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

# Μελέτη και Προσομοίωση Τεχνικών Κωδικοποίησης Διαύλου για Σύγχρονα Συστήματα Ασυρμάτων Επικοινωνιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαρία Σ. Καραμανλή

**Επιβλέπων :** Φίλιππος Κωνσταντίνου
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2008

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

# Μελέτη και Προσομοίωση Τεχνικών Κωδικοποίησης Διαύλου για Σύγχρονα Συστήματα Ασυρμάτων Επικοινωνιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαρία Σ. Καραμανλή

**Επιβλέπων :** Φίλιππος Κωνσταντίνου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

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

Αθήνα, Οκτώβριος 2008

.....
Μαρία Σ. Καραμανλή

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

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

Μαρία Σ. Καραμανλή

Οκτώβριος 2008

# *Περίληψη*

Σκοπός αυτής της διπλωματικής εργασίας είναι η μελέτη συμπαγών τεχνικών κωδικοποίησης διαύλου για σύγχρονα συστήματα ασύρματων επικοινωνιών. Οι κώδικες ελέγχου σφάλματος χρησιμοποιούνται σήμερα σχεδόν σε ολόκληρο το φάσμα των πληροφοριών και επικοινωνιών, σε συστήματα αποθήκευσης και επεξεργασίας με σκοπό τη βέλτιστη επεξεργασία ελέγχου λάθους. Μελετώντας τους συμπαγείς κώδικες αναλύουμε εκτενέστερα μεθόδους κωδικοποίησης και αποκωδικοποίησης Reed-Solomon, ως ένα πολύ αποτελεσματικό και ευρέως χρησιμοποιούμενο κώδικα στην κατηγορία αυτή. Σημειώνονται τα πλεονεκτήματα αυτής της κλάσης των κωδίκων καθώς και των πιο πρόσφατων ευρέως χρησιμοποιούμενων συνελικτικών κωδίκων. Δίνεται ιδιαίτερη έμφαση στον προγραμματιστικό συνελικτικό αλγόριθμο Viterbi, όπου όπως διαπιστώθηκε, οι επιδόσεις του δεν πλησιάζουν τον ισχυρό συνδυασμό από συμπαγείς και συνελικτικούς κώδικες, που ονομάζονται συναλυσώμενοι (concatenated). Στη συνέχεια (στα κεφάλαια 3 και 4), αναλύουμε τη διαδικασία μοντελοποίησης και προσομοίωσης ασύρματου μέσου διάδοσης σε περιβάλλον λευκού προσθετικού θορύβου καθώς και σε περιβάλλον διαλείψεων. Για το σκοπό αυτό χρησιμοποιούμε το προγραμματιστικό περιβάλλον Matlab και τις κατάλληλες βιβλιοθήκες (toolboxes). Βασιζόμενοι στα αποτελέσματα της προσομοίωσης εξάγουμε πίνακες και γραφήματα που δίνουν πιο πλήρη εικόνα της επίδοσης των προαναφερθέντων κωδίκων στους διάφορους διαύλους. Τέλος αναφέρονται εφαρμογές των συμπαγών τεχνικών κωδικοποίησης σε σύγχρονα ασύρματα συστήματα καθώς και προτάσεις για μελλοντική εφαρμογή.

**Λέξεις Κλειδιά:** Κωδικοποίηση Διαύλου, Συμπαγής Κωδικοποίηση (block coding), Reed-Solomon, Συνελικτική Κωδικοποίηση (convolutional coding), Ασύρματες Επικοινωνίες, BER, PER, Προσομοίωση, Matlab

# *Abstract*

The aim of this thesis is to study block channel coding techniques for modern wireless communications systems. The control error codes are now used almost the entire range of information and communications, storage and processing systems for optimal control processing error. In addition, we extensively analyze encoding and decoding methods of the Reed-Solomon, as a highly effective and widely used code in the category of compact codes. Afterwards, we include some advantages of this class of codes and the latest widely used convolutional codes. We give emphasis on Convolutional Viterbi algorithm, whose performance does not come close to the powerful combination of compact and Convolutional codes, called concatenated codes. Then in Chapters 3 and 4, we analyze the modeling and simulation process for wireless communication channel in additive white noise and fading environment. Based on the results of the code in Matlab, tables and graphs give a better description of the above codes' performance on different channels. Finally, we mention compact coding techniques in wireless communication systems and proposals for future implementation



**Key words:** Channel coding, block coding, Reed-Solomon, Mobile Communications, BER, PER, Simulation, Matlab

# *Πίνακας Περιεχομένων*

# *Ευρετήριο Σχημάτων*

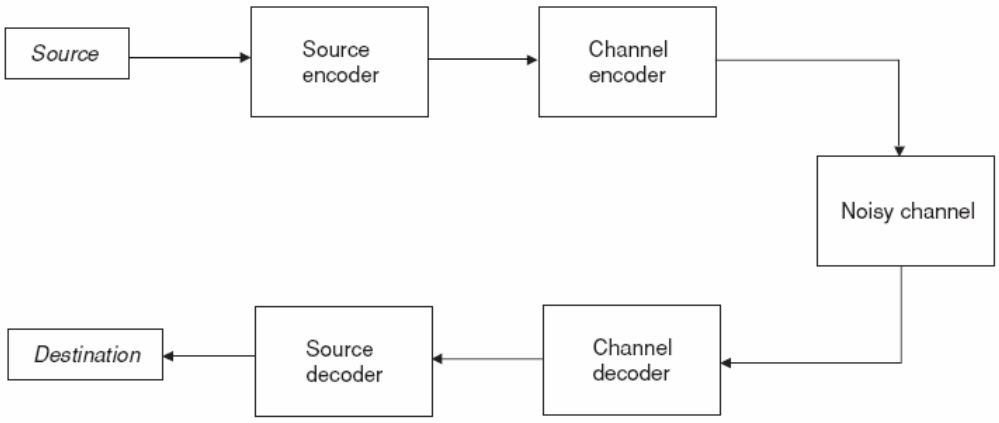
# *Ευρετήριο Πινάκων*

# 1. Εισαγωγή

## 1.1. Γενική Περιγραφή

Η κωδικοποίηση διαύλου ασχολείται με τις τεχνικές που χρησιμοποιούνται για την ενίσχυση του ψηφιακού σήματος ώστε να είναι λιγότερο ευάλωτο στις παρεμβολές του διαύλου που περιλαμβάνουν απλό προσθετικός θόρυβο (AWGN) ή και διάφορα είδη εξασθένησης (Fading). Γενικά, η κωδικοποίηση καναλιού κατηγοριοποιείται σε δύο μορφές, στην κυματομορφική κωδικοποίηση που απαιτεί τη χρήση νέων κυματομορφών για την βελτιωμένη ανίχνευση λαθών, και στη δομημένη ακολουθία. Η τελευταία κατηγορία, περιλαμβάνει τη χρήση πλεοναζόντων bits πληροφορίας που είναι υπεύθυνα για το καθορισμό των λαθών και την διόρθωσή τους. Σκοπός της παρούσας εργασίας, είναι η εξέταση των τριών επικρατέστερων τεχνικών κωδικοποίησης με δομημένη ακολουθία, πιο συγκεκριμένα της συμπαγής, συνελικτικής και συναλυσώμενης κωδικοποίησης. Όπως κάθε προσπάθεια καλύτερης απόδοσης ισχύος σήματος, έτσι και οι τεχνικές κωδικοποίησης διαύλου έχουν κάποιο κόστος, που σχετίζεται κυρίως με την απαίτηση μεγαλύτερου εύρους ζώνης. Όμως, η χρήση μεγάλης κλίμακας ολοκληρωμένων κυκλώματα (LSI) και τεχνικών υψηλής ταχύτητας ψηφιακής επεξεργασίας σήματος (OSP) έδωσαν τη δυνατότητα στη κωδικοποίηση διαύλου να παρέχει 10 dB βελτίωση επιδόσεων σε πολύ λιγότερο κόστος από ο, τι με τη χρήση άλλων μεθόδων, όπως η χρήση πομπών μεγαλύτερης ισχύος ή μεγαλύτερες κεραίες [1].

Τα βασικά στοιχεία ενός ψηφιακού συστήματος επικοινωνίας απεικονίζονται στο λειτουργικό διάγραμμα του σχήματος 1.1. Η έξοδος της πηγής μπορεί να είναι είτε αναλογικό σήμα, όπως το ηχητικό ή οπτικό, ή ένα ψηφιακό σήμα, που είναι διακριτό στο πεδίο του χρόνου. Σε ένα ψηφιακό σύστημα επικοινωνίας, το μήνυμα της πηγής μετατρέπεται σε μια ακολουθία δυαδικών ψηφίων. Η διαδικασία αυτής της μετατροπής της πηγής της πληροφορίας ονομάζεται κωδικοποίηση πηγής (source encoding) ή συμπίεση δεδομένων [9].

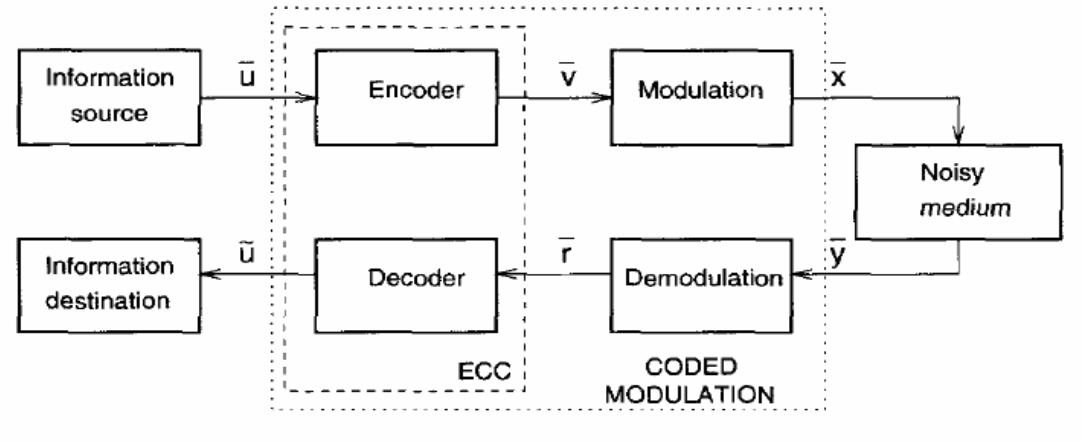



**Σχήμα 1.1** Βασικά Στοιχεία Ψηφιακού Συστήματος Επικοινωνίας

Η ακολουθία των δυαδικών ψηφίων από τον κωδικοποιητή πηγής, διέρχεται στον κωδικοποιητή καναλιού. Η δυαδική ακολουθία στην έξοδο του καναλιού κωδικοποίησης διέρχεται στον ψηφιακό διαμορφωτή, ο οποίος χρησιμεύει ως διεπαφή με το κανάλι επικοινωνίας. Το κανάλι επικοινωνίας είναι το φυσικό μέσο που χρησιμοποιείται για την αποστολή μηνύματος από τον πομπό στο δέκτη. Στην ασύρματη μετάδοση, το κανάλι μπορεί να είναι η ατμόσφαιρα (ελεύθερος χώρος). Από την άλλη πλευρά, στην τηλεφωνική μετάδοση συνήθως χρησιμοποιούνται μια ποικιλία μέσων όπως γραμμές καλωδίων ή καλώδια οπτικών ινών. Ανεξάρτητα από το μέσο που χρησιμοποιείται για τη μετάδοση της πληροφορίας, το ουσιώδες χαρακτηριστικό είναι ότι το μεταδιδόμενο μήνυμα έχει παραποιηθεί με τυχαίο τρόπο από μια ποικιλία πιθανών μηχανισμών, όπως πρόσθετο θερμικό θόρυβο που δημιουργείται από ηλεκτρικές συσκευές, και θόρυβο προερχόμενο από ανθρώπινες δραστηριότητες, όπως προερχόμενο από καύσιμα αυτοκινήτων. Ο ψηφιακός αποδιαμορφωτής επεξεργάζεται την παραποιημένη κυματομορφή και τη μετατρέπει σε μια αλληλουχία αριθμών που αντιπροσωπεύουν εκτιμήσεις των μεταδιδόμενων συμβόλων (δυαδικών ή M-ary συμβόλων). Αυτή η αλληλουχία των αριθμών διαβιβάζεται στο κανάλι αποκωδικοποίησης, το οποίο επιχειρεί να ανασυνθέσει την αρχική ακολουθία πληροφορίας από τη γνώση του κώδικα που χρησιμοποιεί ο κωδικοποιητής. Μέτρο της απόδοσης του αποδιαμορφωτή και αποκωδικοποιητή είναι η συχνότητα με την οποία εμφανίζονται σφάλματα κατά την αποκωδικοποιημένη ακολουθία. Συγκεκριμένα. η μέση πιθανότητα λάθους (bit-error) στην έξοδο του αποκωδικοποιητή είναι συνάρτηση των χαρακτηριστικών του κώδικα, του ποσού και του είδους του θορύβου, της προεπιλεγμένης μορφής διαμόρφωσης καθώς και άλλων παραγόντων που περικλείονται στο σύστημα ψηφιακής μετάδοσης της πληροφορίας [9].

## *1.2. Τεχνικές Κωδικοποίησης Διαύλου*

Η ιστορία των τεχνικών κωδικοποίησης διαύλου ξεκίνησε με την εισαγωγή των κωδίκων Hamming [Ham], την ίδια χρονική περίοδο με την καταλυτική εφαρμογή της Shannon [Sha],ενώ λίγο αργότερα εφευρέθηκαν οι κώδικες Golay [Gol]. Το Σχήμα 1.2 δείχνει το διάγραμμα ενός κανονικού συστήματος ψηφιακών επικοινωνιών. Η πληροφορία πηγής (source) και προορισμού (destination) θα περιλαμβάνει οποιοδήποτε σύστημα κωδικοποίησης που αντιστοιχεί στη φύση της πληροφορίας. Ο κωδικοποιητής δέχεται ως είσοδο τα σύμβολα της πληροφορίας από τη πηγή και προσθέτει πλεονάζοντα σύμβολα σε αυτά έτσι ώστε τα περισσότερα από τα λάθη, που έχουν εισαχθεί στην διαδικασία της διαμόρφωσης ενός σήματος διαβιβασθεί σε ένα θορυβώδη μέσο και αποδιαμορφωθεί, να μπορούν να διορθωθούν.




**Σχήμα 1.2** Σύστημα Ψηφιακών Επικοινωνιών

Συνήθως, το κανάλι, αναμένεται να είναι δείγματα μιας διαδικασίας προσθετικού θορύβου που προσθέτονται στα σύμβολα που διαμορφώνονται. Τα δείγματα θορύβου πρέπει να είναι ανεξάρτητα από την πηγή συμβόλων. Αυτό το μοντέλο είναι σχετικά εύκολο να παρακολουθείτε και περιλαμβάνει κανάλια πρόσθετου λευκού Gaussian θορύβου (AWGN) , επίπεδα κανάλια Rayleigh εξασθένισης, και δυαδικά συμμετρικά κανάλια (BSC). Στο τέλος του δέκτη, ο αποκωδικοποιητής χρησιμοποιεί τα πλεονάζοντα σύμβολα για να διορθώσει τα σφάλματα καναλιού. Σε περίπτωση ανίχνευσης σφάλματος, ο αποκωδικοποιητής μπορεί να θεωρηθεί ως ένας εκ νέου κωδικοποιητής του λαμβανόμενου μηνύματος και να ελέγξει εάν τα πλεονάζοντα σύμβολα, που παρήχθησαν εκ νέου, είναι τα ίδια με τα σύμβολα του λαμβανόμενου μηνύματος.

Στην κλασική θεωρία των τεχνικών κωδικοποίησης σφάλματος, ο συνδυασμός της διαμόρφωσης, του θορυβώδη μέσου και της αποδιαμόρφωσης μοντελοποιείται σαν ένα καθαρό, χωρίς μνήμη κανάλι (discrete memoryless channel) με είσοδο $v$ και έξοδο $r$ . Ένα παράδειγμα, αποτελεί η δυαδική μετάδοση πάνω από ένα AWGN κανάλι, το οποίο μοντελοποιείται ως δυαδικό συμμετρικό channel (BSC) με πιθανότητα λάθους του καναλιού p - ή πιθανότητα μετάβασης – ισοδύναμη της πιθανότητας *bit* λάθους για ένα δυαδικό σήμα πάνω από (AWGN) θόρυβο.

$$p = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Όπου $\quad Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{z^2}{2}} dz \quad , \quad x \geq 0$

Αποτελεί την Gaussian Q συνάρτηση, και Eb/No είναι ο λόγος σήματος προς θόρυβο (SNR) ανά bit. Το 1974, Ο Massey [Mas3] πρότειναν την θεώρηση του κώδικα διόρθωσης λάθους και της διαμόρφωσης ως μια ενιαία οντότητα, γνωστή στη σύγχρονη λογοτεχνία, ως κωδικοποιημένη διαμόρφωση. Η προσέγγιση αυτή παρέχει υψηλότερη

απόδοση και μεγαλύτερο κέρδος κωδικοποίησης από μια σειριακή συνένωση του κώδικα ελέγχου λάθους και της διαμόρφωσης.. Πολλές είναι οι μέθοδοι που συνδυάζουν κωδικοποίηση και διαμόρφωση, μεταξύ των οποίων είναι η Trellis κωδικοποιημένη διαμόρφωσης (Trellis-coded modulation - TCM) και η πολυεπίπεδη κωδικοποιημένη διαμόρφωση (multilevel coded modulation - MCM) . Σε ένα σύστημα κωδικοποιημένης διαμόρφωσης , η (soft-decision) έξοδος του καναλιού εκτελείται απευθείας από τον αποκωδικοποιητή. Αντίθετα, σε ένα κλασικό σύστημα , κωδικοποίησης ελέγχου η (hard-decision) έξοδος του αποδιαμορφωτή τροφοδοτεί ένα δυαδικό αποκωδικοποιητή.

Οι κώδικες μπορούν να συνδυαστούν με διάφορους τρόπους. Ένα παράδειγμα από σειριακή concatenation (concatenation με την κλασική έννοια) είναι το ακόλουθο που για χρόνια υπήρξε από τα πιο δημοφιλή. Αποτελείται από το συνδυασμό ενός εξωτερικού Reed-Solomon κώδικα, μέσω ενδιάμεσων interleaving, και ενός εσωτερικού δυαδικού συνελικτικού κώδικα. Αυτό το σύστημα έχει χρησιμοποιηθεί σε πολλές εφαρμογές, που κυμαίνονται από το χώρο των επικοινωνιών ως την ψηφιακή μετάδοση υψηλής ευκρίνειας τηλεόραση. Η βασική ιδέα είναι ότι η (soft-decision) αποκωδικοποίηση του συνελικτικού code παράγει εκρήξεις λαθών που μπορεί να χωριστούν σε μικρότερα κομμάτια μέσω της deinterleaving διαδικασίας και να αποκωδικοποιηθούν αποτελεσματικά από τον Reed-Solomon κώδικα. Οι Reed-Solomon κώδικες είναι μη δυαδικοί στους οποίους εφαρμόζονται σύμβολα αποτελούμενα από έναν αριθμό από bits και μπορούν να αντιμετωπίσουν πολλαπλές εκρήξεις λαθών. Η σειριακή concatenation έχει το πλεονέκτημα ότι απαιτεί δύο χωριστούς αποκωδικοποιητές, έναν για τον εσωτερικό κώδικα και έναν για τον εξωτερικό, αντί του ενός ενιαίου αλλά πολύ περίπλοκου αποκωδικοποιητή για την συνολικό κώδικα.

Ένας άλλος διαχωρισμός των τύπων κωδικοποίησης είναι ανάλογα με τον χώρο που λαμβάνει μέρος ο αλγόριθμος αποκωδικοποίησης. Στην περίπτωση του Hamming χώρου, παρουσιάζονται (hard-decision) αποκωδικοποιητές ενώ στον Euclidean χώρο έχουμε (soft-decision) για δυαδική μετάδοση αποκωδικοποιητές που επιτυγχάνουν τη μείωση της απαιτούμενης ανά bit μεταδιδόμενης ισχύος τουλάχιστον 2 dB (εν συγκρίσει με τον Hamming χώρο).

