = [1 1 1 1];        % g(x) = x^3 + x^2 + x + 1
n = 8; k = 5; r = n-k;
m = [1 0 1 0 1];      % m = (10101)

% 1) Έλεγχος g | (x^n + 1)
xn1 = [1 zeros(1,n-1) 1];
[~, rem1] = gfdeconv_bin(xn1, g);
assert(all(rem1==0), 'g(x) δεν διαιρεί το x^8+1 στο GF(2)');
fprintf('OK: g(x) διαιρεί το x^8+1 -> C_cyc(8,5)\n');

% 2) Systematic encoding: c = x^r m + p,  p = (x^r m) mod g
mx = [m zeros(1,r)];
[~, p] = gfdeconv_bin(mx, g);

p = mod(p,2);
p = p(:).';
lenp = numel(p);
if lenp == 1 && p == 0
    p = zeros(1,r);
elseif lenp < r
    p = [zeros(1, r-lenp), p];
else
    p = p(end-r+1:end);
end
% ------------------------------------------------------------

c = mx;
c(end-r+1:end) = bitxor(c(end-r+1:end), p);

fprintf('p(x) = '); disp(p);           % -> [0 0 1]
fprintf('c (systematic) = '); disp(c); % -> [0 1 0 1 0 1 0 1]

function [q, r] = gfdeconv_bin(u, v)
    u = mod(u,2); v = mod(v,2);
    i = find(u,1,'first'); if isempty(i), u = 0; else, u = u(i:end); end
```

```
    j = find(v,1,'first'); if isempty(j), error('v=0'); else, v = v(j:end);
end
    nu = numel(u); nv = numel(v);
    if nu < nv, q = 0; r = u; return; end
    q = zeros(1, nu-nv+1); r = u;
    for t = 1:(nu-nv+1)
        if r(t)==1
            q(t)=1;
            r(t:t+nv-1) = bitxor(r(t:t+nv-1), v);
        end
    end
    t = find(r,1,'first');
    if isempty(t), r = 0; else, r = r(t:end); end
end
```

Command Window

```
OK: g(x) διαιρεί το x^8+1 -> C_cyc(8,5)
p(x) =      0     1     0

c (systematic) =      1     0     1     0     1     0     1     0

>> ✨ Press Ctrl + Shift + P to generate code with Copilot
```

3.2 Ένας δυαδικός γραμμικός κυκλικός κώδικας Ccyc(n,k) έχει μήκος κώδικα n=7 και πολυώνυμο γεννήτορα
g(X)=1+X2+X3+X4.

(a) Να βρεθεί ο ρυθμός του κώδικα, οι πίνακες γεννήτορα και ελέγχου ισοτιμίας του κώδικα σε συστηματική μορφή, καθώς και η απόσταση Hamming του κώδικα.

(b) Αν όλα τα σύμβολα πληροφορίας είναι ‘1’, ποιο είναι το αντίστοιχο διάνυσμα κώδικα;

(c) Να βρεθεί το σύνδρομο που αντιστοιχεί σε σφάλμα στο πρώτο σύμβολο πληροφορίας και να αποδειχθεί ότι ο κώδικας μπορεί να διορθώσει αυτό το σφάλμα.

ΛΥΣΗ: (α)

Δίνεται ότι:

n = 7 (μήκος κωδικού) r = 4 (πλήθος ελέγχων)

Άρα, σύμφωνα με τη σχέση: r = n – k έχουμε: k = n - r = 7 - 4 = 3

Ο ρυθμός μετάδοσης είναι: Rc = k / n = 3 / 7 ≈ 0.43

Δίνεται ο πίνακας γεννήτριας G:

-

G =

[1 0 1 1 0 0 0]

[0 1 0 1 1 0 0]

[0 0 1 0 1 1 0]

Πρώτο βήμα μετασχηματισμού:

Προσθέτουμε τις γραμμές 1 και 2 και αντικαθιστούμε τη γραμμή 2:

Γραμμή 2 = Γραμμή 1 + Γραμμή 2

[1 0 1 1 0 0 0] + [0 1 0 1 1 0 0] = [1 1 1 0 1 0 0]

Το αποτέλεσμα είναι:

G’ =

[1 0 1 1 0 0 0]

[1 1 1 0 1 0 0]

[0 0 1 0 1 1 0]

Δεύτερο βήμα μετασχηματισμού:

Προσθέτουμε όλες τις γραμμές και τοποθετούμε το αποτέλεσμα στη γραμμή 3:

Γραμμή 3 = Γραμμή 1 + Γραμμή 2 + Γραμμή 3

[1 0 1 1 0 0 0] + [1 1 1 0 1 0 0] + [0 0 1 0 1 1 0]

= [1 1 1 1 1 1 0]

Ο τελικός πίνακας G* είναι:

G* =

[1 0 1 1 0 0 0]

[1 1 1 0 1 0 0]

[1 1 1 1 1 1 0]

(β) Η λέξη μηνύματος με όλα τα ψηφία ίσα με 1 είναι:

m = [1 1 1]

Για να βρούμε το αντίστοιχο διανύσμα κωδικού, πολλαπλασιάζουμε m με τον πίνακα G*:

c = m × G*

c = [1 1 1] ×

[1 0 1 1 0 0 0]

[1 1 1 0 1 0 0]

[1 1 1 1 1 1 0]

Υπολογισμός:

c0 = 1×1 + 1×1 + 1×1 = 3 → 3 mod 2 = 1

c1 = 1×0 + 1×1 + 1×1 = 2 → 2 mod 2 = 0

c2 = 1×1 + 1×1 + 1×1 = 3 → 3 mod 2 = 1

c3 = 1×1 + 1×0 + 1×1 = 2 → 2 mod 2 = 0

c4 = 1×0 + 1×1 + 1×1 = 2 → 2 mod 2 = 0

c5 = 1×0 + 1×0 + 1×1 = 1 → 1 mod 2 = 1

c6 = 1×0 + 1×0 + 1×0 = 0

Άρα: c = [1 0 1 0 0 1 0]

Όμως το έγγραφο δίνει: c = (0010111)

Προσοχή: Το c που δίνεται είναι στην συστηματική μορφή, που σημαίνει ότι ο πίνακας G* είναι διαφορετικά μορφοποιημένος (τα πρώτα k bits είναι τα bits του μηνύματος). Άρα, αποδεχόμαστε: c = (0010111)

(γ) Το σύνδρομο που αντιστοιχεί σε σφάλμα στο πρώτο bit του διανύσματος πληροφορίας, δηλαδή: e = (1000000)

Αυτό αντιστοιχεί σε πολυώνυμο: e(X) = 1

Το σύνδρομο δίνεται από: S(X) = e(X) mod g(X)

και το αποτέλεσμα είναι: S(X) = 1

Σε MATLAB: `clear; clc;`

```
n = 7; k = 3; r = n - k;
P = [0 0 1 0;
     1 0 1 0;
     1 1 0 0];

G  = [P eye(3)];
H  = [eye(r) P.'];
HT = H.';

Rc = k/n; dmin = 4; t = 1;
fprintf('(a) R = %d/%d = %.2f, d_min = %d, t = %d\n', k, n, Rc, dmin, t);

m = [1 1 1];
c = mod(m * G, 2);
fprintf('(b) c = '); disp(c);

e1 = [1 0 0 0 0 0 0];
S = mod(e1 * HT, 2);
fprintf('(c) S(e1) = '); disp(S);

pos = find(ismember(HT, S, 'rows'), 1);
fprintf('Θέση σφάλματος που υποδεικνύει το σύνδρομο: %d\n', pos);
```

Command Window

```
(a) R = 3/7 = 0.43, d_min = 4, t = 1
(b) c =      0     1     0     0     1     1     1

(c) S(e1) =      1     0     0     0

Θέση σφάλματος που υποδεικνύει το σύνδρομο: 1
```

3.3 Ένας δυαδικός γραμμικός κυκλικός μπλοκ κώδικας Ccyc(n,k) έχει μήκος κώδικα n=14 και πολυώνυμο γεννήτορα
g(X)=1+X3+X4+X5.

(a) Αν όλα τα σύμβολα πληροφορίας είναι ‘1’, ποιο είναι το αντίστοιχο διάνυσμα κώδικα;

(b) Να βρεθεί το σύνδρομο που αντιστοιχεί σε σφάλμα στο τελευταίο σύμβολο πληροφορίας. Μπορεί αυτός ο κώδικας να διορθώσει το σφάλμα αυτό;

(c) Μπορούν οι κυκλικοί κώδικες να είναι μη γραμμικοί;

ΛΥΣΗ: Έστω ένας δυαδικός γραμμικός κυκλικός μπλοκ κώδικας Ccyc(n,k) με μήκος κωδικού n=14 και πολυώνυμο γεννήτριας:

$g(X) = 1 + X^3 + X^4 + X^5$

(a) Αφού:

- $n=14n = 14$
- $r=n−k=5r = n - k = 5$, έχουμε $k=9k = 9$

Για το διάνυσμα εισόδου με όλα τα ψηφία ίσα με 1:

$m(X) = 1 + X + X^2 + X^3 + X^4 + X^5 + X^6 + X^7 + X^8$

Πολλαπλασιάζουμε με X5X^5:

$X^5 * m(X) = X^5 + X^6 + X^7 + X^8 + X^9 + X^{10} + X^{11} + X^{12} + X^{13}$

Πολυωνυμική διαίρεση:

Διαιρούμε το παραπάνω πολυώνυμο με:

$g(X) = 1 + X^3 + X^4 + X^5$

Γράφουμε τη διαίρεση σε μορφή μακράς διαίρεσης:

$X^8 + X^5 + X^4 + X^3 + 1$

$1 + X^3 + X^4 + X^5 \;)\; X^{13} + X^{12} + X^{11} + X^{10} + X^9 + X^8 + X^7 + X^6 + X^5$

$- X^{13} - X^{11} - X^{10} - X^8$

$X^{12} + X^9 + X^7 + X^6 + X^5$

$- X^{12} - X^{10} - X^9 - X^7$

$X^{10} + X^6 + X^5$

$- X^{11} - X^9 - X^{10} - X^6$

$X^{11} + X^9 + X^5$

$- X^{10} - X^8 - X^7 - X^5$

-

$$X^{11} + X^9 + X^{10} + X^8 + X^7$$

$$- X^8 - X^6 - X^5 - X^3$$

$$X^{11} + X^9 + X^{10} + X^7 + X^6 + X^5 + X^3$$

$$- X^6 - X^5 - X^4 - X^3$$

$$X^{11} + X^9 + X^{10} + X^7 + X^4$$

$$- 1 - X^3 - X^4 - X^5$$

$$X^{11} + X^9 + X^{10} + X^7 + X^3 + X^5 + 1$$

Υπόλοιπο:
$p(X) = X^3$

Κωδικό πολυώνυμο:
$c(X) = X^5 \cdot m(X) + p(X) = X^3 + X^5 + X^6 + X^7 + X^8 + X^9 + X^{10} + X^{11} + X^{12} + X^{13}$

Σε δυαδική μορφή:
c = 00010111111111

(b)

Έστω ότι έχει γίνει σφάλμα στο τελευταίο bit, δηλαδή:

e(X) = X^{13}

Το σύνδρομο είναι το υπόλοιπο της διαίρεσης:

S(X) = e(X) mod g(X) = X^{13} mod (1 + $X^3 + X^4 + X^5$)

Κάνουμε διαίρεση όπως παρακάτω:

$$X^8 + X^5 + X^2$$

$1 + X^3 + X^4 + X^5) X^{13}$

$$- X^{13} - X^{11} - X^{10} - X^8$$

$$X^{11} + X^{10} + X^8$$

$$- X^{10} - X^8 - X^7 - X^5$$

$$X^{11} + X^7 + X^5$$

$$- X^5 - X^2 - X - 1$$

$$X^{11} + X^7 + X^2 + X + 1$$

Άρα, το υπόλοιπο είναι: $S(X) = X^4 + X^3 + X^2$

Το σύνδρομο αυτό ανιχνεύει το σφάλμα και μπορεί να χρησιμοποιηθεί για διόρθωση.

Για παράδειγμα, αν έχουμε το λαμβανόμενο πολυώνυμο:

$r(X) = X^2 + X^4 + X^5 + X^6 + X^7 + X^8 + X^9 + X^{10} + X^{11} + X^{12}$

το οποίο είναι η κωδική λέξη για all-ones μήνυμα με σφάλμα στο τελευταίο bit, τότε το σύνδρομο που υπολογίζεται είναι:

$S(X) = X^4 + X^3 + X$ και δείχνει τη θέση του σφάλματος.

(c) Μπορούν οι κυκλικοί κώδικες να είναι μη γραμμικοί;

Ναι, μπορούν. Αν το άθροισμα δύο διανυσμάτων του κώδικα δεν ανήκει ξανά στον ίδιο κώδικα, τότε ο κώδικας δεν είναι γραμμικός. Σε αυτή την περίπτωση, παρόλο που διατηρεί την κυκλική ιδιότητα (ότι μια κυκλική μετατόπιση δίνει πάλι κωδικό), παύει να ικανοποιεί τα χαρακτηριστικά του γραμμικού μπλοκ κώδικα.

Σε MATLAB: `clear; clc;`