## *1.3. Δομή της Διπλωματικής Εργασίας*

Η εργασία θα ακολουθήσει την παρακάτω δομή:

- **Κεφάλαιο 2**: Αναφορά στη γενική θεωρία κωδικοποίησης καναλιού, κατηγοριοποίηση των τεχνικών κωδικοποίησης με ιδιαίτερη έμφαση στις διαδικασίες κωδικοποίησης των συμπαγών κωδίκων.
- **Κεφάλαιο 3**: Αναλυτική περιγραφή των διαδικασιών κωδικοποίησης και αποκωδικοποίησης του κυρίως κώδικα που μελετούμε, του Reed-Solomon

συμπαγούς κώδικα. Πλεονεκτήματα της χρήσης του κώδικα, εφαρμογές αυτού και ευρέως χρησιμοποιούμενοι συνδυασμοί του, ως ισχυροί concatenated κώδικες.

- **Κεφάλαιο 4**: Μοντελοποίηση των ασύρματων διαύλων επικοινωνίας παρουσία λευκού προσθετικού θορύβου και διαλείψεων. Προγραμματιστική υλοποίηση καναλιών μέσω της βοήθειας του matlab και ανάλυση της επίδοσης των κωδίκων υπό την παρουσία διαφορετικού περιβάλλοντος.
- **Κεφάλαιο 5:** Εκτίμηση της επίδοσης (BER) των τεχνικών κωδικοποίησης Reed-Solomon και του συνδυασμού αυτού με τον συνελικτικό Viterbi ως συνδυαστικός συνελυσώμενος κώδικας. Παρουσιάζονται τα αποτελέσματα της προσομοίωσης με περιγραφικούς πίνακες και γραφήματα.
- **Κεφάλαιο 6** : Συμπεράσματα της παρούσας εργασίας καθώς και προτάσεις για μελλοντική έρευνα.

# 2. Τεχνικές Κωδικοποίησης

### 2.1.1. Γενικά

Σύμφωνα με τον τρόπο με τον οποίο τα πλεονάζοντα bits προσθέτονται στο μήνυμα, οι τεχνικές κωδικοποίησης διαύλου μπορούν να διαιρεθούν σε δύο κατηγορίες: συμπαγείς και συνελικτικές. Και οι δύο τύποι συστημάτων κωδικοποίησης έχουν βρει πρακτική εφαρμογή. Ιστορικά, οι συνελικτικοί κώδικες είχαν προτιμηθεί λόγω της διαθεσιμότητα του αλγορίθμου Viterbi στην soft-decision αποκωδικοποίηση και την πεποίθηση για πολλά χρόνια πως οι block κώδικες δεν μπορούν να είναι αποτελεσματικοί στην soft-decision αποκωδικοποίηση. Ωστόσο, οι πρόσφατες εξελίξεις στη θεωρία και το σχεδιασμό των γραμμικών block αλγορίθμων στην soft-decision αποκωδικοποίηση έχουν βοηθήσει στην κατάρρευση αυτής της πεποίθησης. Επιπλέον, οι πιο γνωστές μέχρι σήμερα (αρχές του εικοστού πρώτο αιώνα) μέθοδοι κωδικοποίησης είναι οι συμπαγείς (block ) κώδικες (irregular low-density parity-check codes).

Οι block κώδικες επεξεργάζονται τη πληροφορία σε block-by-block βάση, επεξεργάζοντας κάθε μπλοκ από bits πληροφορίας, ανεξάρτητα από τα άλλα. Με άλλα λόγια, η block κωδικοποίηση είναι μια (memoryless) χωρίς μνήμη διαδικασία, υπό την έννοια ότι οι κωδικοποιημένες λέξεις (codewords) είναι ανεξάρτητες η μία από την άλλη. Αντίθετα, η έξοδος ενός συνελικτικού κωδικοποιητή δεν εξαρτάται μόνον από την τρέχουσα είσοδο πληροφορίας , αλλά και από προηγούμενες εισόδους ή εξόδους, είτε επεξεργάζονται σε block-by-block βάση είτε σε bit-by-bit βάση.

Θα πρέπει να σημειωθεί ότι οι block κώδικες έχουν στην πραγματικότητα μνήμη όταν αναφερόμαστε στην διαδικασία της bit-by-bit κωδικοποίησης και επεξεργαζόμενοι μια μόνο κωδικοποιημένη λέξη (codeword). Πιο πρόσφατα η διαφορά μεταξύ των block και συνελικτικών κωδίκων έχει καταλήξει λιγότερο καλά καθορισμένη, ειδικά μετά την πρόσφατη βελτιωμένη κατανόηση της trellis δομής των block κωδίκων και της tail-biting δομής των συνελικτικών κωδίκων [10].

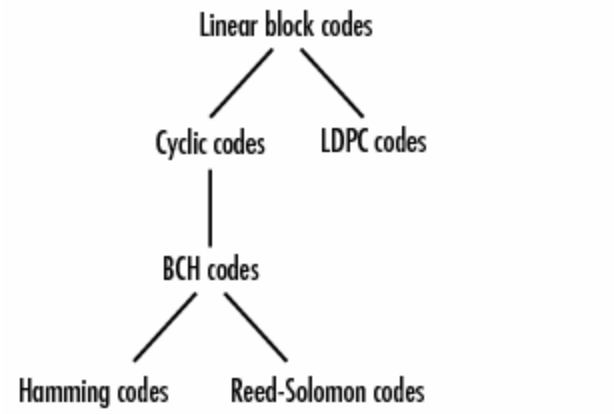
## 2.2. Κατηγοριοποίηση

Η προστασία των ψηφιακών πληροφοριών, με το κατάλληλο κωδικό ελέγχου λάθους, επιτρέπει την αποτελεσματική ανίχνευση και διόρθωση τυχόν λαθών που μπορεί να έχουν εμφανιστεί. Οι κώδικες ελέγχου σφάλματος χρησιμοποιούνται σήμερα σχεδόν σε ολόκληρο το φάσμα των πληροφοριών και επικοινωνιών, σε συστήματα αποθήκευσης και επεξεργασίας. Ταχεία πρόοδος των ηλεκτρονικών και οπτικών συσκευών και συστημάτων επέτρεψαν την υλοποίηση των πολύ ισχυρών κωδίκων με προσέγγιση τη βέλτιστη επεξεργασία ελέγχου λάθους. Επιπλέον, οι νέοι τύποι του κώδικα, καθώς και νέες μέθοδοι αποκωδικοποίησης, έχουν πρόσφατα αναπτυχθεί και αρχίζουν να εφαρμόζονται [3].

Οι Block κώδικες ήταν το πρώτο είδος κωδίκων ελέγχου λάθους που ανακαλύφθηκαν, στη δεκαετία περίπου από το 1940 έως το 1950. Ένα ιδιαίτερα χρήσιμο είδος μπλοκ κώδικα είναι ο κυκλικός κώδικα, με κύρια πρακτική εφαρμογή, την cyclic redundancy check (CRC) κωδικοποίηση για το πρότυπο Ethernet. Δύο πολύ αποτελεσματικές και ευρέως χρησιμοποιούμενες κλάσεις των κυκλικών κωδίκων αποτελούν οι Bose-Chaudhuri-Hocquenghem (OEB) και οι Reed-Solomon (RS) κώδικες, που ονομάστηκε έτσι μετά τους εφευρέτες τους. Οι OEB κώδικες μπορούν να είναι δυαδικοί ή μη δυαδικοί, αλλά οι RS κώδικες είναι μη δυαδικοί και είναι ιδιαίτερα αποτελεσματικοί σε ένα μεγάλο αριθμό σεναρίων ελέγχου λάθους όπως η εφαρμογή τους για τη διόρθωση σφαλμάτων σε οπτικό δίσκο (CD).

Μετά την ανακάλυψη των μπλοκ κωδίκων, ένα δεύτερο είδος κωδίκων ελέγχου λάθους προέκυψε, αρχικά αποκαλούνταν περιοδικοί και αργότερα συνελικτικοί. Η κωδικοποίηση και η αποκωδικοποίηση ακόμα και για ένα αρκετά ισχυρό συνελικτικό κώδικα συνεπάγεται απλές, επαναλαμβανόμενες, σχεδόν συνεχείς διεργασίες, που εφαρμόζεται σε μια πολύ απλή trellis αναπαράσταση του κώδικα, αντί για τις πιο πολύπλοκες block επεξεργασίας που φαίνεται να απαιτούνται στην περίπτωση ενός ισχυρού μπλοκ κώδικα. Αυτό καθιστά σχετικά εύκολη τη χρήση μέγιστης πιθανότητας , λιγότερο αυστηρής (soft) απόφασης, αποκωδικοποίηση με συνελικτικούς κώδικες, με τη μορφή του βέλτιστου Viterbi algorithm (VA). Δυστυχώς, όμως, ακόμη και ένας ισχυρός συνελικτικός κώδικας διαπιστώθηκε ως μη ικανός προς την επίτευξη των επιδόσεων κοντά στα όρια που δημοσιεύθηκαν για πρώτη φορά από το Shannon, ο πατέρας της θεωρίας της πληροφορίας, το 1948. Αυτό εξακολουθούσε να ισχύει μέχρι την επινόηση ισχυρών συνδυασμών από μπλοκ και συνελικτικούς κώδικες, που ονομάζονται συναλυσώμενοι κώδικες. Το επίτευγμα, από Berrou, Glavieux και Thitimajshima το 1993, ήταν να χρησιμοποιήσει ένα ιδιαίτερο είδος διαστρωμένης συνένωσης, σε συνδυασμό με την επαναληπτική soft-decision αποκωδικοποίηση. Όλες οι πτυχές αυτών των πολύ αποτελεσματικών συστημάτων κωδικοποίησης, λόγω της ισχύος των επαναληπτικών αλγορίθμων αποκωδικοποίησης, ονομάζονται turbo κώδικες.

Οι μπλοκ κώδικες είχαν βρεθεί να έχουν trellis παραστάσεις, έτσι ώστε να αποκωδικοποιούνται (soft-decision) με σχεδόν εξίσου καλές επιδόσεις όπως αυτές των συνελικτικών κωδίκων. Επίσης, θα μπορούσαν να χρησιμοποιηθούν σε αποτελεσματικά turbo συστήματα κωδικοποίησης. Η πολυπλοκότητα παρέμεινε ένα πρόβλημα, μέχρι που πρόσφατα, έγινε αντιληπτό ότι μια ιδιαίτερα απλή κλάση των κωδίκων, οι LDPC κώδικες που είχαν ανακαλυφθεί από τον Gallager το 1962, ήταν σε θέση να παρέχουν επιδόσεις πολύ καλύτερες από εκείνες των κωδίκων turbo όταν αποκωδικοποιούνται από κατάλληλο επαναληπτικό αλγόριθμο. Οι LDPC κώδικες είναι ιδιαίτερα αποτελεσματικό για χρήση στα δίκτυα επικοινωνιών. Παρακάτω δίνεται σχηματική κατηγοριοποίηση των block κωδίκων.

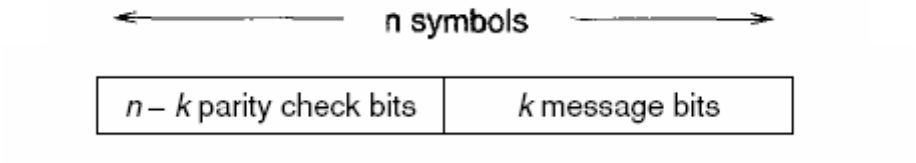



**Σχήμα 2.1** Κατηγοριοποίηση γραμμικών block κωδίκων

## *2.3. Συμπαγής Κωδικοποίηση*

### 2.3.1. Γενική Περιγραφή

Όλοι οι κώδικες διόρθωσης λάθους βασίζονται στην ίδια βασική αρχή: Πλεονάζοντα σύμβολα προσθέτονται στη πληροφορία, προκειμένου να διορθώσουν τυχόν σφάλματα που μπορεί να προκύψουν κατά τη διαδικασία της αποθήκευσης ή μετάδοσης μηνύματος. Σε μια βασική (αλλά και εφαρμοζόμενη) μορφή, πλεονάζοντα σύμβολα προσαρτώνται στα σύμβολα της πληροφορίας και προκύπτει η κωδικοποιημένη λέξη (codeword). Εφαρμόζοντας συστηματική κωδικοποίηση ενός block κώδικα, προκύπτει η κωδικοποιημένη λέξη (codeword) όπως παριστάνεται στο **Σχήμα 2.2**. Παρατηρούμε πως τα σύμβολα της πληροφορίας εμφανίζονται πάντα στις τελευταίες κ θέσεις της κωδικοποιημένης λέξης ενώ οι υπόλοιπες $n-k$ θέσεις περιέχουν σύμβολα όπως προκύπτουν έπειτα από εφαρμογή κατάλληλης συνάρτησης στα σύμβολα της πληροφορίας. Τα πλεονάζοντα αυτά σύμβολα χρησιμοποιούνται για ανίχνευση και διόρθωση των λαθών.




**Σχήμα 2.2** Συστήματα Ψηφιακών Επικοινωνιών

### 2.3.2. Ελάχιστη απόσταση ενός block κώδικα

Η ελάχιστη απόσταση $d_{\min}$ αποτελεί σημαντική παράμετρο του κώδικα, ειδικά στην περίπτωση ενός block κώδικα. Πριν από τον προσδιορισμό αυτής της παραμέτρου, άλλοι χρήσιμοι ορισμοί που σχετίζονται με την ελάχιστη απόσταση προϋποθέτονται.

Ο αριθμός των μη μηδενικών συντελεστών $c_i \neq 0$ ενός δοθέντος διανύσματος $c = (c_0, c_1, ..., c_{n-1})$ διαστάσεων (1 × n) ονομάζεται το βάρος, ή το **βάρος Hamming**, $w(c)$ του εν λόγω διανύσματος. Στην περίπτωση ενός διανύσματος που ορισμένο στο δυαδικό πεδίο GF (2), το βάρος είναι ο αριθμός των '1 ' που περιλαμβάνονται στο διάνυσμα.

**Η απόσταση Hamming** $d(c_1, c_2)$ μεταξύ δύο διανυσμάτων $c_1 = (c_{01,} c_{11}, ..., c_{n-1,1})$ και $c_2 = (c_{02,} c_{12}, ..., c_{n-1,2})$ είναι ο αριθμός των συντελεστών για τους οποίους τα δύο διανύσματα διαφέρουν.
Για παράδειγμα εάν $c_1 = (0011010)$ και $c_2 = (1011100)$ τότε $d(c_1, c_2) = 3$.

Σύμφωνα με τον παραπάνω ορισμό η απόσταση Hamming μπορεί να γραφτεί ως

$$d(c_1, c_2) = w(c_i \oplus c_j)$$

Για ένα δοθέν κώδικα, η ελάχιστη απόσταση μεταξύ όλων των πιθανών συνδυασμών δύο κωδίκων λέξεων μπορεί να υπολογιστεί ως εξής:

$$d_{\min} = \min\{d(c_i, c_j); c_i, c_j \varepsilon C_b; c_i \neq c_j\}$$

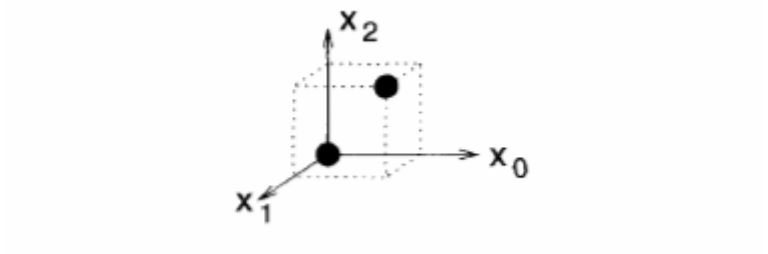
Δεδομένου ότι, σε γενικές γραμμές, οι μπλοκ κώδικες έχουν σχεδιαστεί γραμμικοί, η πρόσθεση δύο διανυσμάτων που ανήκουν στους μπλοκ κώδικες οδηγεί σε ένα άλλο διάνυσμα του κώδικα. Από αυτή την άποψη, κάθε κωδικοποιημένη λέξη (codeword) μπορεί να θεωρηθεί ως η προσθήκη δύο τουλάχιστον άλλων κωδίκων λέξεων. Δεδομένου ότι η απόσταση Hamming είναι ο αριθμός των θέσεων στις οποίες δύο διανύσματα διαφέρουν, και πως το βάρος του άθροισμα δύο διανυσμάτων είναι η απόσταση Hamming μεταξύ των δύο αυτών διανυσμάτων, τότε το βάρος μιας κωδικοποιημένης λέξης είναι ταυτόχρονα η απόσταση μεταξύ δύο άλλων διανυσμάτων του εν λόγω κώδικα [2]. Έτσι, η ελάχιστη τιμή του βάρους αξιολογώντας όλες τις κωδικοποιημένες λέξεις ενός κώδικα, με εξαίρεση το μηδενικό (all-zero) διάνυσμα, αποτελεί την ελάχιστη απόσταση του κώδικα:

$$d_{\min} = \min\{w(c_i \oplus c_j); c_i, c_j \varepsilon C_b; c_i \neq c_j\} = \min\{w(c_m); c_m \varepsilon C_b; c_m \neq 0\}$$

Επομένως, η ελάχιστη απόσταση ενός γραμμικού block κώδικα $C_b(n,k)$ είναι η ελάχιστη τιμή του βάρος μη μηδενικής κωδικοποιημένης λέξης αυτού του κώδικα.

**Παράδειγμα 2.1**: Ο απλούστερος κώδικας για διόρθωση λάθους είναι μια δυαδική επανάληψη κώδικα μήκους 3. Αυτό επαναλαμβάνει κάθε bit, τρεις φορές, έτσι ώστε το «0» να κωδικοποιείται σε διάνυσμα (000) και το «1» σε διάνυσμα (1 1 1). Δεδομένου ότι οι δύο κωδικοποιημένες λέξεις διαφέρουν σε τρεις θέσεις, η απόσταση Hamming μεταξύ τους είναι ίση με τρία.

Το **Σχήμα 2.3** είναι μια εικαστική απεικόνιση αυτού του κώδικα. Ο τρισδιάστατος δυαδικός χώρος αντιστοιχεί στο σύνολο των $2^3 = 8$ κορυφών του τρισδιάστατου κύβου. Η απόσταση Hamming μεταξύ των κωδικοποιημένων λέξεων (000) και (1 11) ισούται με το αριθμός των ακμών σε μια διαδρομή μεταξύ τους. Αυτό είναι ισοδύναμο με τον αριθμό των συντεταγμένων που χρειάζονται να αλλάξουν για ην μετατροπή του (000) σε (1 1 l), ή αντιστρόφως. Έτσι $d_H((000),(111)) = 3$, και δεδομένου ότι υπάρχουν μόνο δύο κωδικοποιημένες λέξεις σε αυτή την περίπτωση, $d_{\min} = 3$.




**Σχήμα 2.3** Τρισδιάστατος δυαδικός χώρος hamming

### 2.3.3. Βασικές έννοιες Block Κωδίκων

Υποθέτουμε ότι το σύνολο C ορίζει έναν δυαδικό γραμμικό κώδικα $(n,k,d_{\min})$. Εφόσον το C είναι ένα υποσύνολο διανυσμάτων k-διαστάσεων, έχει μια βάση τέτοια ώστε κάθε κωδικοποιημένη λέξη να αναπαριστάται ως γραμμικός συνδυασμός των στοιχείων της βάσης:

$$\bar{v} = u_0\overline{v_0} + u_1\overline{v_1} + \ldots + u_{k-1}\overline{v_{k-1}}$$

Όπου $u_i \in \{0,1\}, 1 \le i < k$.