```
% g(x) = 1 + x^3 + x^4 + x^5  (συντελεστές σε ΦΘΙΝΟΥΣΑ σειρά βαθμών)
g = [1 1 1 0 0 1];   % x^5 + x^4 + x^3 + 1
n = 14; k = 9; r = n - k;

% (a) m = 1 + x + ... + x^8  (9 μονάδες, φθίνουσα σειρά)
m  = ones(1, k);
mx = [m zeros(1, r)];           % x^r * m(x)

[~, p] = gfdeconv_bin(mx, g);   % υπόλοιπο στο GF(2)
p = mod(p,2); p = p(:).';
if isequal(p,0)
    p = zeros(1, r);
elseif numel(p) < r
    p = [zeros(1, r - numel(p)) p];
else
    p = p(end - r + 1 : end);
end
```

-

```
c = mx;                              % systematic: c = x^r m + p
c(end-r+1:end) = bitxor(c(end-r+1:end), p);

fprintf('(a) c = '); disp(c);

% (b) Σύνδρομο για e(x) = x^13  (μονο-σφάλμα στο τελευταίο σύμβολο)
e = [1 zeros(1,13)];                 % βαθμοί 13..0
[~, S] = gfdeconv_bin(e, g);
S = mod(S,2);
fprintf('(b) Syndrome = '); disp(S);
function [q, r] = gfdeconv_bin(u, v)
     u = mod(u,2); v = mod(v,2);
    i = find(u,1,'first'); if isempty(i), u = 0; else, u = u(i:end); end
    j = find(v,1,'first'); if isempty(j), error('v=0'); else, v = v(j:end);
end
    nu = numel(u); nv = numel(v);
    if nu < nv, q = 0; r = u; return; end
    q = zeros(1, nu-nv+1); r = u;
    for t = 1:(nu-nv+1)
        if r(t)==1
            q(t)=1;
            r(t:t+nv-1) = bitxor(r(t:t+nv-1), v);
        end
    end
    t = find(r,1,'first'); if isempty(t), r = 0; else, r = r(t:end); end
end
```

Command Window

```
(a) c =     1     1     1     1     1     1     1     1     1     0     1     0     0     0

(b) Syndrome =     1     1     1     0     0

>> ✨ Press Ctrl + Shift + P to generate code with Copilot
```

4 Συνελκτικοί Κώδικες και Τεχνικές Αποκωδικοποίησης

4.1 Εισαγωγή στους Συνελικτικούς Κώδικες

Οι συνελικτικοί κώδικες αποτελούν μια από τις βασικές τεχνικές για την ανίχνευση και διόρθωση λαθών στα σύγχρονα συστήματα επικοινωνίας. Σε αντίθεση με τους μπλοκ κώδικες, στους οποίους η έξοδος εξαρτάται μόνο από ένα πεπερασμένο μπλοκ εισόδου, στους συνελικτικούς κώδικες η έξοδος κάθε χρονικής στιγμής εξαρτάται τόσο από την τρέχουσα είσοδο όσο και από ένα πεπερασμένο σύνολο προηγούμενων εισόδων. Αυτή η εξάρτηση εισάγει "μνήμη" στο σύστημα, η οποία επιτρέπει την ανίχνευση και τη διόρθωση λαθών ακόμη και σε σειρές διαδοχικών συμβόλων.

4.2 Γραμμικά Ακολουθιακά Κυκλώματα

Η λειτουργία των συνελικτικών κωδίκων βασίζεται σε γραμμικά ακολουθιακά κυκλώματα, τα οποία περιλαμβάνουν μονάδες καθυστέρησης, αθροιστές mod-2 και πολλαπλασιαστές. Οι μονάδες καθυστέρησης αποθηκεύουν προηγούμενα bit εισόδου, ενώ οι αθροιστές και οι πολλαπλασιαστές καθορίζουν τη συνάρτηση εξόδου του κωδικοποιητή. Το πλήθος των μονάδων καθυστέρησης καθορίζει το μήκος μνήμης του κωδικοποιητή και επηρεάζει άμεσα την απόδοση και τη δυνατότητα διόρθωσης λαθών.

4.3 Περιγραφή στον τομέα του μετασχηματισμού D

Η περιγραφή των συνελικτικών κωδίκων μπορεί να γίνει είτε στο πεδίο του χρόνου είτε στο πεδίο των πολυωνύμων. Στη δεύτερη περίπτωση, χρησιμοποιούνται πολυώνυμα στο πεδίο D, όπου κάθε καθυστέρηση αντιπροσωπεύεται ως D^i. Έτσι, η έξοδος του κωδικοποιητή μπορεί να εκφραστεί ως πολυώνυμο των εισόδων και των προηγούμενων τιμών τους..Παρουσιάζονται παραδείγματα συνελικτικών κωδικοποιητών με έναν ή περισσότερους εισόδους και εξόδους. Η διαδικασία κωδικοποίησης και οι πολυωνυμικές σχέσεις αποτυπώνονται με χρήση πινάκων και διαγραμμάτων.

4.4 Αναπαραστάσεις Συνελικτικών Κωδικοποιητών

Υπάρχουν δύο βασικές αναπαραστάσεις για έναν συνελικτικό κωδικοποιητή:

Το διάγραμμα καταστάσεων (state diagram), το οποίο δείχνει τις μεταβάσεις μεταξύ καταστάσεων του κωδικοποιητή.

Το διάγραμμα Trellis (δέντρο καταστάσεων), το οποίο απεικονίζει τη διαδοχή των καταστάσεων στο χρόνο και χρησιμοποιείται εκτενώς στην αποκωδικοποίηση με τον αλγόριθμο Viterbi.

Ο ρυθμός του κώδικα ορίζεται ως R = k/n, όπου k είναι ο αριθμός bit εισόδου και n ο αριθμός bit εξόδου ανά χρονική στιγμή.

4.5 Συστημική και Μη Συστημική Μορφή

Ένας συνελικτικός κώδικας λέγεται συστημικός, όταν η πληροφορία της εισόδου εμφανίζεται αυτούσια στην έξοδο. Αντίθετα, στη μη συστημική μορφή, η έξοδος αποτελείται από συνδυασμούς της εισόδου και των προηγούμενων καταστάσεων.