Η παραπάνω εξίσωση μπορεί να γραφτεί υπό της μορφής γινομένου ενός πίνακα γεννήτριας $G$ με το διάνυσμα μηνύματος $(u_0, u_1, ..., u_{k-1})$, ως ακολούθως:

$$\bar{v} = \bar{u} * G$$

όπου

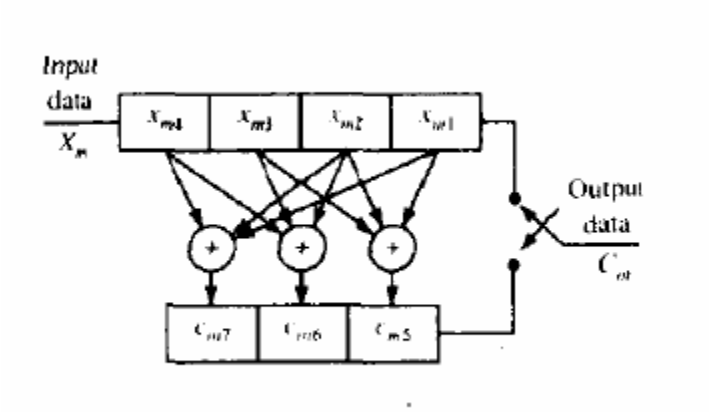
$$G = \begin{pmatrix} \overline{v_0} \\ \overline{v_1} \\ \cdots \\ \overline{v_{k-1}} \end{pmatrix} = \begin{pmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,k-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,k-1} \\ \cdots & \cdots & \cdots & \cdots \\ v_{k-1,0} & v_{k-1,1} & \cdots & v_{k-1,k-1} \end{pmatrix}$$

Σύμφωνα με το γεγονός πως ένας γραμμικός block κώδικας $(n,k)$ με $2^k$ κωδικοποιημένες λέξεις αποτελεί υπόχωρο του διανύσματος κ, οι γραμμές του **πίνακα γεννήτριας** $G$ πρέπει να είναι γραμμικά ανεξάρτητα διανύσματα στον υπόχωρο διάστασης κ. Επισημαίνουμε πως το σύνολο βάσης στον υπόχωρο δεν είναι μοναδικό και πως ο βαθμός του πίνακα γεννήτριας ισούται με κ. Η συστηματική μορφή του πίνακα γεννήτριας για έναν block $(n,k)$ κώδικα δίνεται από την παρακάτω ισότητα.

$$\mathbf{G} = [\mathbf{I}_k \vdots \mathbf{P}] = \left[\begin{array}{ccccc:ccc} 1 & 0 & 0 & \cdots & 0 & p_{11} & \cdots & p_{1n-k} \\ 0 & 1 & 0 & \cdots & 0 & p_{21} & \cdots & p_{2n-k} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & p_{k1} & \cdots & p_{kn-k} \end{array}\right]$$

Ο πίνακας $P$, διάστασης $k*(n-k)$ καθορίζει τα $(n-k)$ πλεονάζοντα bits, ή bits ελέγχου ισοτιμίας.

Ένας κωδικοποιητής γραμμικού block κώδικα, μπορεί να υλοποιηθεί χρησιμοποιώντας καταχωρητή k θέσεων μετάβασης και (n-k) modulo-2 αθροιστές συνδεδεμένους στα κατάλληλα στάδια του καταχωρητή. Οι (n-k) αθροιστές παράγουν τα bits ελέγχου ισοτιμίας που αποθηκεύονται προσωρινά σε ένα δεύτερο καταχωρητή μετάβασης μήκους (n-k). Τα bits εισόδου αποτελούν την πηγή πληροφορίας που εισέρχονται στον πρώτο καταχωρητή και τα bits που αποθηκεύονται στον δεύτερο καταχωρητή είναι γραμμικοί συνδυασμοί των k bits πληροφορίας και αποτελούν τα (n-k) πλεονάζοντα bits των κωδικοποιημένων λέξεων.
Η διαδικασία κωδικοποίησης που περιγράψαμε παραπάνω απεικονίζεται στο **Σχήμα 2.4**.




**Σχήμα 2.4** Διαδικασία κωδικοποίησης για ένα γραμμικό block κώδικα.

Όπου C αποτελεί χώρο διανυσμάτων k-διαστάσεων. Αντίστοιχα στον συμπληρωματικό χώρο $C^T$ (n - k) διαστάσεων, ορίζεται ο **πίνακας ισοτιμίας H** όπου ισχύει $G * H^T = 0$ , και συγκεκριμένα για κάθε κωδικοποιημένη λέξη $\overline{v_i}$ ισχύει

$$\overline{v_i} * H^T = 0 .$$

Από τη παραπάνω σχέση συμπεραίνουμε πως οποιαδήποτε κωδικοποιημένη λέξη $\overline{v_i}$ είναι ορθογώνια σε κάθε γραμμή του πίνακας ισοτιμίας $H$ και ακολούθως ισχύει

$$H = [-P^T : I_{n-k}]$$

Το σύμβολο ‘’-’’ από την παραπάνω σχέση μπορεί να αφαιρεθεί όταν αναφερόμαστε σε δυαδικούς κώδικες αφού η modulo-2 αφαίρεση ισοδυναμεί με modulo-2 πρόσθεση.

Κατά την διαδικασία της αποκωδικοποίησης χρήσιμη είναι η έννοια του πρότυπου πίνακα (standard array) στον οποίο παρουσιάζονται όλες οι πιθανές ληφθείσες κωδικοποιημένες λέξεις όπως προκύπτουν από όλους τους πιθανούς συνδυασμούς αθροίσματος των απεσταλμένων κωδικοποιημένων λέξεων με διανύσματα προσθετικού θορύβου.

| | | | | |
|---|---|---|---|---|
| $\bar{s}$ | $\bar{u}_0 = \bar{0}$ | $\bar{u}_2$ | $\cdots$ | $\bar{u}_{k-1}$ |
| $\bar{0}$ | $\bar{v}_0 = \bar{0}$ | $\bar{v}_1$ | $\cdots$ | $\bar{v}_{2^k-1}$ |
| $\bar{s}_1$ | $\bar{e}_1$ | $\bar{e}_1 + \bar{v}_1$ | $\cdots$ | $\bar{e}_1 + \bar{v}_{2^k-1}$ |
| $\bar{s}_2$ | $\bar{e}_2$ | $\bar{e}_2 + \bar{v}_1$ | $\cdots$ | $\bar{e}_2 + \bar{v}_{2^k-1}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $\bar{s}_{2^{n-k}-1}$ | $\bar{e}_{2^{n-k}-1}$ | $\bar{e}_{2^{n-k}-1} + \bar{v}_1$ | $\cdots$ | $\bar{e}_{2^{n-k}-1} + \bar{v}_{2^k-1}$ |

**Σχήμα 2.5** standard array για έναν δυαδικό γραμμικό block κώδικα

Ο standard array περιλαμβάνει $2^{n-k}$ γραμμές και $2^k + 1$ στήλες. Χρήσιμη είναι επίσης η έννοια του συνδρόμου στην διαδικασία της αποκωδικοποίησης. Αν υποθέσουμε πως $\bar{r} = \bar{v} + \bar{e}$ η ληφθείσα κωδικοποιημένη λέξη, τότε:

$$\bar{s} = \bar{r} * H^T = (\bar{v} + \bar{e}) * H^T = \bar{e} * H^T$$

Όπου $\bar{s}$ ονομάζεται το $(n-k)$ διαστάσεων διάνυσμα συνδρόμου λάθους, που έχει μηδενικές συνιστώσες όταν ικανοποιείται ο έλεγχος ισοτιμίας και μη μηδενικές συνιστώσες στην αντίθετη περίπτωση. Πρέπει να τονίσουμε ότι το σύνδρομο $\bar{s}$ είναι χαρακτηριστικό του πρότυπου διανύσματος λάθους, και όχι του απεσταλμένου μηνύματος. Λαμβανομένου υπόψιν του αριθμού των πιθανών λαθών $2^n$ εν συγκρίσει με τον αριθμό ων παραγόμενων συνδρόμων $2^{n-k}$, καταλήγουμε στην ύπαρξη διανυσμάτων λάθους που αντιστοιχούν στο ίδιο διάνυσμα συνδρόμου.

Κατά την **διαδικασία της αποκωδικοποίησης** ενός γραμμικού block κώδικα, ανατρέχουμε στον standard array και συγκεκριμένα στην αντιστοίχιση του διανύσματος ελάχιστης απόστασης λάθους με την τιμή του διανύσματος συνδρόμου διόρθωσης λάθους όπως υπολογίζεται από την σχέση $\bar{s} = \bar{r} * H^T$. Κατόπιν παράγουμε το αποκωδικοποιημένο μήνυμα από την εξίσωση:

$$\bar{c} = \bar{r} + \bar{\hat{e}}$$

Όπου $\bar{r}$ είναι ο απεσταλμένο μήνυμα και $\bar{\hat{e}}$ είναι το μοντέλο ελάχιστης απόστασης λάθους όπως συμπληρώνεται για κάθε πίνακα ξεχωριστά ανάλογα με τα χαρακτηριστικά του κώδικα και την ικανότητα διόρθωσης λάθους.

**Παράδειγμα 2.2**: Παράδειγμα αποκωδικοποίησης του γραμμικού block $(5,2)$ κώδικα, με προσθήκη πραγματικού διανύσματος λάθους $\bar{e} = [1 \quad 0 \quad 1 \quad 0 \quad 0]$ και standard array όπως παρατίθεται παρακάτω:

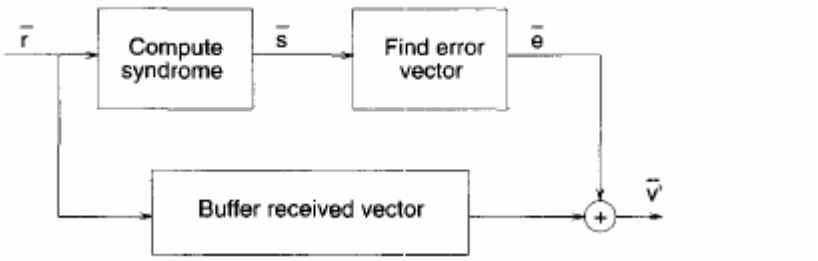
**Πίνακας 2.1** standard array του γραμμικού block $(5,2)$ κώδικα

| Syndrome | Error pattern |
|---|---|
| 0 0 0 | 0 0 0 0 0 |
| 0 0 1 | 0 0 0 0 1 |
| 0 1 0 | 0 0 0 1 0 |
| 1 0 0 | 0 0 1 0 0 |
| 0 1 1 | 0 1 0 0 0 |
| 1 0 1 | 1 0 0 0 0 |
| 1 1 0 | 1 1 0 0 0 |
| 1 1 1 | 1 0 0 1 0 |

Όπως παρατηρούμε από τον παραπάνω πίνακα, η ικανότητα διόρθωσης λάθους του κώδικα περιορίζεται στην διόρθωση όλων των λαθών βάρους 1 και στην διόρθωση δύο

από τα λάθη βάρους 2. Η λογική αυτή στηρίζεται στο ότι ο αριθμός των γραμμών του standard array είναι προκαθορισμένος και ίσος με $2^{n-k} = 2^{5-2} = 2^3 = 8$. Συνεπώς αρχίζοντας με το μη μηδενικό διάνυσμα λάθους στην πρώτη γραμμή, συνεχίζοντας με όλους τους πιθανούς συνδυασμούς βάρους 1, έχουμε στη διάθεσή μας να εισάγουμε μόνο 2 τελευταίες γραμμές λάθους βάρους 2 ($\bar{\hat{e}} = [1 \quad 1 \quad 0 \quad 0 \quad 0], \bar{\hat{e}} = [1 \quad 0 \quad 0 \quad 1 \quad 0]$).
Το σύνδρομο όπως παράγεται από τη παραπάνω σχέση $\bar{s} = \bar{e} * H^T$ ισούται με $\bar{s} = [0 \quad 0 \quad 1]$. Για την τιμή του συνδρόμου αυτού από το παραπάνω standard array προκύπτει διάνυσμα λάθους $\bar{\hat{e}} = [0 \quad 0 \quad 0 \quad 0 \quad 1]$ και κατόπιν από τη δοσμένη σχέση $\bar{c} = \bar{r} + \bar{\hat{e}}$ παράγουμε το αποκωδικοποιημένο μήνυμα. Στο παράδειγμά μας, που ο προσθετικός θόρυβος είναι βάρους 1 περιμένουμε το αποκωδικοποιημένο μήνυμα να ισούται με το κωδικοποιημένο και να έχουμε ακριβή διόρθωση του λάθους.

Η διαδικασία hard-decision αποκωδικοποίησης ενός block κώδικα περιγράφεται παρακάτω σύμφωνα με το παρακάτω σχήμα:




**Σχήμα 2.6** Δομή hard-decision αποκωδικοποίησης ενός block κώδικα

Για μια hard-decision διαδικασία αποκωδικοποίησης, η εκτίμηση του πιο πιθανού διανύσματος λάθους σε πρακτικές εφαρμογές δεν είναι δυνατή με υλοποίηση αντιστοίχησης μέσω του standard array. Για τον λόγο αυτό διάφοροι τρόποι υπολογισμού του διανύσματος λάθους επιχειρούνται. Ένας από αυτούς είναι η επίλυση της εξίσωσης $\bar{s} = \bar{e} * H^T$ ως προς $\bar{\hat{e}}$:

$$\bar{\hat{e}} = \bar{s} * (H^T)'$$

Έτσι ώστε $H' * (H^T)' = I_n$.
Επιπλέον, η διαδικασία παραγωγής συνδρόμου και αποκωδικοποίησης ενός κυκλικού block κώδικα περιγράφεται μέσω καταχωρητή και με πράξεις μεταξύ πολυωνύμων. Λεπτομερέστατα η διαδικασία αυτή αναλύεται στην ενότητα του Reed-Solomon κώδικα ο οποίος ανήκει στην κατηγορία αυτή.

## *2.4. Συνελικτική Κωδικοποίηση*

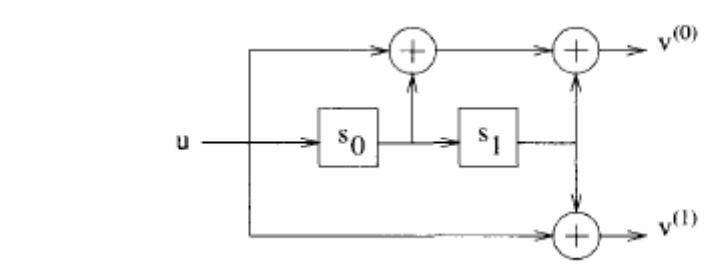
### 2.4.1. Γενική Περιγραφή

Ένας συνελικτικός κώδικας είναι κώδικας διόρθωση λάθους που επεξεργάζεται πληροφορίες σειριακά ή διαδοχικά, σε σύντομα μήκους μπλοκ. Ένας συνελικτικός κωδικοποιητής έχει μνήμη, με την έννοια ότι τα σύμβολα εξόδου δεν εξαρτώνται μόνο από τα σύμβολα εισόδου, αλλά και από προηγούμενες καταστάσεις εισροών ή εκροών. Με άλλα λόγια, ο κωδικοποιητής είναι ένα συνεχές κύκλωμα ή μια μηχανή πεπερασμένου αριθμού καταστάσεων. Στα προγράμματα ηλεκτρονικών υπολογιστών που εφαρμόζουν του αλγορίθμου Viterbi και άλλες trellis διαδικασίες αποκωδικοποίηση, βρέθηκε πίνακας μετάβασης, υποδεικνύοντας τη σχέση μεταξύ της εισόδου, της προηγούμενη και της τρέχουσας κατάστασης, καθώς και της τρέχουσα παραγόμενης εξόδου. Στη θεωρία, οι διαδοχικές ακολουθίες που παράγουν έναν συνελικτικό κώδικα έχουν άπειρη διάρκεια ενώ στην πράξη, η τρέχουσα κατάσταση ενισχύεται περιοδικά σε μια γνωστή κατάσταση και οι ακολουθίες του κώδικα παράγονται σε μορφή block ακολουθιών.

Γενικά ένας συνελικτικός κωδικοποιητής με ρυθμό k/n αποτελείται από k καταχωρητές μετάβασης, έναν για κάθε bit εισόδου πληροφορίας και n κωδικοποιημένα bit εξόδου ως γραμμικοί συνδυασμοί των περιεχομένων των καταχωρητών και των bit της πληροφορίας εισόδου.

### 2.4.2. Βασικές έννοιες συνελικτικών κωδίκων

Για μια λεπτομερή περιγραφή των συνελικτικών κωδίκων εξετάζουμε τον κωδικοποιητή του σχήματος με ρυθμό κωδικοποίησης ½ , που ισοδυναμεί με παραγωγή 2 bit εξόδου για κάθε bit εισόδου πληροφορίας.




**Σχήμα 2.7** συνελικτικός κωδικοποιητής με ρυθμό κωδικοποίησης ½

Το συνολικό μήκος των καταχωρητών μετάβασης στον κωδικοποιητή αναφέρεται ως μνήμη. Ως ετικέτες κατάστασης I αναφερόμαστε στους ακεραίους που σχετίζονται με την δυαδική αναπαράσταση των περιεχομένων μνήμης σύμφωνα με την σχέση:

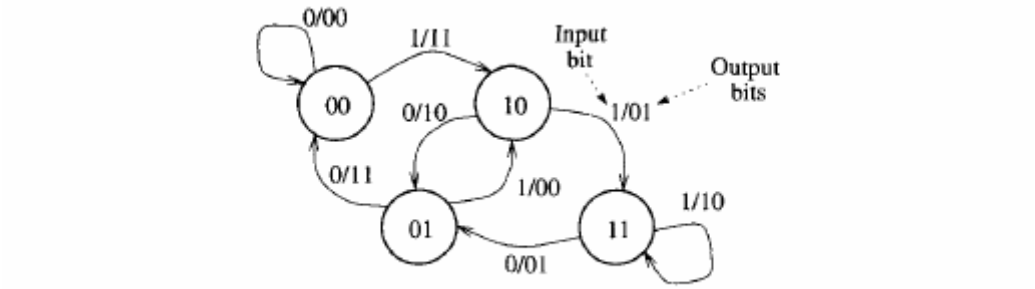
$$I = \sum_{j=0}^{m-1} s_j[i]2^{\tilde{m}-1-j}.$$

Το περιορισμένο μήκος, το οποίο για ρυθμό κωδικοποίησης ½ ισούται με $K = m + 1$, ορίζεται ως ο αριθμός των εισόδων $(u[i], u[i-1], ..., u[i-m])$ που επηρεάζει την έξοδο $(v^{(0)}[i], ..., v^{(n-1)}[i])$ την χρονική στιγμή i.

**Πίνακας 2.2** bits εισόδου, καταστάσεις μεταφοράς και bits εξόδου

| Initial state $s_0[i]s_1[i]$ | Information $u[i]$ | Final state $s_0[i+1]s_1[i+1]$ | Outputs $v^{(0)}[i]v^{(1)}[i]$ |
|---|---|---|---|
| 00 | 0 | 00 | 00 |
| 00 | 1 | 10 | 11 |
| 01 | 0 | 00 | 11 |
| 01 | 1 | 10 | 00 |
| 10 | 0 | 01 | 10 |
| 10 | 1 | 11 | 01 |
| 11 | 0 | 01 | 01 |
| 11 | 1 | 11 | 10 |

Ένας συνελικτικός κωδικοποιητής με m-μνήμη και 1/n ρυθμό κωδικοποίησης μπορεί να παρασταθεί με διάγραμμα καταστάσεων όπου ο αριθμός των καταστάσεων θα ισούται με $2^m$. Στο παράδειγμά μας, όπου έχουμε ένα μόνο bit πληροφορίας, προκύπτουν δύο κλάδοι εισόδου-εξόδου με ετικέτες κατάστασης $u_{[i]} / v^{(0)}[i], ..., v^{(n-1)}[i]$.



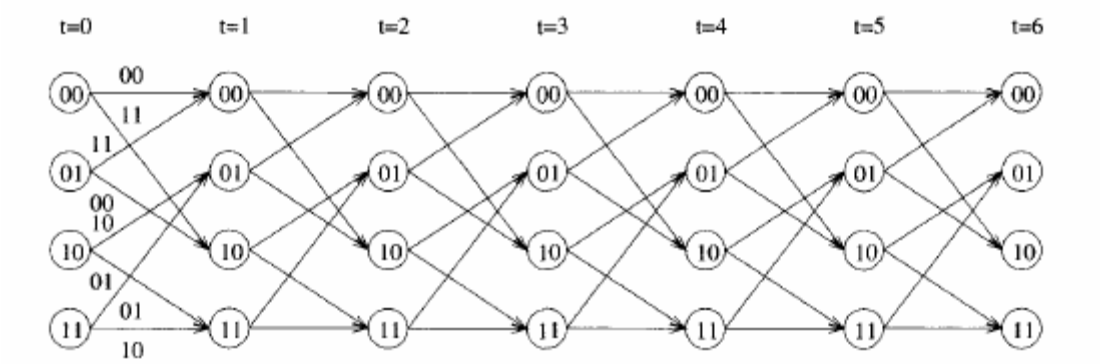

**Σχήμα 2.8** Διάγραμμα κατάστασης ενός συνελικτικού κωδικοποιητή μνήμης 2 και ρυθμό κωδικοποίησης ½

Υπάρχουν n παλμοί στην περίπτωση συνελικτικού κωδικοποιητή με 1/n ρυθμό κωδικοποίησης, ένας παλμός για κάθε έξοδο $\overline{v}^{(j)}, j = 1, ..., n-1$. Καθώς οι παλμοί διέρχονται τα στοιχεία μνήμης του κωδικοποιητή, λαμβάνουν τους κλάδους που συνδέουν τα στοιχεία μνήμης με τις εξόδους. Συνεπώς αυτή η ερμηνεία αντιστοιχεί στα **FIR (finite-impulse-response) συστήματα** μη συστηματικής κωδικοποίησης. Υποθέτουμε το σύνολο $\{\overline{g_0}, ..., \overline{g_{n-1}}\}$ ως το σύνολο των παλμών που αποτελούν το φυσικό μέσο σύνδεσης για τον κωδικοποιητή οι οποίοι ονομάζονται και αλληλουχίες γεννήτριας

ή γεννήτορες. Πρέπει τέλος να σημειώσουμε πως η αλληλουχία γεννήτριας είναι μηδενική μετά από μήκος K bits και ακολούθως εξετάζουμε διανύσματα αυτού του μήκους.

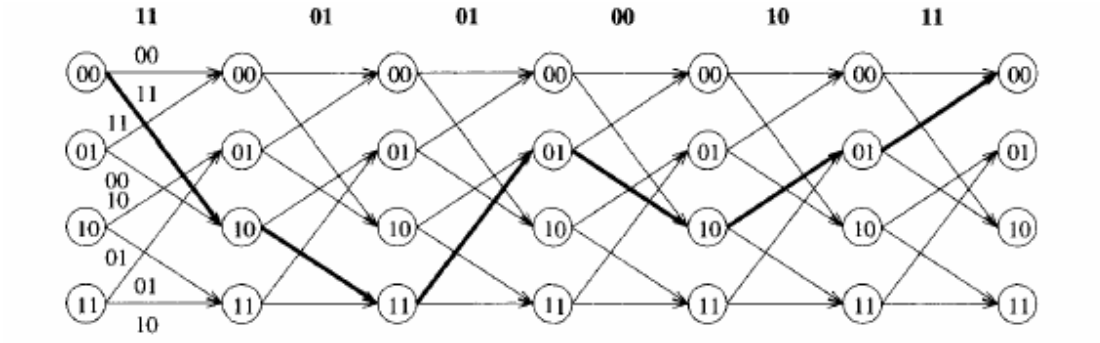
Λόγω της δυναμικής δομής του συνελικτικού κωδικοποιητή, το διάγραμμα κατάστασης μπορεί να παρασταθεί κατά την εξέλιξη του χρόνου με ένα trellis διάγραμμα. Αξίζει να σημειωθεί ότι στην περίπτωση των FIR συστημάτων, το μήνυμα της πληροφορίας εισόδου δεν εμφανίζεται στο διάγραμμα αλλά υπολογίζεται έμμεσα από την συνάρτηση μεταφοράς και καταλήγει στον σχεδιασμό των καταστάσεων ως εξής:

$$(s_0 s_1 \ldots s_{m-1}) \rightarrow (u s_0 \ldots s_{m-2})$$




**Σχήμα 2.9** trellis αναπαράσταση ενός συνελικτικού κωδικοποιητή μνήμης 2 και ρυθμό κωδικοποίησης ½

**Παράδειγμα 2.3**: Υποθέτουμε συνελικτικό κωδικοποιητή με τα χαρακτηριστικά του **Πίνακας 2.2** και αλληλουχία εισόδου $\bar{u} = (1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0)$. Η αλληλουχία εξόδου μπορεί να παραχθεί είτε από το δοσμένο πίνακα, είτε από το trellis μονοπάτι όπως σημειώνεται στο παρακάτω σχήμα:




**Σχήμα 2.10** μονοπάτι στην trellis αναπαράσταση ενός συνελικτικό κωδικοποιητή μνήμης 2 και ρυθμού κωδικοποίησης ½

Ένας συνελικτικός κωδικοποιητής είναι ένα γραμμικό, χρονικά αμετάβλητο σύστημα, με παλμούς να δίνονται από τις ακολουθίες $(\overline{g}_0(D), \overline{g}_1(D), ..., \overline{g}_n(D))$, όπου

$$\overline{g}_j(D) = \overline{g}_j[0] + \overline{g}_j[1]D + \overline{g}_j[2]D^2 + ... + \overline{g}_j[m]D^m \quad 0 \le j < n$$

Οι αλληλουχίες εξόδου $\overline{v^{(j)}}(D)$, $0 \le j < n$ ισοδυναμούν με διακριτή συνέλιξη μεταξύ της αλληλουχίας εισόδου $\overline{u}(D)$ με τις αλληλουχίες γεννήτριας $\overline{g}_0(D), \overline{g}_1(D), ..., \overline{g}_n(D)$. Από το γεγονός αυτό προκύπτει και το όνομα των κωδίκων αυτών ως συνελικτικοί-συνελικτικοί κώδικες.
Καταλήγοντας σε πράξεις μεταξύ πινάκων προκύπτει η έξοδος $\overline{v}$:

$$\overline{v} = \overline{u} * G$$

Όπου ο πίνακας γεννήτριας G ενός συνελικτικού κώδικα δίνεται:

$$G = \begin{pmatrix} g_0[0]g_1[0] & \cdots & g_0[m]g_1[m] & & \\ & g_0[0]g_1[0] & \cdots & g_0[m]g_1[m] & \\ & & g_0[0]g_1[0] & \cdots & g_0[m]g_1[m] \\ & & \ddots & & \ddots \end{pmatrix},$$

Αναλύοντας τα **FIR (finite-impulse-response) συστήματα** μη συστηματικής συνελικτικής κωδικοποίησης, μπορούμε να αναφέρουμε πως η αναδρομική συστηματική συνελικτική κωδικοποίηση αποτελεί **IIR (infinite-impulse-response) σύστημα.** Οι διαφορές των δύο συστημάτων κωδικοποίησης βασίζονται στις παρακάτω εξισώσεις του πολυωνύμου γεννήτριας και του διανύσματος εισόδου:

Αν $G(D) = (\overline{g}_o(D) \quad \overline{g}_1(D) \quad ... \quad \overline{g}_{n-1}(D))$ (FIR συστήματα)

$$G'(D) = (I \quad \frac{\overline{g}_1(D)}{\overline{g}_0(D)} \quad ... \quad \frac{\overline{g}_{n-1}(D)}{\overline{g}_0(D)}) \qquad \text{(IIR συστήματα)}$$

$$\overline{u}'(D) = \overline{g}(D) * \overline{u}(D) \qquad \text{(IIR συστήματα)}$$

Η **ελεύθερη απόσταση** $d_f$ ενός συνελικτικού κώδικα είναι η μικρότερη απόσταση μεταξύ δύο οποιονδήποτε ξεχωριστών ακολουθιών του κώδικα. Το μήκος των ακολουθιών πρέπει να είναι αρκετά μεγάλο, πολύ μεγαλύτερο από το περιορισμένο μήκος του κώδικα. Υπάρχει μια **σύνδεση μεταξύ συνελικτικών κωδίκων και block**

**κωδίκων**. Σε έναν κωδικοποιητή συνελικτικό είναι σύνηθες οι ακολουθίες της πληροφορίας εισόδου να διαιρεθεί σε block πεπερασμένου μήκους. Γενικά, μια σταθερή ακολουθία μήκους m προσαρτάται στο τέλος κάθε ακολουθίας πληροφοριών. Αυτή η ακολουθία είναι συνήθως μια μοναδική λέξη που εξυπηρετεί το συγχρονισμό του δέκτη και ενισχύει τον συνελικτικό κωδικοποιητή να επιστρέφει σε μία γνωστή κατάσταση.

Ωστόσο, για γραμμικούς μπλοκ κώδικες που λαμβάνονται από την περάτωση συνελικτικών κωδίκων για μεγάλου μήκους ακολουθίες, η μέγιστη πιθανότητα αποκωδικοποίησης είναι περίπλοκη και αναποτελεσματική διαδικασία να εφαρμοστεί. Μια αποτελεσματική λύση στο πρόβλημα της αποκωδικοποίησης είναι ένας δυναμικός προγραμματιστικός αλγόριθμο γνωστός ως **Viterbi αλγόριθμος**, ή ως αποκωδικοποιητής Viterbi (VD). Ο VD είναι ο αποκωδικοποιητής μέγιστης πιθανοφάνειας με την παρακάτω έννοια.. Ο VD διαπιστώσει την πλησιέστερη κωδικοποιημένη ακολουθία μέσω της ληφθείσας ακολουθίας επεξεργαζόμενος bit-by-bit τους κλάδους της βασικής δομής trellis. Ο αλγόριθμος αυτός δεν κρατάει πιθανότητες για τις πιθανές λέξεις που μπορεί να έχουν κωδικοποιηθεί, αλλά ανιχνεύει τις καταστάσεις από το αντίστοιχο διάγραμμα [10].

Ένας μέγιστης πιθανότητας αποκωδικοποιητής MLD (maximum-likelihood decoder), επιλέγει την κωδικοποιημένη ακολουθία $\overline{u'}$ που μεγιστοποιεί την παρακάτω εξίσωση:

$$P(\overline{r} / \overline{u}) = \prod_{i=0}^{n-1} P(r_i / u_i)$$

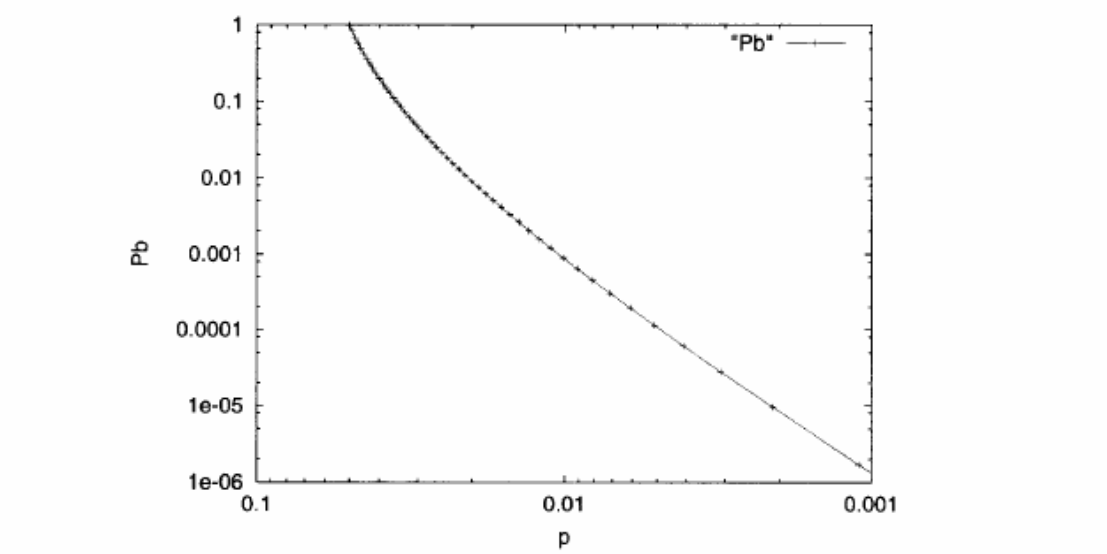
Για ένα BSC κανάλι, αυτό είναι ισοδύναμο με την ελαχιστοποίηση της απόστασης Hamming, όπου δίνεται από τον παρακάτω τύπο:

$$d_H(\overline{r} / \overline{u}) = \sum_{i=0}^{n-1} d_H(r_i / u_i)$$

Όμοια για ένα κανάλι προσθετικού θορύβου AWGN, αυτό είναι ισοδύναμο με την ελαχιστοποίηση της Euclidean απόστασης:

$$d_E(\overline{r} / \overline{u}) = \sum_{i=0}^{n-1} (\overline{r} - m(\overline{u}))^2$$

Στο παρακάτω σχήμα περιγράφεται η μέγιστη πιθανότητα λάθους bit ενός συνελικτικού κωδικοποιητή, ελεύθερης απόστασης 5, μνήμης 2 και ρυθμού κωδικοποίησης ½.




**Σχήμα 2.11** Μέγιστη πιθανότητα αποκωδικοποίησης ενός συνελικτικού κωδικοποιητή, $d_f = 5$ μνήμης 2 και ρυθμού κωδικοποίησης ½

### 2.4.3. Αλγόριθμος Viterbi

Αν υποθέσουμε ότι $s_i^{(k)}$ κατάσταση διαγράμματος trellis στο στάδιο i, σε κάθε κατάσταση εκχωρείται μία μέτρηση $M(s_i^{(k)})$ και ένα μονοπάτι $\overline{y}^{(k)}$, η βασική αρχή στην εφαρμογή του Viterbi αλγορίθμου είναι: Στο χρονικό διάστημα i, τα πιο πιθανά μονοπάτια $\overline{y}^{(k)}$ (αυτά που είναι πλησιέστερα στη ληφθείσα ακολουθία), τελικά θα συμπήσουν σε χρόνο $i-l$. Στη θεωρία του Viterbi υποδεικνύεται ότι $l > 2m$ (m η μνήμη του συνελικτικού κώδικα). Επίσης προϋποθέτει ότι ο αριθμός των bits εξόδου ανά κατάσταση L (γνωστό ως βάθος αποκωδικοποίησης) , να ικανοποιεί την σχέση $L > l$. Παρακάτω αναλύονται τα βασικά βήματα αποκωδικοποίησης:

- Βήμα 0:
  $i$=0, αρχικό στάδιο (χρονική στιγμή)
  $M(s_i^{(k)})$=0 , $\overline{y}^{(k)}$=( ), όπου το μονοπάτι αρχικοποιείται με μια άδεια λίστα.

- Βήμα 1:
  Στο στάδιο $i$ υπολογίζουμε τις μετρήσεις όπως υποδεικνύουν οι τύποι:

$BM_i^{(b)} = \left| d_H(\bar{r}[i], \bar{v}[i]), \right.$ όπου

$b \triangleq \sum_{\ell=0}^{n-1} v_\ell[i] 2^{n-1-\ell}$

- Βήμα 2:
  $S_i^{(k)}$ , $k = 0,1,...,2^{m-1}$ καταστάσεις την χρονική στιγμή $i$
  Για τις καταστάσεις που έχουμε περάσει $S_{i-1}^{(k_1)}$ και $S_{i-1}^{(k_2)}$ ,
  σύγκρινε ζευγάρια κλάδων $M(S_{i-1}^{(k_1)}) + BM_i^{(b_1)}$ και
  $M(S_{i-1}^{(k_2)}) + BM_i^{(b_2)}$ , $b_j = \sum_{\ell=0}^{n-1} v_\ell[i] 2^{n-1-\ell}, i = 1,2$
  Επέλεξε τον κλάδο που ικανοποιεί την σχέση:

$$M(S_i^{(k)}) = \min\{M(S_{i-1}^{(k_1)}) + BM_i^{(b_1)}, M(S_{i-1}^{(k_2)}) + BM_i^{(b_2)}\}.$$

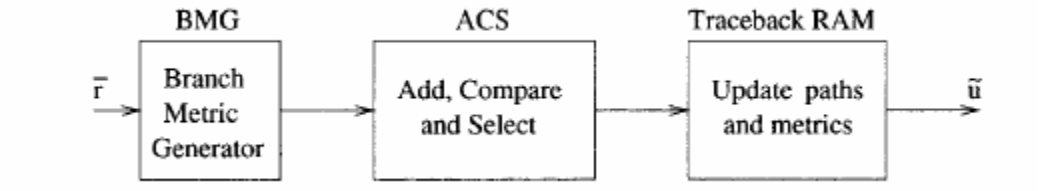
- Βήμα 3:
  Για κάθε κατάσταση $S_i^{(k)}$ , $k = 0,1,...,2^{m-1}$
  Αντικατέστησε το μονοπάτι της τελευταίας κατάστασης με την έξοδο του κλάδου $\bar{v}_{k_j}$ $j \in \{1,2\}$ που τηρεί την προηγούμενη σχέση. Συνεπώς ανανεώνουμε τη λίστα του μονοπατιού σε:

  $\bar{y}_i^{(k)} = (\bar{y}_{i-1}^{(k_j)}, \bar{v}_{k_j})$

- Βήμα 4:

  Εάν $i > L$, η έξοδος υπολογίζεται ως $\bar{y}_{i-L}^{(k')}$, όπου $k'$ δείκτης της κατάστασης $S^{(k')}$ με τη μικρότερη μέτρηση όπως αναφέραμε και παραπάνω.

  Εάν $i \le L$ : $i = i + 1$ και οδηγούμαστε στο Βήμα 1.

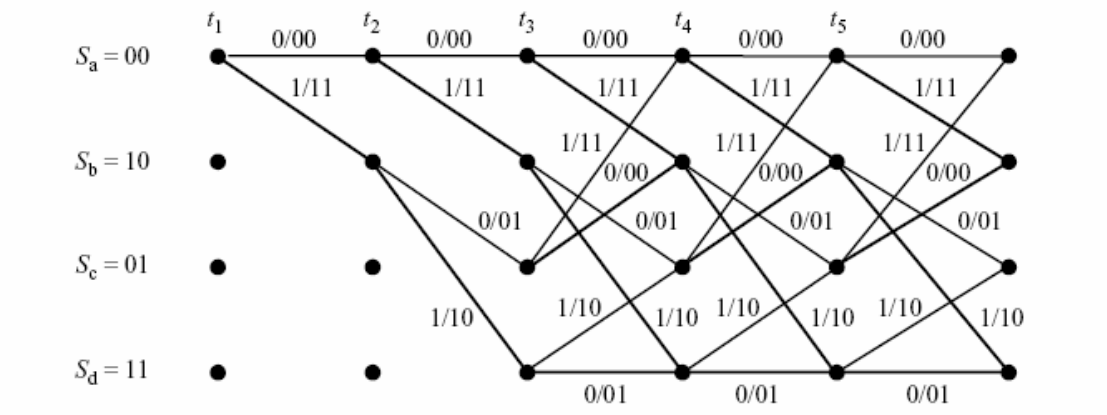



**Σχήμα 2.12** Διάγραμμα αποκωδικοποιητή Viterbi

Θα πρέπει να τονισθεί ότι αυτός δεν είναι ο μόνος τρόπος για την εφαρμογή του αλγορίθμου Viterbi [10]. Η παραπάνω διαδικασία μπορεί να θεωρηθεί ως ένας κλασικός αλγόριθμος. Υπάρχουν εναλλακτικές υλοποιήσεις που, ανάλογα με τη συγκεκριμένη δομή του υποκείμενου συνελικτικού κωδικοποιητή, μπορεί να προσφέρει διαφορετικό πλεονέκτημα. Επιπλέον, κατά το τελευταίο βήμα του αλγορίθμου, αποκωδικοποίηση συμβόλων μπορεί να εφαρμοστεί, ως άμεση αποκωδικοποίηση των bits της πληροφορίας. Αυτό συνήθως χρησιμοποιείται σε εφαρμογές λογισμικού της Viterbi. Όσο για τις hardware υλοποιήσεις, προτιμάται η μέθοδος που βασίζεται σε traceback memory όπου υπολογίζεται έμμεσα η αρχική ακολουθία της πληροφορίας βάση των καταστάσεων μετάβασης.