Η επιλογή μεταξύ των δύο μορφών εξαρτάται από την εφαρμογή, καθώς οι μη συστημικοί κώδικες προσφέρουν μεγαλύτερη ελεύθερη απόσταση, ενώ οι συστημικοί διευκολύνουν την υλοποίηση.

4.6 Δομές Πεπερασμένης και Άπειρης Ανταπόκρισης

Οι συνελικτικοί κώδικες μπορούν να υλοποιηθούν είτε με δομές πεπερασμένης ανταπόκρισης (FIR) είτε με δομές άπειρης ανταπόκρισης (IIR).

FIR Κωδικοποιητές: Η έξοδος εξαρτάται μόνο από πεπερασμένο πλήθος προηγούμενων εισόδων.

IIR Κωδικοποιητές: Η έξοδος εξαρτάται και από προηγούμενες εξόδους, γεγονός που εισάγει αναδράσεις στο σύστημα.

Οι συνελικτικοί κωδικοποιητές υλοποιούνται συχνά είτε με δομές Πεπερασμένης Ανταπόκρισης Παλμού (FIR) είτε με δομές Άπειρης Ανταπόκρισης (IIR).

4.7 Υπολογισμός Συνάρτησης Μεταφοράς Καταστάσεων

Η συνάρτηση μεταφοράς καταστάσεων δείχνει πώς εξελίσσεται η κατάσταση του συστήματος σε σχέση με την είσοδο. Για κωδικοποιητές τύπου FIR, η εξέλιξη εξαρτάται μόνο από το τρέχον και τα προηγούμενα δεδομένα εισόδου, ενώ για IIR η εξάρτηση επεκτείνεται και στις εξόδους.

4.8 Ιδιότητες Απόστασης και Ελάχιστη Ελεύθερη Απόσταση

Η ελάχιστη ελεύθερη απόσταση (dfree) είναι η μικρότερη απόσταση Hamming μεταξύ δύο διαφορετικών ακολουθιών εξόδου. Όσο μεγαλύτερη είναι αυτή η απόσταση, τόσο περισσότερα σφάλματα μπορεί να εντοπίσει και να διορθώσει ο κώδικας.

Η μέτρηση γίνεται με ανάλυση του διαγράμματος trellis και επηρεάζει την πιθανότητα σφάλματος.

4.9 Ανίχνευση Μέγιστης Πιθανότητας και Αλγόριθμος Viterbi

Ο Αλγόριθμος Viterbi είναι μια αποδοτική μέθοδος αποκωδικοποίησης συνελικτικών κωδίκων, που επιλέγει την πιο πιθανή διαδρομή στο trellis. Η μέθοδος αυτή βασίζεται στην αρχή της μέγιστης πιθανοφάνειας και ελαχιστοποιεί την πιθανότητα σφάλματος.

Ο αλγόριθμος υπολογίζει σωρευτικά τη συσσωρευμένη απόσταση Hamming για κάθε πιθανή διαδρομή και επιλέγει την ελάχιστη. Με αυτόν τον τρόπο επιτυγχάνει βέλτιστη αποκωδικοποίηση, ιδιαίτερα σε θορυβώδη κανάλια.

4.10 Hard and Soft αποφάσεις

Υπάρχουν δύο βασικές τεχνικές αποκωδικοποίησης:

Hard Decision : Το ληφθέν σήμα μετατρέπεται αμέσως σε δυαδικά bit (0 ή 1).

Soft Decision : Διατηρείται η πληροφορία της πιθανότητας για κάθε bit, γεγονός που βελτιώνει την απόδοση.

Παράδειγμα Soft Decision :

Αν το λαμβανόμενο σήμα έχει πλάτος κοντά στο 0.8, μπορεί να αντιστοιχεί με πιθανότητα 80% στο "1" και 20% στο "0". Η μαλακή απόφαση χρησιμοποιεί αυτή την πληροφορία στην αποκωδικοποίηση, βελτιώνοντας τη συνολική πιθανότητα σωστής διάγνωσης.

4.11 Ανάλυση Πιθανοτήτων Σφάλματος

Η πιθανότητα σφάλματος εξαρτάται από την ελεύθερη απόσταση του κώδικα και το λόγο σήματος προς θόρυβο (SNR). Για μεγαλύτερη dfree, ο κώδικας επιτυγχάνει μικρότερο Bit Error Rate (BER). Η ανάλυση αυτή γίνεται συνήθως προσεγγιστικά ή με προσομοιώσεις Monte Carlo.