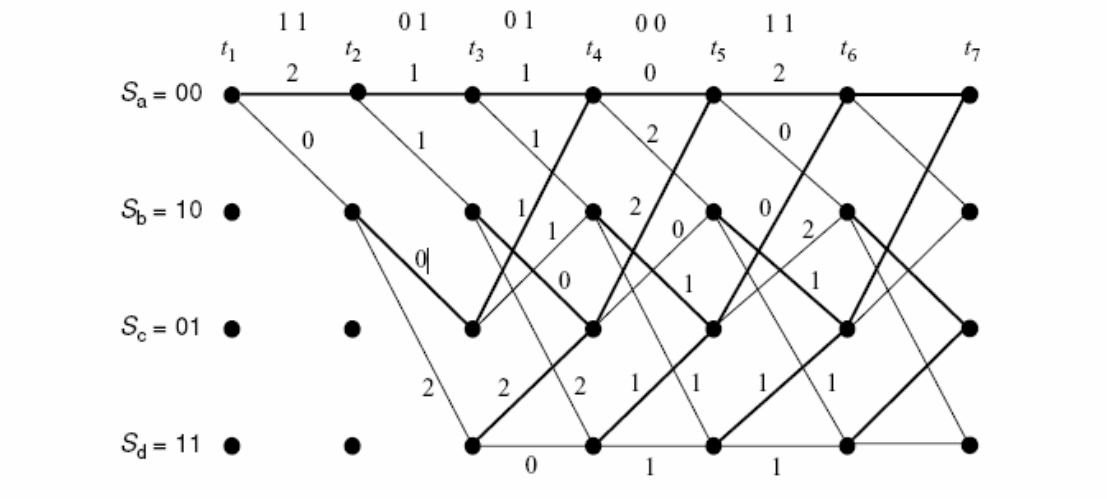
**Παράδειγμα 2.4**: Εφαρμόζουμε τον αλγορίθμου Viterbi στο συνελικτικού κώδικα του παρακάτω σχήματος, σύμφωνα με το παράδειγμα της αναφοράς [2].

| | | | | | |
|---|---|---|---|---|---|
| Ακολουθία Μηνύματος | 1 | 0 | 1 | 0 | 1 |
| Κωδικοποιημένη Ακολουθία | 11 | 01 | 11 | 00 | 11 |
| Ληφθείσα Ακολουθία | 11 | 01 | 01 | 00 | 11 |



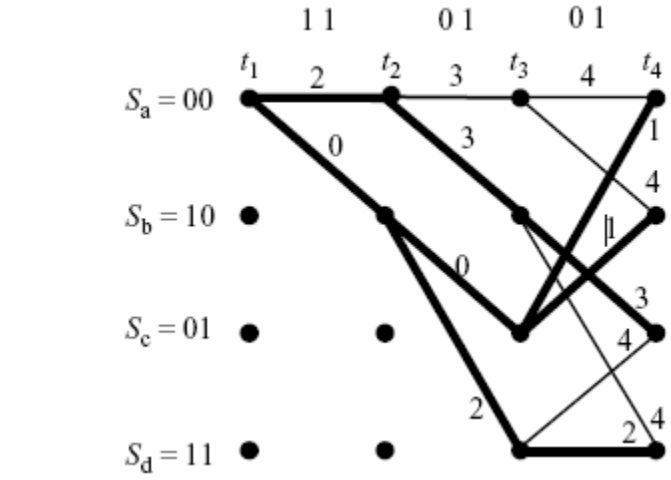

**Σχήμα 2.13** Διάγραμμα Trellis αποκωδικοποιητή Viterbi

Το πρώτο βήμα για την εφαρμογή αυτού του αλγορίθμου είναι να καθοριστεί η απόσταση Hamming μεταξύ της ληφθείσας ακολουθίας και των τιμών της εξόδου για τις διάφορες καταστάσεις και τις χρονικές θυρίδες.




**Σχήμα 2.14** Υπολογισμός Hamming απόστασης σε εφαρμογή αλγορίθμου Viterbi

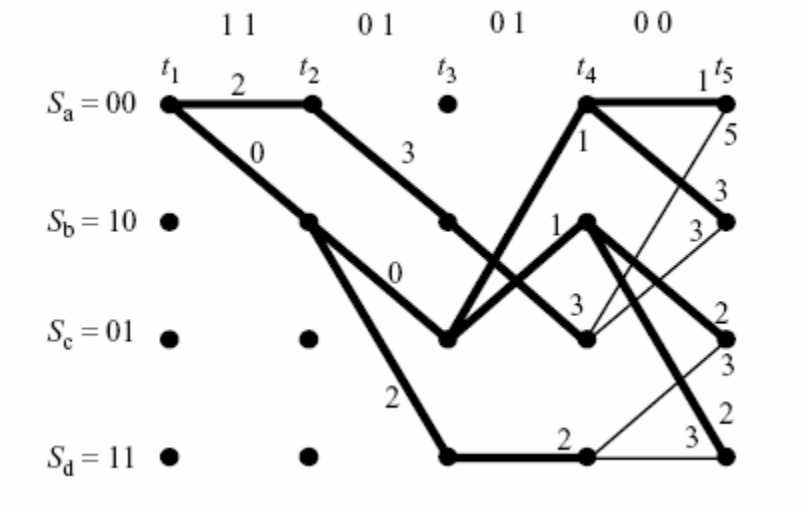
Η ουσία του αλγορίθμου Viterbi είναι ότι όταν δύο ή περισσότερες διαδρομές καταλήξουν σε μια δεδομένη χρονική στιγμή και κατάσταση, μόνο μία από αυτές θα έχει την ελάχιστη συνολική απόσταση, και θα πρέπει να επιλεχθεί μεταξύ των άλλων ως επικρατέστερη. Αποφάσεις με αυτό το κριτήριο αρχίζουν να εκτελούνται από τη στιγμή που δύο ή περισσότερες διαδρομές φτάνουν στην ίδια κατάσταση για κάποια δεδομένη χρονική στιγμή. Για το παράδειγμά μας η αρχή αυτή αρχίζει να εφαρμόζεται για τη χρονική στιγμή $t_4$. Όπως παρατηρούμε στο παρακάτω σχήμα, για κάθε κατάσταση μιας δεδομένης χρονικής στιγμής έχουμε μία διαδρομή επικρατέστερη και τονίζεται με σκούρα μαύρη γραμμή, ενώ οι άλλες εναλλακτικές διαδρομές σημειώνονται με αχνό μαύρο.




**Σχήμα 2.15** Επικρατεί μονοπάτια σύμφωνα με τον αλγόριθμο Viterbi

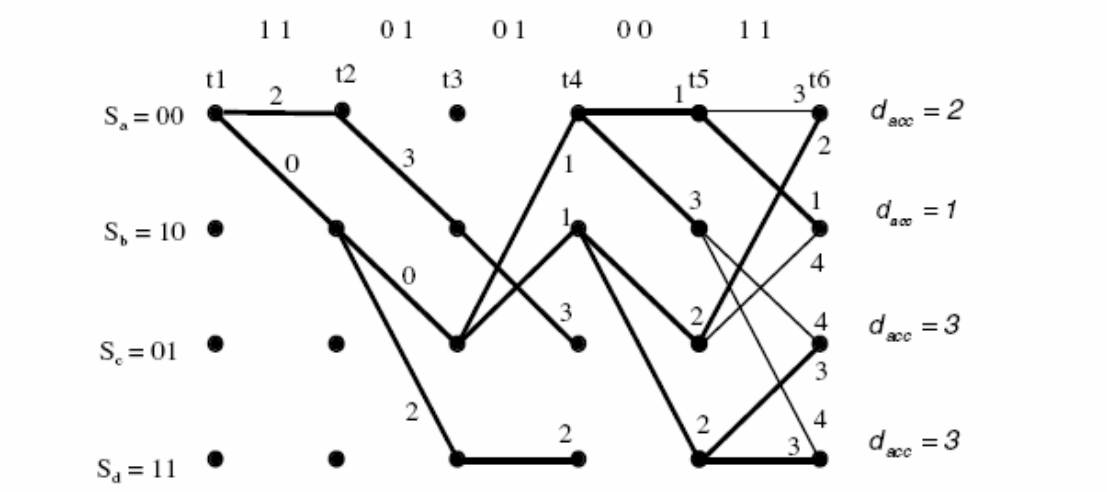
Ωστόσο, είναι αντιληπτό ότι στο χρονικό διάστημα $t_5$, στην κατάσταση $S_b$, καταλήγουν δύο μονοπάτια με την ίδια συνολική Hamming απόσταση, όπως απεικονίζεται και στο παρακάτω σχήμα. Στην περίπτωση αυτή, η απόφαση για την

επικρατέστερη διαδρομή λαμβάνεται με την τυχαία επιλογή μιας από τις δύο. Κατά μέσο όρο, οι τυχαίες επιλογές δεν εμποδίζουν την αποτελεσματική λειτουργία του αλγορίθμου Viterbi: Αν η ικανότητα διόρθωσης του κώδικα καλύπτει το βάρος της λανθασμένης ακολουθίας, τότε η αποκωδικοποιημένη ακολουθία, δεν θα διέρχεται από τον εν λόγο κόμβο. Αντίθετα εάν η ικανότητα διόρθωσης του κώδικα έχει ξεπεραστεί, τότε ο αποκωδικοποιητής αποτυγχάνει ούτως ή άλλως, και συνήθως έχει ως έξοδο μεγάλες ακολουθίες σφαλμάτων.



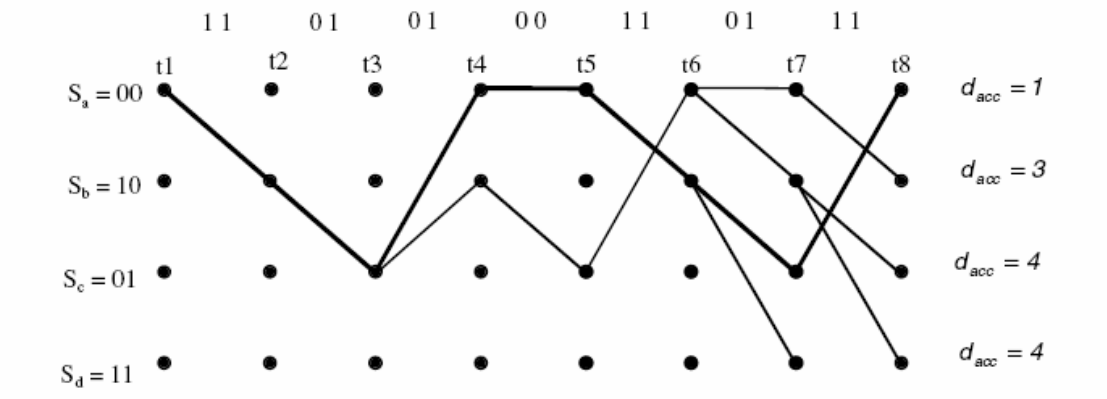

**Σχήμα 2.16** Επικρατή μονοπάτια σύμφωνα με τον αλγόριθμο Viterbi

Όπως είναι φανερό, αν ανατρέξουμε στα δεδομένα του παραδείγματος, παρατηρούμε πως η ακολουθία εξόδου του αλγορίθμου Viterbi (όπως απεικονίζεται στο παρακάτω σχήμα) συμπίπτει με το απεσταλμένο κωδικοποιημένο μήνυμα και ακολούθως πως η σειρά των bits εισόδου είναι η ισχύουσα αρχική. Και σε αυτό το σημείο μπορεί επίσης να επισημανθεί ότι ο Viterbi αποκωδικοποιητής είναι ικανός στην διόρθωση του σφάλματος με βάση της γνωστής ληφθείσας ακολουθίας. Συνεπώς ο Viterbi αποκωδικοποιητής πραγματοποιεί τη διόρθωση των σφαλμάτων χωρίς την ανάγκη του standard-array αποκωδικοποίησης, ή τη χρήση αλγεβρικών εξισώσεων, όπως στην κλασσική περίπτωση υπολογισμού συνδρόμου αποκωδικοποίησης των block κωδίκων. Αυτό το γεγονός καθιστά τους συνελικτικούς κωδικούς ιδιαίτερα αποτελεσματικούς στην εφαρμογή FEC συστημάτων, ή γενικότερα, για τις διαδικασίες κωδικοποίησης όπου η διόρθωση λαθών καθίσταται προτιμότερη της ανίχνευσης λαθών.




**Σχήμα 2.17** Επικρατέστερη ακολουθία εξόδου του αλγορίθμου Viterbi

Στο παραπάνω παράδειγμα, η απόφαση για την επιλογή της διαδρομής με τη μέγιστη πιθανότητα ήταν εύκολη εξαιτίας της ύπαρξης μοναδικής διαδρομής με τη μικρότερη αθροιστική απόσταση Hamming. Ωστόσο υπάρχουν περιπτώσεις όπου τη χρονική στιγμή περάτωσης του αλγορίθμου Viterbi δεν ικανοποιείται η παραπάνω μοναδικότητα και τότε ακολουθούμε την εξής τεχνική: Υποθέτοντας πως $d_H = 1$ όχι μόνο για την δεύτερη αλλά και για την πρώτη κατάσταση της χρονικής στιγμής $t_6$, παρατηρούμε πως οι δύο επικρατέστερες διαδρομές που καταλήγουν σε αυτές τις δύο καταστάσεις συμπίπτουν με τις αντίστοιχες του πρώτου σταδίου ($t_1 \to t_2$). Αυτό υποδεικνύει πως με μεγάλη πιθανότητα, η αποκωδικοποιημένη πληροφορία σε αυτό το στάδιο είναι [1 1] και πως το αρχικό bit αποστολής είναι [1]. Παρόμοια συμπεράσματα εξάγουμε αν συνεχίσουμε με την ίδια λογική και για το επόμενο διάστημα αλλά στην συνέχεια τα στάδια με τις επικρατέστερες διαδρομές δεν συμπίπτουν όπως παρατηρούμε και στο ακόλουθο σχήμα όπου έχουμε ως χρόνο περάτωσης την χρονική στιγμή $t_8$. Αν η ακολουθία μηνύματος, η κωδικοποιημένη ακολουθία και η ληφθείσα ακολουθία είναι μεγάλου μήκους, τότε μπορεί να αποδειχθεί ότι τα επικρατή μονοπάτια συμπίπτουν με υψηλή πιθανότητα τη στιγμή $t_i$, και ότι η απόφαση αποκωδικοποίησης κατά τη χρονική στιγμή $t_i$ θα είναι σωστή αν τα επικρατή μονοπάτια επεκταθούν στην τη χρονική στιγμή $t_{i+J}$ , όπου J ορίζεται ως βάθος αποκωδικοποίησης. Αποδεικνύεται ότι η συνελικτική αποκωδικοποίηση με την βέλτιστη ικανότητα διόρθωσης λάθους επιτυγχάνεται όταν J είναι περίπου ίσο με πέντε φορές το περιορισμένο μήκος του κώδικα ($J \approx 5(K+1)$). Αυτό συνεπάγεται ότι, αφενός, οι συνελικτικοί κώδικες είναι πιο ισχυροί για μεγαλύτερου μήκους μηνύματα, και, αφετέρου, είναι αναγκαίο να προστεθούν K μηδενικά στο τέλος της ακολουθίας μηνύματος προκειμένου να επιτευχθεί η βέλτιστη δυνατότητα διόρθωσης λαθών του κώδικα.




**Σχήμα 2.18** Επικρατέστερη ακολουθία εξόδου έπειτα από επέκταση των σταδίων εφαρμογής του αλγορίθμου Viterbi.

## *2.5. Συναλυσώμενη Κωδικοποίηση*

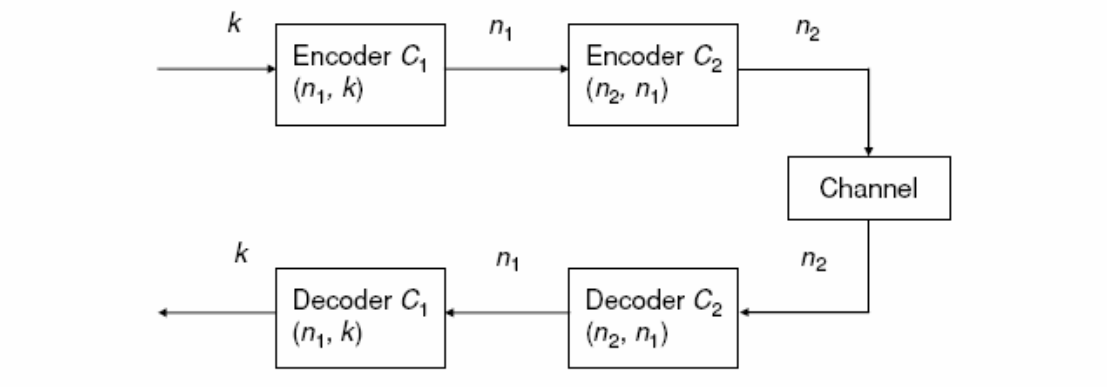
Εξετάζοντας τους συνελικτικούς κώδικες όπως είδαμε στην προηγούμενη ενότητα, εάν η ικανότητα διόρθωσης λαθών του κώδικα ξεπεραστεί από το βάρος της ακολουθίας λάθους που θα προστεθεί από το κανάλι, τότε είναι πιθανόν η αποκωδικοποιημένη πληροφορία να περιέχει τμήματα μεγάλου μήκους λαθών. Για το λόγο αυτό, οι συνελικτικοί κώδικες χρησιμοποιούνται ευρέως ως εσωτερικοί κώδικες της σειράς συνενωμένων συστήματα κωδικοποίησης, όπου ως εξωτερικοί χρησιμοποιούνται συνήθως οι Reed-Solomon (RS) που διαθέτουν υψηλή ικανότητα διόρθωσης μεγάλου μήκους λαθών. Με αυτόν τον τρόπο, με τη βοήθεια του εξωτερικού κώδικα, οποιαδήποτε ακολουθία λαθών κατά αποκωδικοποίηση συνελικτικών κωδίκων μπορεί να εξαλειφθεί. Αυτός ο συνδυασμός συνελικτικού (εσωτερικό) και κώδικες RS (εξωτερικό) των κωδίκων έχει βρεθεί σε πολλά πρακτικές εφαρμογές, συμπεριλαμβανομένης της ψηφιακής δορυφορικής μετάδοσης καθώς και διαστημικές επικοινωνίες [2].

### 2.5.1. Concatenation Μέθοδοι

Οι Concatenation κώδικες [2] είναι μια πολύ χρήσιμη τεχνική που οδηγεί στην κατασκευή πολύ αποτελεσματικών κωδίκων με τη χρήση δύο ή περισσότερων συνιστωσών κωδίκων σχετικά μικρού μεγέθους και πολυπλοκότητας. Έτσι, ένας ισχυρός κώδικας με υψηλές BER (Bit Error Performance) επιδόσεις αλλά ανέφικτης πολυπλοκότητας, μπορεί να κατασκευαστεί σε μια ισοδύναμη μορφή από την συνένωση συνδυασμών δύο ή περισσότερων κωδίκων ώστε να παρέχουν τις ίδιες επιδόσεις με μικρότερο κόστος από την άποψη της πολυπλοκότητας. Η μειωμένη πολυπλοκότητα είναι ιδιαίτερα σημαντική για την αποκωδικοποίηση των κωδίκων αυτών, οι οποίες μπορούν να επωφεληθούν από τη συνδυασμένη δομή ενός Concatenation κώδικα. Η αποκωδικοποίηση γίνεται με το συνδυασμό δύο ή

περισσοτέρων σχετικά χαμηλής πολυπλοκότητας αποκωδικοποιητές, έτσι που αποσυντίθενται αποτελεσματικά το πρόβλημα της αποκωδικοποίησης ενός μεγάλου κώδικα. Εάν αυτοί οι αποκωδικοποιητές μοιραστούν κατάλληλα την πληροφορία προς αποκωδικοποίηση με τη χρήση επαναληπτικής τεχνικής, για παράδειγμα, τότε δεν θα υπάρχει απώλεια της απόδοσης.

Υπάρχουν ουσιαστικά δύο τρόποι υλοποίησης concatenating κωδίκων: παραδοσιακά, με τη χρήση της λεγόμενης σειριακής (serial) concatenation, και πιο πρόσφατα, συνενώνοντας παράλληλα δομές της πρώτης turbo κατηγορίας συστημάτων κωδικοποίησης. Η επαναληπτική διαδικασία αποκωδικοποίησης επιτρέπεται στην χρήση και των δύο αυτών concatenation τεχνικών .




**Σχήμα 2.19** Σειριακή συνέλιξη κωδίκων

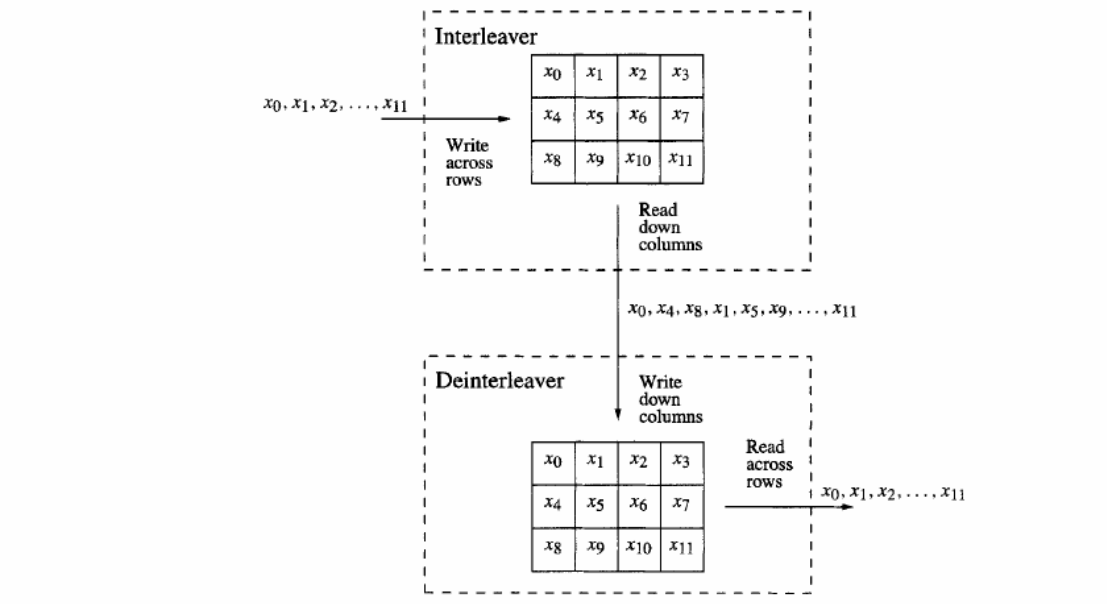
## 2.5.2. Block Interleaving

Ένας interleaver παίρνει μια ακολουθία από σύμβολα και τα συνδυάζει μεταξύ τους. Στο δέκτη, το ακολουθία συνδυάζεται έτσι ώστε να επιστρέψει στην αρχική της σειρά από το σύστημα του deinterleaver. Οι Interleavers είναι αποτελεσματικοί στην αντιμετώπιση μεγάλων εξαπλώσεων λαθών, διότι καθώς μεταθέτει τα σύμβολα στο δέκτη, σφάλματα που εμφανίζονται σε κοντινή απόσταση μπορούν να κατατμηθούν και να εξαπλωθούν σε μεγάλο εύρος, δημιουργώντας έτσι ένα αποτελεσματικό τυχαίο κανάλι [3].

Ένας κοινός τρόπος συνδυασμού και μετατόπισης συμβόλων είναι οι μπλοκ interleavers. Αποτελούνται από πίνακα $N * M$ διαστάσεων που μπορεί να διαβαστεί και να γραφτεί με διάφορους συνδυασμούς ακολουθιών. Συνήθως, η εισερχόμενη ακολουθία συμβόλων είναι γραμμένη στον interleaver ακολουθώντας τις σειρές και διαβάζεται ακολουθώντας τις στήλες. Στο παρακάτω σχήμα απεικονίζεται ένας 3 x 4 interleaver. Η ακολουθία των εισροών $x_0, x_1, ..., x_{11}$ δίνεται σε σειρές του πίνακα, όπως φαίνεται, και διαβάζεται όπως η ακολουθία:

$$x_0 \quad x_4 \quad x_8 \quad x_1 \quad x_5 \quad x_9 \quad x_2 \quad x_6 \quad x_{10} \quad x_3 \quad x_7 \quad x_{11}$$

Συχνά, το μήκος M (ενός πίνακα $N * M$ διαστάσεων ) επιλέγεται για να αποτελέσει το μήκος μιας κωδικοποιημένης λέξης.




**Σχήμα 2.20** Ένας 3*4 interleaver - deinterleaver

# 3. *Τεχνικές Κωδικοποίησης Reed – Solomon*

## *3.1. Γενικά*

Γενικά οι κώδικες block Cb(n, k) ορισμένοι στο πεδίο GF(q) αποτελούν υπόχωρο διάστασης κ του χώρου διανυσμάτων Vn n στοιχείων. Ένας κυκλικός κώδικας ορισμένος στο πεδίο GF(q) παράγεται από το πολυώνυμο γεννήτριας βαθμού n - k που οι συντελεστές του είναι στοιχεία του GF(q) .

Γνωστή κατηγορία των κωδίκων αυτών αποτελούν οι δυαδικοί BCH κώδικες που τους χρησιμοποιούμε για γενικεύσεις πάνω στο GF(q) πεδίο. Αν v , t θετικοί ακέραιοι αριθμοί ,υπάρχει κώδικας μήκους $n = q^v - 1$ ορισμένος στο GF(q) ικανός να διορθώσει οποιοδήποτε συνδυασμό μήκους t ,που είναι δομημένος το λιγότερο από 2vt πλεονάζοντα στοιχεία ελέγχου. Εάν α είναι πρωτεύον στοιχείο του πεδίου GF(q) , το πολυώνυμο γεννήτριας BCH κώδικα βάσης- q ικανό να διορθώσει οποιοδήποτε συνδυασμό μήκους μικρότερο ή ίσου του t είναι πολυώνυμο ελαχίστου βαθμού με συντελεστές από το πεδίο GF(q) και ρίζες $a, a^2, ..., a^{2t}$ .Ο μέγιστος βαθμός του πολυωνύμου γεννήτριας είναι ο 2vt ,που είναι ισοδύναμα και ο μέγιστος αριθμός των πλεοναζόντων στοιχείων ελέγχου.

Στην περίπτωση που q = 2 ο ορισμός αντιστοιχεί στους δυαδικούς BCH κώδικες. Μια υποκατηγορία των BCH κωδίκων βάσης- q έχουμε όταν v =1 που ονομάζονται Reed Solomon κώδικες προς τιμή των ερευνητών τους. Στο επεκταμένο πεδίο Galoi $GF\left(p^m\right)$ , όπου $q=p^m$ , γίνεται αναλυτική περιγραφή παρακάτω.

Ένας RS κώδικας $C_{RS}(n, k)$ ,ικανός να διορθώσει οποιοδήποτε συνδυασμό μήκους μικρότερου ή ίσου του t ,ορισμένος στο πεδίο GF(q) έχει ως παραμέτρους :

| | |
|---|---|
| Μήκος κωδικοποιημένης λέξης | $n = q - 1$ |
| Αριθμός των πλεοναζόντων στοιχείων ελέγχου | $n - k = 2t$ |
| Ελάχιστη απόσταση | $dmin = 2t + 1$ |
| Ικανότητα διόρθωσης λαθών | t στοιχεία για κάθε κωδικοποιημένη λέξη |

### 3.1.1. Εφαρμογές του κώδικα Reed – Solomon

Οι Reed – Solomon κώδικες έχουν βρει πολυάριθμες εφαρμογές στα συστήματα τηλεπικοινωνιών και ψηφιακής επεξεργασίας σήματος. Παραδείγματα περιλαμβάνουν ο γνωστός RS(255, 223, 33) κώδικας για διαχείριση επικοινωνιών της NASA ,πιο μικροί

κώδικες RS στο πεδίο $GF(28)$ για CD-ROM,DVD και επίγειες ψηφιακές HDTV εφαρμογές μετάδοσης, ένας επεκταμένος $RS(128,122,7)$ κώδικας στο πεδίο $GF(27)$ για ενσύρματα μόντεμ και πολλές άλλες εφαρμογές.

### 3.1.2. Εισαγωγικά για την υλοποίηση του κώδικα

Οι Reed – Solomon κώδικες ανήκουν στη κατηγορία των γραμμικών ,μη δυαδικών κυκλικών κωδίκων ελέγχου, υποκατηγορία των κυκλικών BCH κωδίκων που αποτελούν γενίκευση πάνω από Galois field GF(q) , όπου q είναι δύναμη ενός αριθμού βάσης $q=p^m$ ,όπου m είναι ένας θετικός ακέραιος.

Οι RS κώδικες μπορούν να οριστούν σαν κωδικοποιημένες λέξεις με συντελεστές ισοδύναμους της τιμής χαρακτηριστικού πολυωνύμου. Εδώ μπορούμε να αναφερθούμε στις ιδιότητες των γενικευμένων δυαδικών κυκλικών BCH κωδίκων που επεκτείνονται στους RS κώδικες.

Αρχικά κάθε κυκλικός κώδικας είναι γραμμικός, ικανότητα που τους κάνει κατάλληλους για εφαρμογές hardware. Για κάθε κωδικοποιημένη λέξη u συνδέεται ένα πολυώνυμο u(x)

$$u=(u_0,u_1,...,u_{n-1}) \rightarrow u(x)=u_0+u_1 \bullet x+...+u_{n-1} \bullet x^{n-1} \qquad (3\text{-}1)$$

Η κυκλικότητα ορίζεται ως την ιδιότητα κάθε κωδικοποιημένης λέξης μετά από οποιαδήποτε κυκλική εναλλαγή των συντελεστών της να παραμένει κωδικοποιημένη λέξη.

$$u=(u_0,u_1,...,u_{n-1}) \in C \leftrightarrow u'=(u_{n-1},u_0,...,u_{n-2}) \in C \qquad (3\text{-}2)$$

Κάθε κυκλικός $(n,k)$ κώδικας, χαρακτηρίζεται από το πολυώνυμο γεννήτριας $g(x)$

$$g(x)=g_0+g_1 \bullet x+...+g_{n-k} \bullet x^{n-k} \qquad (3\text{-}3)$$

παράγοντα του πολυωνύμου $X^n+1$ και βαθμού $n-k$ .Δηλαδή $X^n+1=g(x) \bullet h(x)$
Το μήνυμα που εισάγεται στο σύστημα είναι της μορφής

$$m(x)=m_0+m_1 \bullet x+m_2 \bullet x^2+...+m_{k-1} \bullet x^{k-1} \qquad (3\text{-}4)$$

Συνεπώς η κωδικοποιημένη λέξη που εισέρχεται στο κανάλι προέρχεται από τις παραπάνω σχέσεις και είναι της μορφής:

$$u(x) = g(x) * m(x) \qquad (3\text{-}5)$$

$$u(x) = u_0 + u_1 \bullet x + \ldots + u_{n-1} \bullet x^{n-1} \qquad (3\text{-}6)$$

Γενικά για τους κυκλικούς κώδικες, πρώτα υλοποιείται το πολυώνυμο:

$$X^{n-k} \bullet m(X) = m_0 \bullet X^{n-k} + m_1 \bullet X^{n-k+1} + \cdots + m_{k-1} \bullet X^{n-1}$$
$$u(X) = X^{n-k} \bullet m(X) + p(X)$$
$$= p_0 + p_1 \bullet X + \cdots + p_{n-k-1} X^{n-k-1} + m_0 X^{n-k} + m_1 X^{n-k+1} + \cdots + m_{k-1} X^{n-1}$$
$$u = (p_0, p_1, \ldots, p_{n-k-1}, m_0, m_1, \ldots, m_{k-1})$$
$$X^{n-k} \bullet m(X) = m_0 \bullet X^{n-k} + m_1 \bullet X^{n-k+1} + \cdots + m_{k-1} \bullet X^{n-1}$$

Στη συνέχεια διαιρείται με το πολυώνυμο γεννήτριας $g(x)$:

$$X^{n-k} \bullet m(X) = q(X) \bullet g(X) + p(X)$$

Όπου $p(X)$ είναι πολυώνυμο υπόλοιπο της παραπάνω διαίρεσης βαθμού μικρότερο ή ίσου του $n - k - 1$.
Τροποποιώντας την παραπάνω εξίσωση προκύπτει :

$$X^{n-k} \bullet m(X) + p(X) = q(X) \bullet g(X)$$

Είναι φανερό τώρα πως το πολυώνυμο $X^{n-k} \bullet m(X) + p(X)$ είναι παράγοντας του $g(X)$ και συνεπώς αποτελεί τη κωδικοποιημένη λέξη ή κωδικό πολυώνυμο . Αναλυτικά ο όρος $X^{n-k} \bullet m(X)$ αντιπροσωπεύει το μήνυμα μετατοπισμένο κατά $n-k$ θέσεις δεξιά ενώ το $p(X)$ αποτελεί τα περιττά σύμβολα που μεταφέρονται και εμφανίζονται στις πρώτες κ δυνάμεις της κωδικοποιημένης λέξης

$$u(X) = X^{n-k} \bullet m(X) + p(X) = X^2 + X^5 + X^3$$
$$X^5 + X^3 \quad | X^3 + X + 1$$
$$X^5 + X^3 + X^2 \quad X^2$$
$$X^2$$
$$p(X) = X^2$$
$$u(X) = X^{n-k} \bullet m(X) + p(X)$$
$$= p_0 + p_1 \bullet X + \cdots + p_{n-k-1} X^{n-k-1} + m_0 X^{n-k} + m_1 X^{n-k+1} + \cdots + m_{k-1} X^{n-1} \qquad (3\text{-}7)$$

που όταν εκφράζεται σαν διάνυσμα είναι της μορφής

$u= (p_0, p_1, \ldots, p_{n-k-1}, m_0, m_1, \ldots, m_{k-1})$

**Παράδειγμα 3.1**: Για ένα γραμμικό κυκλικό κώδικα Ccyc(7, 4) που παράγεται από το πολυώνυμο γεννήτριας $g(X) = 1 + X + X^3$ ,καθορίστε την κωδικοποιημένη λέξη αν το αρχικό μήνυμα παριστάνεται με το διάνυσμα $m = (1010)$.

$m(X) = 1 + X^2$ , $n - k = 7 - 4 = 3$

υπολογίζοντας το γινόμενο $X^3 \bullet m(X) = X^3 + X^5$ και διαιρώντας το με το πολυώνυμο γεννήτριας έχουμε

$X^5 + X^3 \quad | X^3 + X + 1$

$X^5 + X^3 + X^2 \quad X^2$

$X^2$

$p(X)= X^2$

Η αντίστοιχη κωδικοποιημένη λέξη προκύπτει συνεπώς :

$u(X) = X^{n-k} \bullet m(X) + p(X)= X^2 + X^5 + X^3$

### 3.1.3. θεωρία Galois fields

Οι κώδικες RS ανήκουν στην κατηγορία των BCH κωδίκων που οι τιμές των συντελεστών τους προκύπτουν μέσα από το πεδίο $GF(2^m)$ . Το πεδίο αυτό είναι επέκταση του GF(p) πεδίου , όπου p είναι ένας αριθμός βάσης (στην περίπτωση δυαδικών κωδίκων p=2 και οι αντίστοιχοι συντελεστές παίρνουν την τιμή 0 ή 1. Όμως για τους κώδικες RS χρειαζόμαστε το επεκταμένο πεδίο Galoi $GF(2^m)$ και ένα πρωτεύον πολυώνυμο f(x) (βαθμού m) χαρακτηριστικό για κάθε m έτσι ώστε ο μικρότερος θετικός ακέραιος n που το f(x) διαιρεί ακριβώς την παράσταση $X^n +1$ να είναι $n=2^m -1$ .

Με τη βοήθεια του matlab εισάγοντας κάθε φορά τον αριθμό m των bits που αποτελούν ένα σύμβολο, προκύπτει αυτόματα το πρωτεύον πολυώνυμο f(x) και η μετατροπή των συντελεστών από το πεδίο GF στην αντίστοιχη ακολουθία m bits γίνεται αυτόματα. Ένα αρχικό στάδιο για τον υπολογισμό των στοιχείων του πεδίου είναι να εισάγουμε τα στοιχεί (0, 1, a) ως αρχικό σύνολο και μετά από κατάλληλες εξισώσεις να το επεκτείνουμε και να υπολογίσουμε το τελικό σύνολο. Προέκταση του αρχικού συνόλου προκύπτει από πολλαπλασιασμό του τελικού κάθε φορά στοιχείου με το a

$F=\{0,1,a\ ,a^2, a^3,...\}$

Όπου $F$ αποτελεί ένα σύνολο συντελεστών-στοιχείων ενός *Galoi Field* .

Όμως $X^n + 1=0$ , και επειδή μηδενίζοντας την παράσταση αυτή μηδενίζεται και το πολυώνυμο γεννήτριας, όπως έχουμε ήδη αναφέρει, για τις ρίζες του πολυωνύμου γεννήτριας ισχύει:

$$a^{2^m-1}+1=0 \rightarrow a^{2^m-1}=1=a^0$$

$$\rightarrow a^4=1+a$$

$$a^4=1+a$$

$$a^{2^m-1}+1=0$$

$$n < 2^{n-k}$$

$$C_m$$

$$Y = Y = C_m + e + e$$

$$a^{2^m-1}=1=a^0$$

Άρα , $F=\left\{0,1,a,a^2,a^3,...,a^{2^m-2}\right\}$

Χρησιμοποιώντας και την εξίσωση του πρωτεύοντος πολυωνύμου προκύπτουν σχέσεις και ισότητες μεταξύ των δυνάμεων του α. Δημιουργείται ακολούθως ο πίνακας αθροίσματος (modulo-2 άθροισμα ων αντίστοιχων bits κάθε συμβόλου) των στοιχείων του πεδίου GF και ο πίνακας γινομένου.

Μια απλή εφαρμογή για την ευκολότερη κατανόηση αποτελεί το ακόλουθο παράδειγμα που σχηματίζει την αντιστοιχία συμβόλων από το galoi field στους δεκαδικούς αριθμούς.