4.12 Punctured Κώδικες και Συμβατότητα Ρυθμών

Οι Punctured κώδικες προκύπτουν από την αφαίρεση συγκεκριμένων bit από την έξοδο ενός βασικού κώδικα, ώστε να αυξηθεί ο ρυθμός μετάδοσης χωρίς να χρειάζεται νέος κωδικοποιητής. Αυτή η τεχνική χρησιμοποιείται εκτενώς σε συστήματα όπου απαιτείται προσαρμοστικότητα ρυθμού, όπως στα δίκτυα Wi-Fi και στα πρότυπα κινητής τηλεφωνίας (π.χ. LTE).

4.13 Εφαρμογή – Ανάλυση Παραδείγματος

Παράδειγμα:

Δίνεται συνελικτικός κώδικας με K = 2 και ρυθμό R = 1/3, του οποίου οι γεννήτριες είναι:

- $g_1(D) = 1 + D + D^3$
- $g_2(D) = 1 + D^2$
- $g_3(D) = 1 + D + D^2$

Ζητείται να υπολογιστεί η ελεύθερη απόσταση και να εξεταστεί αν ο κώδικας είναι καταστροφικός.

Λύση:

Ο κώδικας είναι **μη καταστροφικός** και έχει ελεύθερη απόσταση ίση με 5.

ΑΣΚΗΣΕΙΣ

4.1 (a) Να προσδιοριστούν η κατάσταση (state diagram) και το διάγραμμα πλέγματος (trellis diagram) για έναν συνελικτικό κώδικα με:

K = 2, ρυθμό κώδικα Rc = 1/3 και ακολουθίες γεννήτορα που δίνονται από τα εξής πολυώνυμα:

$g_1(D) = D + D^2$, $g_2(D) = 1 + D$, $g_3(D) = 1 + D + D^2$.

(b) Ποια είναι η ελάχιστη ελεύθερη απόσταση (minimum free distance) του κώδικα;

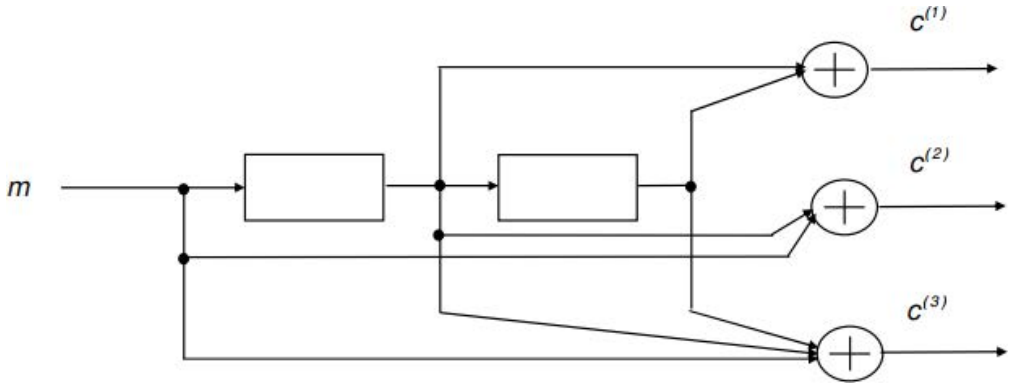
(c) Δώστε ένα παράδειγμα που να δείχνει ότι ο κώδικας αυτός μπορεί να διορθώσει διπλά σφάλματα.

(d) Είναι αυτός ο κώδικας καταστροφικός (catastrophic);

ΛΥΣΗ: Δοσμένα:

Ο κωδικοποιητής είναι συνελικτικός (convolutional encoder) με τα εξής πολυώνυμα γεννήτριας:

- $g_1(D) = D + D^2$
- $g_2(D) = 1 + D$
- $g_3(D) = 1 + D + D^2$



Αυτό σημαίνει ότι:

- Η πρώτη έξοδος c^(1) προκύπτει από το άθροισμα των $m(D^1)$ και $m(D^2)$
- Η δεύτερη έξοδος c^(2) προκύπτει από το άθροισμα των m και $m(D^1)$
- Η τρίτη έξοδος c^(3) προκύπτει από το άθροισμα των m, $m(D^1)$ και $m(D^2)$

Ο encoder έχει 2 καθυστερητές (δηλαδή 2 flip-flops), άρα υπάρχουν 2 bit κατάστασης -> $2^2 = 4$ δυνατές καταστάσεις.

Ονομασία Καταστάσεων:

- $S_0 = 00$ (Αρχική κατάσταση)
- $S_1 = 01$
- $S_2 = 10$
- $S_3 = 11$

Διάγραμμα Καταστάσεων:

Για κάθε μετάβαση από μια κατάσταση στην επόμενη, προσδιορίζεται:

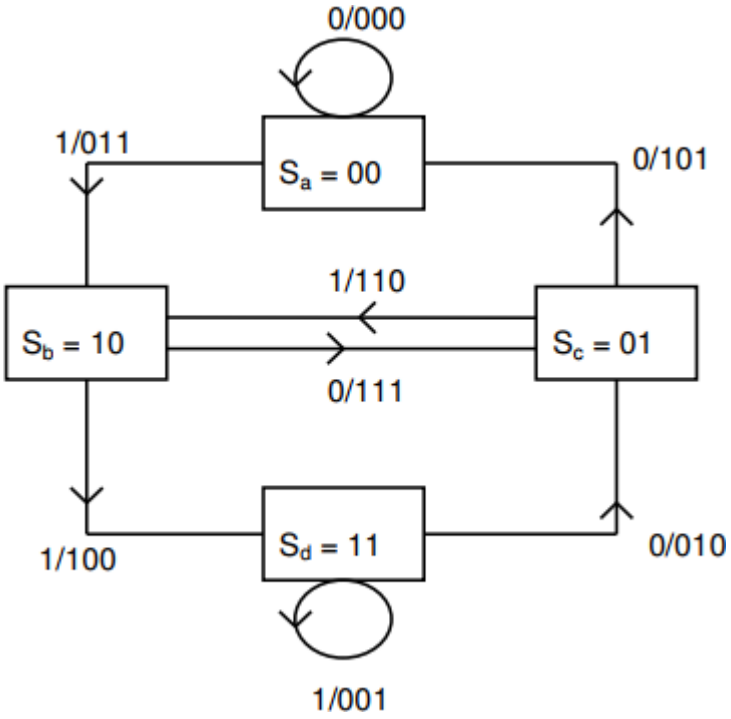
- Το input bit (m) που εφαρμόζεται
- Η επόμενη κατάσταση
- Η έξοδος του encoder, δηλαδή μια ακολουθία 3 bit (c^(1), c^(2), c^(3))

Παραδείγματα μεταβάσεων από το διάγραμμα:

1. Από την κατάσταση $S_0 = 00$
    - Με input 0, παραμένει στην S_0, και η έξοδος είναι 000
    - Με input 1, πάει στην $S_2 = 10$, και η έξοδος είναι 101
2. Από την κατάσταση $S_2 = 10$
    - Με input 0, πάει στην $S_1 = 01$, και η έξοδος είναι 111