**Παράδειγμα 3.2**: Υπολογίστε το άθροισμα των αριθμών (1, 0, 1, 1) + (0, 1, 0, 1) .
Με βάση τις ιδιότητες του modulo-2 αθροίσματος θα πρέπει να ισχύει: ( 1, O , 1, 1) + (0, 1,0, 1) = (1, 1, 1,O)
Αρχικά επιλέγουμε ένα πρωτεύον πολυώνυμο βαθμού 4. Το πολυώνυμο αυτό που πρέπει να ικανοποιεί τις προαναφερθείσες ιδιότητες έστω ότι είναι το $f(x) = 1 + x + x^4$ . Όσο για την υλοποίηση του κώδικα το πρωτεύον πολυώνυμο που χρησιμοποιώ κάθε φορά για την εκτέλεσή του είναι πολυώνυμο default που έχει στη μνήμη του το matlab. Στο επεκταμένο πεδίο $GF(2^4)$ ορίζουμε a ως ρίζα του f(x) :

$$a^4+a+1=0$$

$$a^4=1+a$$

Διαδοχικές δυνάμεις του a προκύπτουν ως εξής:

$$a^5 = a \bullet (a^4) = a + a^2$$

$$a^6 = a^2 \bullet (a^4) = a^2 + a^3$$

$$a^7 = a^3 \bullet (a^4) = a^3 \bullet (1+a) = a^3 + 1 + a$$

**Πίνακας 3.1** Παράσταση πολυωνύμων $GF(2^4)$ χρησιμοποιώντας πρωτεύον πολυώνυμο $f(a) = 1+a+a^4$

| Polynomial Representation | Vector Representation | Vector Representation (integer) | Power Representation $\alpha^n$ |
|---|---|---|---|
| 0 | 0 0 0 0 | 0 | – |
| 1 | 1 0 0 0 | 1 | $1 = \alpha^0$ |
| $\alpha$ | 0 1 0 0 | 2 | $\alpha$ |
| $\alpha^2$ | 0 0 1 0 | 4 | $\alpha^2$ |
| $\alpha^3$ | 0 0 0 1 | 8 | $\alpha^3$ |
| $1+\alpha$ | 1 1 0 0 | 3 | $\alpha^4$ |
| $\alpha+\alpha^2$ | 0 1 1 0 | 6 | $\alpha^5$ |
| $\alpha^2+\alpha^3$ | 0 0 1 1 | 12 | $\alpha^6$ |
| $1+\alpha+\alpha^3$ | 1 1 0 1 | 11 | $\alpha^7$ |
| $1+\alpha^2$ | 1 0 1 0 | 5 | $\alpha^8$ |
| $\alpha+\alpha^3$ | 0 1 0 1 | 10 | $\alpha^9$ |
| $1+\alpha+\alpha^2$ | 1 1 1 0 | 7 | $\alpha^{10}$ |
| $\alpha+\alpha^2+\alpha^3$ | 0 1 1 1 | 14 | $\alpha^{11}$ |
| $1+\alpha+\alpha^2+\alpha^3$ | 1 1 1 1 | 15 | $\alpha^{12}$ |
| $1+\alpha^2+\alpha^3$ | 1 0 1 1 | 13 | $\alpha^{13}$ |
| $1+\alpha^3$ | 1 0 0 1 | 9 | $\alpha^{14}$ |

### 3.1.4. Εύρεση πολυωνύμου γεννήτριας

Για ένα Reed Solomon κώδικα με χαρακτηριστικά:

$$(n,k) = (2^m - 1, 2^m - 1 - 2t) \qquad (3\text{-}8)$$

όπου $n-k = 2t$ είναι ο αριθμός των πλεοναζόντων συμβόλων, και $t$ η ικανότητα διόρθωσης του κώδικα, το πολυώνυμο γεννήτριας παίρνει την μορφή :

$$g(x) = g_0 + g_1 x + g_2 x^2 + \ldots + g_{2t-1} x^{2t-1} + x^{2t} \qquad (3\text{-}9)$$

Ο βαθμός του πολυωνύμου γεννήτριας είναι ίσος με τον αριθμό των πλεοναζόντων συμβόλων. Εφόσον ο βαθμός του πολυωνύμου γεννήτριας είναι $2t$ , πρέπει να υπάρχουν ακριβώς $2t$ διαδοχικές δυνάμεις του a που αποτελούν ρίζες του πολυωνύμου. Ορίζουμε

τις ρίζες του g(x) ως $a, a^2, ..., a^{2t}$ (χωρίς να είναι υποχρεωτικό να αρχίσουμε με την πρώτη δύναμη του $a$ ως ρίζα). Υποθέτοντας κώδικα $RS(7,3)$, 2-συμβόλων ικανότητας λάθους, το πολυώνυμο γεννήτριας βαθμού $n-k=4$ ή ισοδύναμα 4 ριζών είναι :

$$g(x) = (x-a)(x-a^2)(x-a^3)(x-a^4) \qquad (3\text{-}10)$$

που μετά από πράξεις γινομένου και πρόσθεσης στο πεδίο $GF(8)$ μεταξύ των δυνάμεων του $a$ προκύπτει

$$g(x) = a^3 + a^1 x + a^0 x^2 + a^3 x^3 + x^4 \qquad (3\text{-}11)$$

Για μεγαλύτερη ευκολία παρατίθενται οι πίνακες αθροίσματος και γινομένου μεταξύ των δυνάμεων του $a$ στο πεδίο $GF(8)$. Ο πίνακας γινομένου προκύπτει εύκολα από τον γνωστό πολλαπλασιασμό δυνάμεων κοινής βάσης, ενώ ο πίνακας πρόσθεσης υπολογίζεται έπειτα από αντιστοιχία δύναμης σε διάνυσμα συμβόλου στο πεδίο $GF(8)$ και $\mathrm{mod}\,ulo-2$ άθροισμα των *bits* των συμβόλων του αθροίσματος.

**Πίνακας 3.2** Πίνακας αθροίσματος στο $GF(8)$ πεδίο ( $f(x) = 1 + x + x^3$ )

| | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ |
|---|---|---|---|---|---|---|---|
| $a^0$ | 0 | $a^3$ | $a^6$ | $a^1$ | $a^5$ | $a^4$ | $a^2$ |
| $a^1$ | $a^3$ | 0 | $a^4$ | $a^0$ | $a^2$ | $a^6$ | $a^5$ |
| $a^2$ | $a^6$ | $a^4$ | 0 | $a^5$ | $a^1$ | $a^3$ | $a^0$ |
| $a^3$ | $a^1$ | $a^0$ | $a^5$ | 0 | $a^6$ | $a^2$ | $a^4$ |
| $a^4$ | $a^5$ | $a^2$ | $a^1$ | $a^6$ | 0 | $a^0$ | $a^3$ |
| $a^5$ | $a^4$ | $a^6$ | $a^3$ | $a^2$ | $a^0$ | 0 | $a^1$ |
| $a^6$ | $a^2$ | $a^5$ | $a^0$ | $a^4$ | $a^3$ | $a^1$ | 0 |

**Πίνακας 3.3** Πίνακας γινομένου στο $GF(8)$ πεδίο ( $f(x) = 1 + x + x^3$ )

| | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ |
|---|---|---|---|---|---|---|---|
| $a^0$ | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ |
| $a^1$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ | $a^0$ |
| $a^2$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ | $a^0$ | $a^1$ |

$$
\begin{array}{cccccccc}
a^3 & a^3 & a^4 & a^5 & a^6 & a^0 & a^1 & a^2 \\
a^4 & a^4 & a^5 & a^6 & a^0 & a^1 & a^2 & a^3 \\
a^5 & a^5 & a^6 & a^0 & a^1 & a^2 & a^3 & a^4 \\
a^6 & a^6 & a^0 & a^1 & a^2 & a^3 & a^4 & a^5
\end{array}
$$

### 3.1.5. Ικανότητα διόρθωσης λαθών του κώδικα

Για την ανίχνευση λάθους χρησιμοποιούμε την παράσταση $s = rH^T$ που ονομάζεται σύνδρομο. Όπου r είναι διάνυσμα n διάστασης με στοιχεία από το πεδίο GF(q) και H είναι πίνακας πλεοναζόντων στοιχείων ελέγχου για τον συγκεκριμένο κώδικα. Η παράσταση αυτή χρησιμοποιείται για την ανίχνευση εάν το r αποτελεί κωδικοποιημένη λέξη ή εάν έχει αλλοιωθεί από την προσθήκη θορύβου κατά την διέλευση του σήματος από το κανάλι.

Από τον ορισμό ισχύει $s = 0$ για ακριβώς εκείνες τις θέσεις που το κανάλι δεν εισάγει λάθη.

$$r = c + e \qquad (3\text{-}12)$$

όπου e είναι το διάνυσμα λάθους και ακολούθως ισχύει :

$$s = rH^T = (c + e)H^T = eH^T \qquad (3\text{-}13)$$

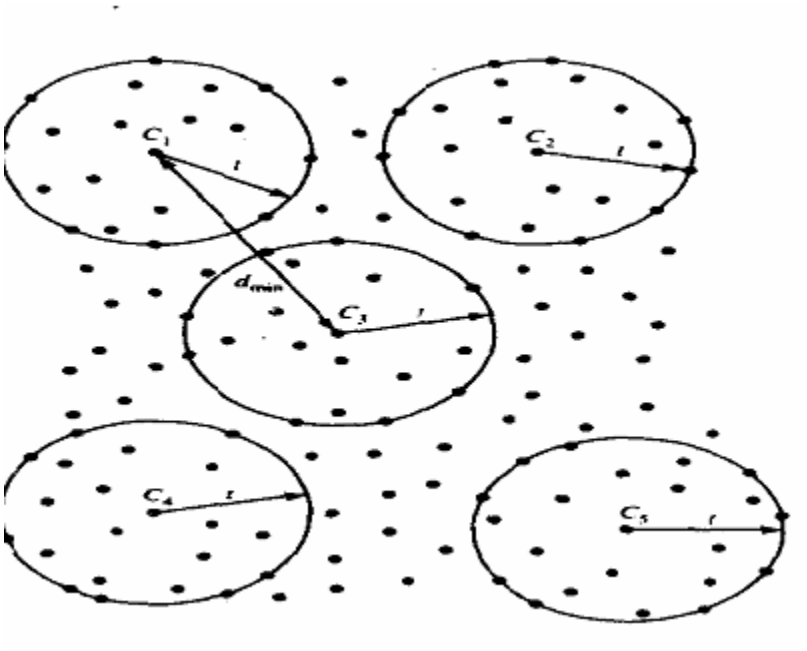
Το διάνυσμα r που λαμβάνεται μπορεί να είναι οποιαδήποτε ακολουθία ανήκει στο GF(q), $s = 0$ εάν r αποτελεί κωδικοποιημένη λέξη, διαφορετικά $s \neq 0$ και η τιμή του χρησιμοποιείται για την ανίχνευση του λάθους.

Είναι τώρα φανερό πως όταν όλες οι τιμές του συνδρόμου είναι μηδενικές ,τότε η λαμβανόμενη κωδικοποιημένη λέξη είναι μια από τις $2^k$ απεσταλμένες κωδικοποιημένες λέξεις. Επειδή ο ελάχιστος διαχωρισμός ανάμεσα σε δυο κωδικοποιημένες λέξεις είναι $d_{min}$ ,είναι δυνατό για ένα λάθος βάρους $d_{min}$ να λάβουμε ως κωδικοποιημένη λέξη μια από τις $2^k$ λέξεις που ανήκουν στον κώδικα και όχι όμως τη σωστή που στάλθηκε. Εάν αυτό συμβεί θα έχουμε ένα μη ανιχνεύσιμο λάθος. Στη γενική όμως περίπτωση εάν ο αριθμός των λαθών είναι μικρότερος τη τιμής $d_{min}$ τότε $s \neq 0$ και τα λάθη ανιχνεύονται.

Συγκεκριμένα η ικανότητα διόρθωσης λάθους ενός κώδικα βασίζεται στην ελάχιστη απόσταση $d_{min}$. Με απεικόνιση των $2^k$ κωδίκων λέξεων σε ένα διανυσματικό χώρο n διαστάσεων σαν τα κέντρα σφαιρών ακτίνας t ,

$$t=\left\lfloor \frac{1}{2}\left(d_{\min}-1\right)\right\rfloor \qquad (3\text{-}14)$$

εξασφαλίζουμε πως για μέγιστο αριθμό λαθών $t$ δεν αλληλοκαλύπτονται οι σφαίρες, δηλαδή οι $2^k$ κωδικοποιημένες λέξεις και έτσι λαμβάνουμε την σωστή λέξη. Τα παραπάνω οδηγούν στο ότι ένας $(n,k)$ κώδικας είναι ικανός να ανιχνεύσει $n-k$ λάθη και να διορθώσει $t$ λάθη.




**Σχήμα 3.1**Αναπαράσταση των κωδίκων λέξεων σαν κέντρα σφαιρών ακτίνας $t$

Όσον αφορά τον $RS(n,k)$ κώδικα που εξετάζουμε , μπορούμε να εξηγήσουμε με μαθηματικούς υπολογισμούς την παράσταση $\mathrm{d}_{\min}=n-k+1$ .

Επειδή $m(\mathrm{x})=m_0+m_1\bullet x+m_2\bullet x^2+\ldots+m_{k-1}\bullet x^{k-1}$ ,οι κωδικοποιημένες λέξεις έχουν το πολύ $k-1$ μηδενικά και συνεπώς ισχύει

$$\mathrm{dmin} \geq \mathrm{n} - (\mathrm{k} - 1) \qquad (3\text{-}15)$$

.

Όμως επειδή για τα $n-k$ πλεονάζοντα στοιχεία που εισάγονται κατά την κωδικοποίηση ισχύει πως ο πίνακας Η έχει $n-k$ γραμμικώς ανεξάρτητες στήλες ,οποιοσδήποτε συνδυασμός $n-k+1$ στηλών πρέπει να είναι γραμμικός εξαρτημένος. Συνεπώς η ελάχιστη απόσταση δεν μπορεί να είναι μεγαλύτερη από $n-k+1$, δηλαδή για όλους τους γραμμικούς κώδικες ισχύει

$\text{dmin} \leq \text{n - (k - 1)}$ (3-16)

Συνεπώς για τον $RS(n,k)$ κώδικα θα ισχύει η ισότητα:

$\text{dmin} = \text{n - (k - 1)}$ (3-17)

Συμπεραίνουμε από τα παραπάνω πως για τον $RS(n,k)$ κώδικα ,όπως και για όλους τους μη δυαδικούς κώδικες, για το ίδιο μήκος $(\text{n},\text{k})$ κώδικα ,η ελάχιστη απόσταση $\text{dmin}$ είναι πολύ μεγαλύτερη συγκριτικά με τους δυαδικούς κώδικες.
Στο συμπέρασμα αυτό οδηγούμαστε και με το ακόλουθο παράδειγμα.

**Παράδειγμα 3.3**: Υποθέτουμε κώδικα $(\text{n},\text{k})=(7,3)$

Στην περίπτωση δυαδικού κώδικα, προκύπτουν $2^3 = 8$ κωδικοποιημένες λέξεις, ενώ το σύνολο των συμβόλων που μεταφέρονται και είναι δυνατό να προκύψουν είναι $2^7 = 128$.
Το ποσοστό των έγκυρων κωδίκων λέξεων εκφράζεται από το κλάσμα

$$\frac{8}{128} = \frac{1}{16} \ .$$

Αντίθετα στην περίπτωση των μη δυαδικών κωδίκων με έστω $m = 3bits$ το μήκος κάθε συμβόλου , το κλάσμα των κωδίκων λέξεων από το σύνολο των πιθανών λέξεων που προκύπτουν στην έξοδο είναι πολύ μικρότερο, συγκεκριμένα ισούται με

$$\frac{2^{km}}{2^{nm}} = \frac{512}{2097152} = \frac{1}{4096} \ .$$

Παράλληλα αυξάνοντας το μήκος $m$ του συμβόλου , μειώνεται το κλάσμα των κωδίκων λέξεων συναρτήσει των πιθανών λέξεων , και ακολούθως αυξάνεται η ελάχιστη απόσταση μεταξύ των έγκυρων κωδίκων λέξεων του κώδικα.

### 3.1.6. Ελάχιστη απόσταση κώδικα ως συντελεστής στην μέγιστη πιθανότητα ανίχνευσης κ διόρθωσης λαθών κατά την διαδικασία της αποκωδικοποίησης.

Τα n bits από τον αποδιαμορφωτή που αντιστοιχούν σε μια ληφθείσα κωδικοποιημένη λέξη εισέρχονται στον κωδικοποιητή, ο οποίος συγκρίνει τη ληφθείσα κωδικοποιημένη λέξη με τις M πιθανές μεταδιδόμενες κωδικοποιημένες λέξεις και αποφασίσει υπέρ της κωδικοποιημένης λέξης που βρίσκεται πλησιέστερα στην απόσταση Hamming (αριθμός των bit θέσεων στις οποίες δύο λέξεις διαφέρουν) για την ληφθείσα κωδικοποιημένη λέξη. Αυτός ο κανόνας ελάχιστης απόστασης αποκωδικοποίησης είναι βέλτιστος ως προς το ότι

οδηγεί σε μια ελάχιστη πιθανότητα λάθους για μια κωδικοποιημένη λέξη για συμμετρικά δυαδικά κανάλια.

Μια εννοιολογικά απλή, έστω και υπολογισμών αναποτελεσματική, μέθοδος αποκωδικοποίησης είναι πρώτα να προσθέσετε (modulo 2) το διάνυσμα της ληφθείσας κωδικοποιημένης λέξης σε όλες τις M δυνατές μεταδιδόμενες κωδικοποιημένες λέξεις Ci για να αποκτήσετε τα διανύσματα λάθους ej. Ως εκ τούτου, ej αντιπροσωπεύει το σφάλμα που πρέπει να έχει πραγματοποιηθεί στο κανάλι, προκειμένου να μετασχηματίσει την κωδικοποιημένη λέξη Cj στην ληφθείσα κωδικοποιημένη λέξη. Τα ορισμένα λάθη στη μετατροπή της Cj στη ληφθείσα κωδικοποιημένη λέξη είναι ακριβώς ίσα σε αριθμό με αυτό των ei. Έτσι, αν θέλουμε απλώς να υπολογίσουμε το βάρος του καθενός από τα M διανύσματα σφάλματος (ej) και να αποφασίσουμε υπέρ της κωδικοποιημένης λέξης που οδηγεί στο διάνυσμα με το μικρότερο βάρος σφάλματος, που έχουμε, είναι στην πραγματικότητα, η υλοποίηση του κανόνα της ελάχιστης απόστασης αποκωδικοποίησης.

Μια πιο αποτελεσματική μέθοδος για πιο αυστηρή ( *hard* ) απόφαση αποκωδικοποίησης κάνει χρήση του πίνακα ελέγχου ισοτιμίας H. Πιο αναλυτικά, υποθέτοντας ότι $C_m$ είναι η μεταδιδόμενη κωδικοποιημένη λέξη και $Y$ είναι η ληφθείσα κωδικοποιημένη λέξη στην έξοδο του αποδιαμορφωτή. Σε γενικές γραμμές, η λέξη Y μπορεί να εκφραστεί ως :

$$Y = C_m + e \qquad (3\text{-}18)$$

Όπου $e$ υποδηλώνει ένα αυθαίρετο λάθος δυαδικό φορέα. Η παράσταση YH' παράγει:

$$YH' = (Cm + e)H'$$

$$= CmH' + eH'$$

$$= eH' = S \qquad (3\text{-}19)$$

όπου το διάνυσμα S , (n - k) διαστάσεων ονομάζεται σύνδρομο του πρότυπου σφάλματος. Με άλλα λόγια, το διάνυσμα S έχει συνιστώσες που είναι μηδενικές όταν ικανοποιούνται οι εξισώσεις του ελέγχου ισοτιμίας και μη μηδενικές για όλες τις εξισώσεις του ελέγχου ισοτιμίας που δεν ικανοποιούνται. Έτσι, το σύνδρομο S , περιέχει τον τρόπο διεξαγωγής των αποτυχιών ελέγχων ισοτιμίας.

Τονίζουμε ότι το σύνδρομο S είναι ένα χαρακτηριστικό του πρότυπου σφάλματος και όχι της μεταδιδόμενης κωδικοποιημένης λέξης. Επιπλέον, παρατηρούμε ότι υπάρχουν $2^n$ πιθανά πρότυπα λάθους και μόνο $2^{n-k}$ σύνδρομα. Συνεπώς, διάφορα πρότυπα λάθους καταλήγουν στο ίδιο σύνδρομο.