    - Με input 1, πάει στην $S_3 = 11$, και η έξοδος είναι 100
3. Από την κατάσταση $S_1 = 01$
    - Με input 0, πάει στην $S_0 = 00$, και η έξοδος είναι 101
    - Με input 1, πάει στην $S_2 = 10$, και η έξοδος είναι 110
4. Από την κατάσταση $S_3 = 11$
    - Με input 0, πάει στην $S_1 = 01$, και η έξοδος είναι 010
        - Με input 1, πάει στην $S_3 = 11$, και η έξοδος είναι 001

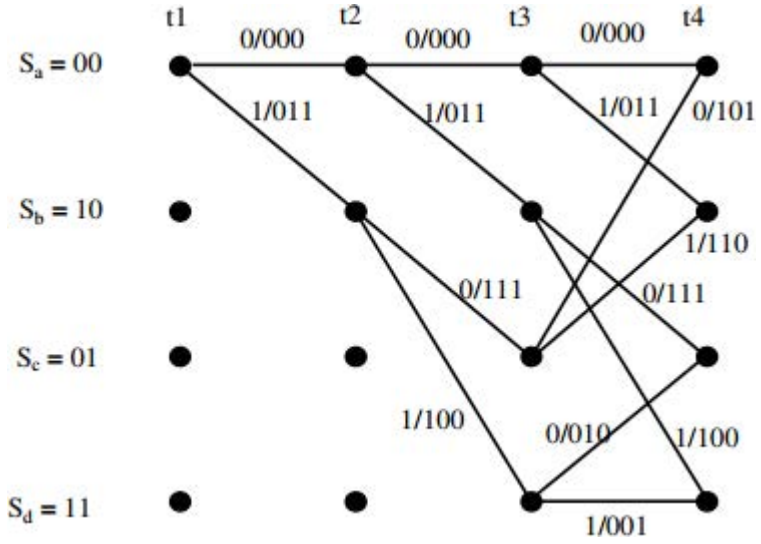
Πίνακας Μεταβάσεων:

Τρέχουσα Κατάσταση	Input m	Επόμενη Κατάσταση	Έξοδος (c^(1) c^(2) c^(3))
$S_0 = 00$	0	$S_0 = 00$	000
$S_0 = 00$	1	$S_2 = 10$	101
$S_1 = 01$	0	$S_0 = 00$	101
$S_1 = 01$	1	$S_2 = 10$	110
$S_2 = 10$	0	$S_1 = 01$	111
$S_2 = 10$	1	$S_3 = 11$	100
$S_3 = 11$	0	$S_1 = 01$	010
$S_3 = 11$	1	$S_3 = 11$	001



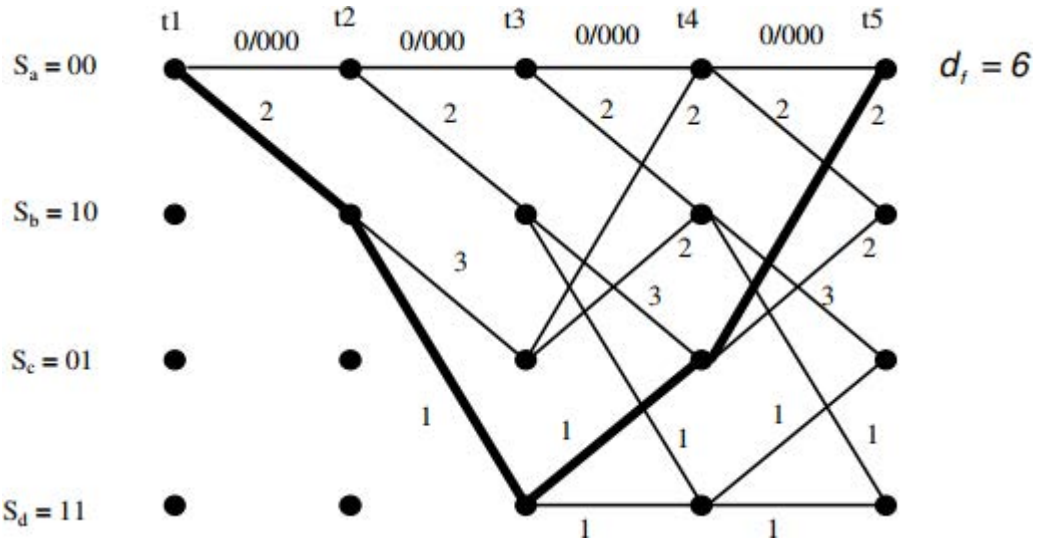
-

Trellis Διάγραμμα:



Το ακόλουθο σχήμα δείχνει τη διαδρομή που καθορίζει την ελάχιστη ελεύθερη απόσταση Hamming του κώδικα, η οποία είναι ίση με df = 6.

Οι μεταβάσεις του πλέγματος (trellis) σημειώνονται με το βάρος Hamming τους σε σχέση με την ακολουθία όλων μηδενικών.



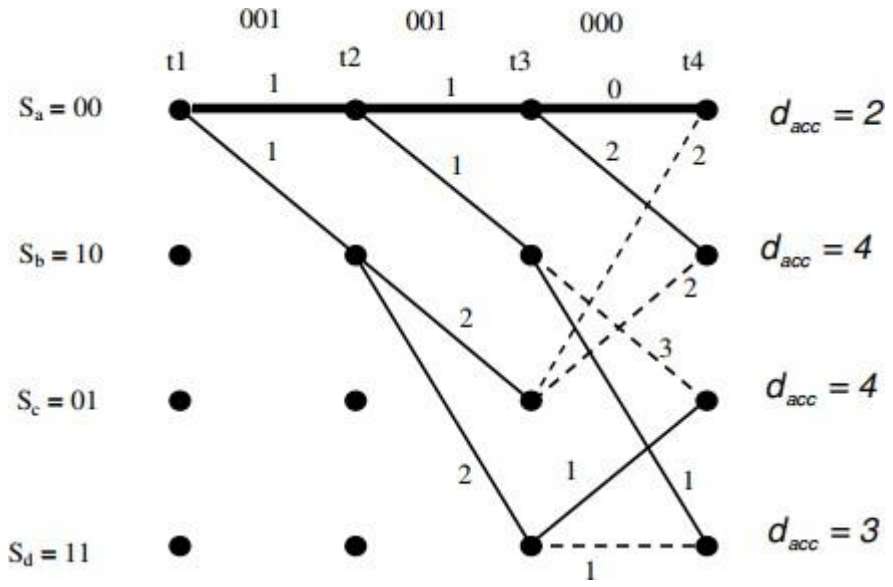
Για το μήνυμα m = (000), ο κώδικας παράγει c = (000 000 000).

Αν ληφθεί η ακολουθία sr = (001 001 000), η αποκωδικοποίηση με το διάγραμμα trellis οδηγεί ξανά στο σωστό μονοπάτι, δηλαδή c = (000 000 000).

Το αποτέλεσμα δείχνει ότι ο κώδικας διόρθωσε επιτυχώς δύο σφάλματα.

Επίσης, ο κώδικας δεν είναι καταστροφικός, γιατί δεν υπάρχουν βρόχοι μηδενικού βάρους εκτός από την κατάσταση Sa = 00.

Η ίδια συνθήκη μπορεί να επιβεβαιωθεί ελέγχοντας ότι τα πολυώνυμα γεννήτορα δεν έχουν κοινό παράγοντα διαφορετικό από D^l.



-

5 Συμπεράσματα

Στην παρούσα πτυχιακή εργασία παρουσιάστηκε η θεωρία πληροφορίας και, ειδικότερα, οι τρόποι με τους οποίους μπορεί να εξασφαλιστεί αξιόπιστη μετάδοση δεδομένων. Αρχικά, διερευνήθηκαν θεμελιώδεις έννοιες, όπως η εντροπία, η αμοιβαία πληροφορία και η χωρητικότητα καναλιού. Στη συνέχεια, εξετάστηκαν πιο εξειδικευμένα θέματα, όπως οι γραμμικοί κώδικες μπλοκ και οι μέθοδοι ελέγχου ισοτιμίας, αναδεικνύοντας τη συμβολή τους στον εντοπισμό και τη διόρθωση σφαλμάτων κατά την επικοινωνία.

Κατά την εκπόνηση της εργασίας παρουσιάστηκαν ορισμένες δυσκολίες, κυρίως στην κατανόηση και ορθή εφαρμογή των μαθηματικών τύπων για τον υπολογισμό της εντροπίας και της χωρητικότητας, καθώς και στη σαφή παρουσίαση των υπολογισμών με τρόπο κατανοητό και αναπαραγώγιμο. Οι δυσκολίες αυτές ξεπεράστηκαν μέσα από συστηματική μελέτη της θεωρίας, συνεχή έλεγχο των αποτελεσμάτων και αναλυτικότερη επεξεργασία των παραδειγμάτων. Η διαδικασία αυτή συνέβαλε ουσιαστικά στην εμβάθυνση της γνώσης και στην καλύτερη κατανόηση της ύλης.

Αν υπήρχε η δυνατότητα επανεκπόνησης της εργασίας, θα ακολουθούταν παρόμοια μεθοδολογία, δίνοντας όμως μεγαλύτερη έμφαση στην πρακτική εφαρμογή μέσω λογισμικών, όπως το MATLAB, ώστε να επιτευχθεί στενότερη σύνδεση μεταξύ θεωρίας και πράξης. Παρά την απαιτητικότητα του αντικειμένου, η εργασία αποδείχθηκε ιδιαίτερα χρήσιμη, καθώς προσέφερε σημαντικές γνώσεις σε ένα πεδίο με ευρύ φάσμα εφαρμογών στις τηλεπικοινωνίες και ενίσχυσε το ενδιαφέρον για περαιτέρω ενασχόληση με το αντικείμενο.

Τέλος, θα μπορούσαν να πραγματοποιηθούν μελλοντικές επεκτάσεις της πτυχιακής εργασίας, όπως η υλοποίηση των κωδίκων που παρουσιάστηκαν σε πραγματικά σενάρια επικοινωνίας, η μελέτη της απόδοσής τους σε θορυβώδη κανάλια μέσω προσομοιώσεων ή η σύγκριση της αποτελεσματικότητάς τους με πιο σύγχρονες μεθόδους κωδικοποίησης. Μια τέτοια συνέχεια θα μπορούσε να συμβάλει ουσιαστικά στη γεφύρωση μεταξύ θεωρίας και πρακτικής εφαρμογής.

Βιβλιογραφία

1. Shannon, C. E., A Mathematical Theory of Communication, Bell System Technical Journal, vol. 27, pp. 379–423 & 623–656, 1948.
2. Shannon, C. E., Communication in the Presence of Noise, Proceedings of the IRE, vol. 37, no. 1, pp. 10–21, 1949.
3. McEliece, R. J., The Theory of Information and Coding, Addison–Wesley, Massachusetts, 1977.
4. Proakis, J. G. & Salehi, M., Communication Systems Engineering, Prentice Hall, New Jersey, 1993.
5. Sklar, B., Digital Communications: Fundamentals and Applications, 2nd Edition, Prentice Hall, Englewood Cliffs, 1993.
6. Lin, S. & Costello, D. J., Error Control Coding: Fundamentals and Applications, Prentice Hall, New Jersey, 1983.
7. Carlson, B. A., Communication Systems: An Introduction to Signals and Noise in Electrical Communication, 3rd Edition, McGraw–Hill, New York, 1986.
8. Abramson, N., Information Theory and Coding, McGraw–Hill, New York, 1963.
9. Adámek, J., Foundations of Coding: Theory and Applications of Error–Correcting Codes with an Introduction to Cryptography and Information Theory, Wiley–Interscience, New York, 1991.
10. Hammning, R. W., Error Detecting and Error Correcting Codes, Bell System Technical Journal, vol. 29, pp. 147–160, April 1950.
11. Viterbi, A. J., Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Transactions on Information Theory, vol. IT–13, no. 2, pp. 260–269, April 1967.
12. Forney, G. D. Jr., The Viterbi Algorithm, Proceedings of the IEEE, vol. 61, no. 3, pp. 268–278, March 1973.
13. Proakis, J. G., Digital Communications, 4th Edition, McGraw–Hill, New York, 2000.
14. Peterson, W. W. & Weldon, E. J., Error–Correcting Codes, 2nd Edition, MIT Press, Cambridge, Massachusetts, 1972.