Ας υποθέσουμε ότι έχουμε κατασκευάσει την πίνακα αποκωδικοποίησης στο οποίο θα περιέχονται όλες οι πιθανές κωδικοποιημένες λέξεις στην πρώτη σειρά, αρχίζοντας με τις μηδενικές κωδικοποιημένες λέξεις στην πρώτη (αριστερή πλέον) στήλη. Η μηδενική αυτή κωδικοποιημένη λέξη αντιπροσωπεύει επίσης τη συνιστώσα μηδέν του πρότυπου λάθους.
Συμπληρώνουμε την πρώτη στήλη από την απαρίθμηση όλων των $n-1$ πρότυπων

λάθους βάρους 1, στην περίπτωση όπου $n < 2^{n-k}$, θα προχωρήσουμε στην ένταξη προτύπων λάθους βάρους 2,έπειτα στην ένταξη προτύπων λάθους βάρους 3 κτλ. μέχρι να έχουμε συνολικά $2^{n-k}$ εγγραφές στην πρώτη στήλη. Έτσι, ο αριθμός των γραμμών που μπορούμε να έχουμε είναι $2^{n-k}$, που είναι ίσος με το αριθμό των συντελεστών του συνδρόμου. Έπειτα προσθέτουμε κάθε πρότυπο λάθος της πρώτης στήλης στην αντίστοιχη κωδικοποιημένη λέξη $C_m$. Έτσι συμπληρώνουμε τον πίνακα διάστασης $n \times (n-k)$,όπως αναφέρεται στο ως ακολούθως:

**Πίνακας 3.4** Πρότυπος πίνακας – Standard array

| | | | | |
|---|---|---|---|---|
| $C_1$ | $C_2$ | $C_3$ | ... | $C_{2^k}$ |
| $e_2$ | $C_2 + e_2$ | $C_3 + e_2$ | ... | $C_{2^k} + e_2$ |
| $e_3$ | $C_2 + e_3$ | $C_3 + e_3$ | ... | $C_{2^k} + e_3$ |
| ... | ... | ... | ... | ... |
| $e_{n-k}$ | $C_2 + e_{n-k}$ | $C_3 + e_{n-k}$ | ... | $C_{2^k} + e_{n-k}$ |

Ο πίνακας αυτός ονομάζεται πρότυπος πίνακας. Κάθε σειρά, συμπεριλαμβανομένης της πρώτης, αποτελείται από κ πιθανές ληφθείσες κωδικοποιημένες λέξεις που προκύπτουν από το αντίστοιχο πρότυπο σφάλμα της πρώτης στήλης. Κάθε σειρά ονομάζεται coset και η πρώτη (αριστερή πλέον) κωδικοποιημένη λέξη (ή πρότυπο λάθους) ονομάζεται coset leader. Ως εκ τούτου, ένα coset αποτελείται όλες τις πιθανές ληφθείσες κωδικοποιημένες λέξεις που προέρχονται από ένα συγκεκριμένο πρότυπο λάθους (coset leader).

### 3.1.7. Απόδοση RS κώδικα σε κανάλια με εκρηκτικό θόρυβο.

Οι Reed-Solomon κώδικες και άλλοι κώδικες βασισμένοι σε πεδία μεγαλύτερα του δυαδικού, έχουν κάποια εγγενή ικανότητα διόρθωσης εξαπλωμένων-εκρηκτικών δυαδικών σφαλμάτων. Για έναν κώδικα σε ένα τομέα $GF(2^m)$, κάθε κωδικοποιημένη λέξη μπορεί αποτελεί μια ακολουθία από m bits. Σύμφωνα με αυτή την ερμηνεία, ένας Reed-Solomon $RS(n,k)$ κώδικας πάνω σε ένα $GF(2^m)$ πεδίο ισοδυναμεί με ένα δυαδικό $(nm,km)$ κώδικα.

Ο RS κώδικας είναι ικανός για τη διόρθωση μέχρι t σύμβολα του λάθους. Δεν έχει σημασία ότι ένα ενιαίο σύμβολο μπορεί να έχει πολλαπλά bits κατά λάθος - εξακολουθεί να είναι ένα ενιαίο σύμβολο σφάλμα από την προοπτική του RS αποκωδικοποιητή. Ένα ενιαίο σύμβολο μπορεί να έχει μέχρι m bits ανά λάθος. Κάτω από τις καλύτερες των περιστάσεων, στη συνέχεια, όταν όλα τα λάθη επηρεάζουν τα παρακείμενα bits ενός συμβόλου, ο RS κώδικας μπορεί να διορθώσει μέχρι mt bits ανά λάθος. Αυτό σημαίνει ότι οι RS κώδικες είναι φυσικά αποτελεσματικοί για τη μετάδοση

πάνω από εκρηκτικά δυαδικά κανάλια: αφού εξαπλώσεις λαθών τείνουν να ομαδοποιούνται μαζί, μπορεί να υπάρχουν πολλά δυαδικά λάθη συμβάλλοντας σε ένα ενιαίο εσφαλμένο σύμβολο. Ως παράδειγμα από την αναφορά [9], μια έκρηξη του μήκους $3m + 1$ δεν μπορεί να επηρεάσει περισσότερο από 4 σύμβολα, έτσι και ένας κώδικας RS με ικανότητα διόρθωσης λαθών 4,μπορεί να διορθώσει τυχόν έκρηξη μήκους $3m + 1$ . Όμοια εφόσον οποιαδήποτε έκρηξη μήκους $m + 1$ δεν μπορεί να επηρεάσει περισσότερο από δύο bytes, οπότε κώδικας RS με ικανότητα διόρθωσης λαθών 4, θα μπορούσε να διορθώσει μέχρι δύο εκρήξεις εξαπλωμένων λαθών μήκους $m + 1$. Σε γενικές γραμμές, ένας κώδικας RS με ικανότητα διόρθωσης λαθών –t, πάνω στο $GF(2^m)$ πεδίο, μπορεί να διορθώσει τυχόν συνδυασμούς από

$$\frac{t}{1+\frac{(l+m-2)}{m}}$$

ή λιγότερες εκρήξεις λάθους μήκους $l$ , ή μια ενιαία έκρηξη μέχρι μήκους

$(t-1)m+1$ .

Φυσικά, όπως είναι γνωστό μέχρι τώρα, ένας RS κώδικας διορθώνει οποιοδήποτε συνδυασμό από $t$ ή λιγότερα τυχαία σφάλματα.

## *3.2. Reed – Solomon Κωδικοποίηση*

Οι Reed – Solomon κώδικες μπορούν να κωδικοποιηθούν ακριβώς όπως οι άλλοι κυκλικοί κώδικες με την προϋπόθεση πως οι πράξεις γίνονται στο σωστό κάθε φορά αριθμητικό πεδίο. Δίνοντας ένα διάνυσμα μηνύματος:

$m = (m_0, m_1, \ldots, m_{\kappa-1})$

Και το αντίστοιχο πολυώνυμο μηνύματος

$m(x) = m_0 + m_1 x + \ldots + m_{\kappa-1} x^{\kappa-1}$

όπου κάθε $m_i \in GF(q)$, προκύπτει συστηματική διαδικασία κωδικοποίησης

$$c(x) = m(x)x^{n-k} - R_{g(x)}[m(x)x^{n-k}] \qquad (3\text{-}20)$$

όπου $R_{g(x)}[\cdot]$ ορίζει την πράξη λήψης του υπολοίπου μετά την διαίρεση με το $g(x)$ .

Τυπικά ο κώδικας ορίζεται για κάποιο $m$ στο πεδίο $GF(2^m)$. Τα σύμβολα μηνύματος $m_i$ μπορούν να σχηματιστούν επιλέγοντας $m$ *bits* δεδομένων και αναπαριστώντας τα ως στοιχεία του $GF(2^m)$ πεδίου.

**Παράδειγμα 3.4**: Υποθέτουμε κώδικα $(n,k)=(15,9)$ ορισμένο στο πεδίο $GF(2^4)$ με πρωτεύον πολυώνυμο $p(x) = 1 + x + x^4$
και πολυώνυμο γεννήτριας $g(x)=a^6+a^9 x + a^6x^2 + a^4x^3+a^{14}x^4+a^{10}x^5+x^6$

Έστω $4-bit$ ροή δεδομένων $m = (5,2,1,6,8,3,10,15,4)$ προς κωδικοποίηση.
Το αντίστοιχο πολυώνυμο μηνύματος είναι:

$$m(x) = 5 + 2x + x^2 + 6x^3 + 8x^4 + 3x^5 + l0x^6 + 15x^7 + 4x^8$$

Χρησιμοποιώντας την αντιστοίχηση από:
Vector representation → power representation
$5=1010_2 \Leftrightarrow a^8$
$2=0010_2 \Leftrightarrow a$
$1=0001_2 \Leftrightarrow 1,$

όπως προκύπτει και από τον πίνακα 1.1 .
Εκφρασμένο σε μορφή δυνάμεων το πολυώνυμο μηνύματος είναι

$$m(x) = a^8 + ax + x^2 + a^5x^3+a^3x^4+a^4x^5+a^9x^6+a^{12}x^7+a^2x^8$$

το κωδικοποιημένο πολυώνυμο έπειτα από εφαρμογή συστηματικής κωδικοποίησης προκύπτει:

$$c(x) = a^8 + a^2x + a^{14}x^2 + a^3x^3 + a^5x^4 + ax^5 + a^8x^6 + ax^7 + x^8 +$$
$$+ x^8 + a^5x^9 + a^3x^{10} + a^4x^{11}+a^9x^{12}+a^{12}x^{13}+a^2x^{14}$$

όπου οι συντελεστές του μηνύματος υποδεικνύονται καθαρά ως οι τελευταίοι συντελεστές του κωδικοποιημένου πολυωνύμου.

Η εφαρμογή αυτή είναι χρήσιμη στην περίπτωση όπου $m = 8$ *bits*, ισοδύναμα με ένα *byte*, διότι ένα αρχείο που αποτελείται από $k-bytes$ ροή δεδομένων ερμηνεύεται ως μήνυμα $m(x)$ με αντίστοιχους συντελεστές του πολυωνύμου τα $k$ *bytes* προσαρμοσμένους σε μορφή δυνάμεων $GF(2^8)$ πεδίου.

### 3.2.1. Κωδικοποίηση κώδικα ως προέκταση κωδικοποίησης δυαδικών κυκλικών κωδίκων.

Η κωδικοποίηση κωδίκων λέξεων ενός δυαδικού κυκλικού κώδικα μπορεί να είναι μη συστηματική ή συστηματική, εξαρτάται κάθε φορά από την διαδικασία που επεξεργαζόμαστε το μήνυμα $\mathrm{m(x)}$.

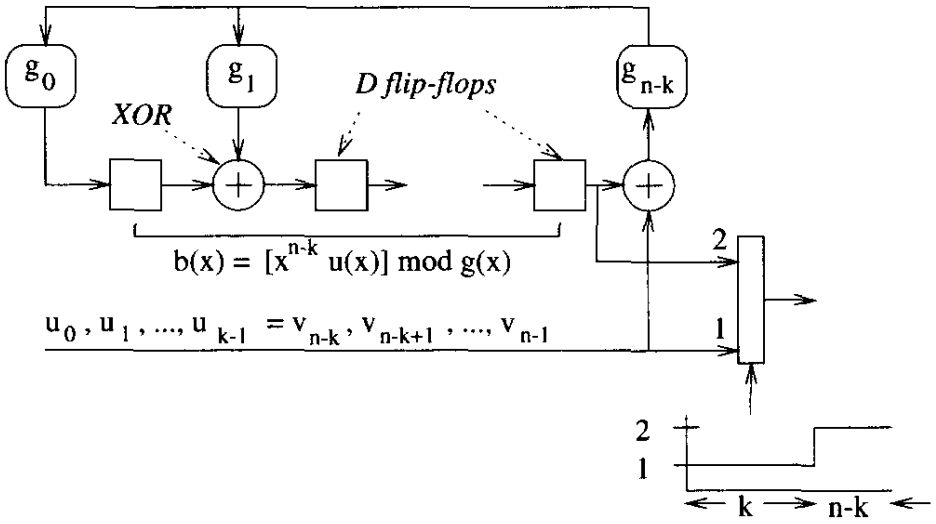
- Μη συστηματική κωδικοποίηση

$$c(x) = m(x)g(x) \qquad (3\text{-}21)$$

- Συστηματική κωδικοποίηση

$$c(x) = x^{n-k}m(x) + [x^{n-k}m(x) \bmod g(x)] \qquad (3\text{-}22)$$

Η κωδικοποίηση πραγματοποιείται με την αναπαράσταση ενός καταχωρητή γραμμικής ανατροφοδότησης κατάστασης, βαθμού ίσου με το βαθμό του πολυωνύμου $g(x)$ και συντελεστές βαρύτητας του καταχωρητή, τους αντίστοιχους συντελεστές του πολυωνύμου γεννήτριας του κώδικα. Στο παρακάτω σχήμα (**Σχήμα 3.2**) αναπαριστάται σχηματικά η μέθοδος συστηματικής κωδικοποίησης σε κύκλωμα ανατροφοδότησης.




**Σχήμα 3.2**Συστηματική κωδικοποίηση σε μορφή κυκλώματος ανατροφοδότησης κατάστασης

Έστω RS κώδικας $C_{RS}(7, 3)$,με δοσμένο το μήνυμα $m(x)$ 3-συμβόλων και το πολυώνυμο γεννήτριας (n-k=7-3) $4_{ου}$ βαθμού, θα υπολογίσουμε αναλυτικά την κωδικοποιημένη λέξη που προκύπτει και θα περιγράψουμε την διαδικασία μέσα από τα στάδια που πραγματοποιούνται στον καταχωρητή ανατροφοδότησης.

Το μήνυμα $m(x)$ 3-συμβόλων προς κωδικοποίηση είναι

$\mathrm{m}=\{\mathrm{a}^1,\mathrm{a}^3,\mathrm{a}^5\}=\{010\ \ 110\ \ 111\}$

Σε μορφή πολυωνύμου παίρνουμε $\mathrm{m(x)}=\mathrm{a}^1+\mathrm{a}^3x+\mathrm{a}^5x^2$ που πολλαπλασιασμένο με $x^{n-k}=x^4$ παράγει $\mathrm{a}^1x^4+\mathrm{a}^3x^5+\mathrm{a}^5x^6$. Έπειτα πραγματοποιώντας διαίρεση με το πολυώνυμο γεννήτριας που χρησιμοποιήσαμε και στην §1.1.4 $g(x)=a^3+a^1x+a^0x^2+a^3x^3+x^4$ και ακολουθώντας πράξεις πολλαπλασιασμού και πρόσθεσης στο $GF(8)$ μέσα από τους παραπάνω πίνακες προκύπτει

$$u(x)=a^0+a^2x+a^4x^2+a^6x^3+a^1x^4+a^3x^5+a^5x^6$$
$$=p(x)+x^{n-k}m(x)$$

Η μη δυαδική αυτή διαδικασία κωδικοποίησης είναι ανάλογη της δυαδικής με διαφορά πως οι συντελεστές μηνύματος, πολυωνύμου γεννήτριας και κωδικοποιημένης λέξης είναι σύμβολα που ανήκουν στο *galoi field* αντί των δυαδικών αριθμών.
Οι πολλαπλασιαστικοί παράγοντες (από αριστερά προς τα δεξιά) στο **Σχήμα 3.2** αντιστοιχούν στους συντελεστές του πολυωνύμου γεννήτριας $g(x)$ που περιγράφτηκε παραπάνω (από τον χαμηλό προς τον υψηλό βαθμό του πολυωνύμου).

Σε αντίθεση με τον απλό δυαδικό κυκλικό κώδικα όπου ο καταχωρητής κατάστασης αποθηκεύει τιμές 0 ή 1 , στην περίπτωση του RS κώδικα $\mathrm{C_{RS}}(7,3)$ στη μνήμη καταχωρούνται $3-bits$ σύμβολα με $2^m=2^3=8$ δυνατές τιμές. Οι θέσεις του καταχωρητή και στις δύο περιπτώσεις είναι $n-k$, ίσες δηλαδή με τον αριθμό των πλεοναζόντων στοιχείων ή τον βαθμό του πολυωνύμου $p(x)$. Είναι ήδη γνωστό πως στην έξοδο του καταχωρητή περιμένουμε στις πρώτες $k$ θέσεις τους όρους του πολυωνύμου $m(x)$ και στις υπόλοιπες $n-k$ θέσεις, τους όρους του $p(x)$ όπως προκύπτει και από την εξίσωση: $c(x)=p(x)+x^{n-k}m(x)$.

Η επεξεργασία του μηνύματος $\mathrm{m(x)}=\mathrm{a}^1+\mathrm{a}^3x+\mathrm{a}^5x^2$ μέσα από τον καταχωρητή ανατροφοδότησης κατάστασης παρουσιάζεται στο **Σχήμα 3.3** και η αναλυτική περιγραφή των βημάτων μέσα από κάθε ξεχωριστό κύκλο ρολογιού περιγράφεται από τα 5 παρακάτω βήματα καθώς και από τον **Πίνακας 3.4**.

- Κατά την διάρκεια των $k=3$ κύκλων ρολογιού ο διακόπτης 1 (*switch* 1) είναι κλειστός ώστε να διέρχονται τα σύμβολα του μηνύματος, κάθε ένα σε χρόνο ενός κύκλου ρολογιού, σαν πολλαπλασιαστικοί παράγοντες για την εκχώρηση τιμών στις $n-k$ θέσεις του καταχωρητή.

- Η έξοδος του καταχωρητή που βρίσκεται μετά τον δεύτερο διακόπτη περιέχει την τελική κωδικοποιημένη λέξη αποτελούμενη όπως γνωρίζουμε από $n$ σύμβολα. Τα πρώτα $k$ σύμβολα θα πρέπει να είναι τα στοιχεία του μηνύματος $\mathrm{m(x)}$. Έτσι γίνεται φανερό πως ταυτόχρονα με το κλείσιμο του διακόπτη 1 στους $k=3$ πρώτους κύκλους ρολογιού, θα πρέπει ο διακόπτης 2 να βρίσκεται στην κάτω

θέση (όπως απεικονίζεται και στο **Σχήμα 3.3**) ώστε τις πρώτες 3 θέσεις της εξόδου του καταχωρητή που αποτελούν τους πρώτους όρους της κωδικοποιημένης λέξης $u(x)$ που ζητάμε να καταλάβουν οι όροι του εισερχόμενου μηνύματος $\mathrm{m(x)}$.

- Στη συνέχεια η έξοδος του καταχωρητή θα πάρει τιμές που προκύπτουν από επεξεργασία του μηνύματος σε 3 χρονικούς κύκλους ρολογιού και θα πρέπει ο διακόπτης 2 να περιστραφεί και να κλείσει στην πάνω θέση.

- Οι υπόλοιποι $n-k$ όροι της κωδικοποιημένης λέξης προκύπτουν σε $n-k$ κύκλους ρολογιού με απλή μετακίνηση του περιεχομένου των $n-k$ θέσεων του καταχωρητή όπως είχε προκύψει αυτούσιο στο τέλος του $3_{ου}$ χρονικού κύκλου.

- Συνεπώς ο αριθμός των κύκλων ρολογιού για την παραλαβή της κωδικοποιημένης λέξης στην έξοδο του καταχωρητή είναι $n = 2^m - 1 = 7$.

**Πίνακας 3.5** Οι $k = 3$ πρώτοι κύκλοι ρολογιού και το περιεχόμενο στον καταχωρητή ανατροφοδότησης κατάστασης

| INPUT QUEUE | CLOCKCYCLE | REGISTER CONTENTS | FEEDBACK |
|---|---|---|---|
| $a^1$ $a^3$ $a^5$ | 0 | 0 0 0 0 | $a^5$ |
| $a^1$ $a^3$ | 1 | $a^1$ $a^6$ $a^5$ $a^1$ | $a^0$ |
| $a^1$ | 2 | $a^3$ 0 $a^2$ $a^2$ | $a^4$ |
| - | 3 | $a^0$ $a^2$ $a^4$ $a^6$ | - |

Η κωδικοποιημένη λέξη $u(x)$,της οποίας τα πρώτα $k$ στοιχεία είναι αυτά του μηνύματος και τα τελευταία $n-k$ προκύπτουν από τις $n-k$ θέσεις του καταχωρητή κατάστασης στο κ κύκλο ρολογιού, είναι όμοια με την αναμενόμενη από τις αλγεβρικές πράξεις όπως προέκυψε στην αρχή της ενότητας αυτής.

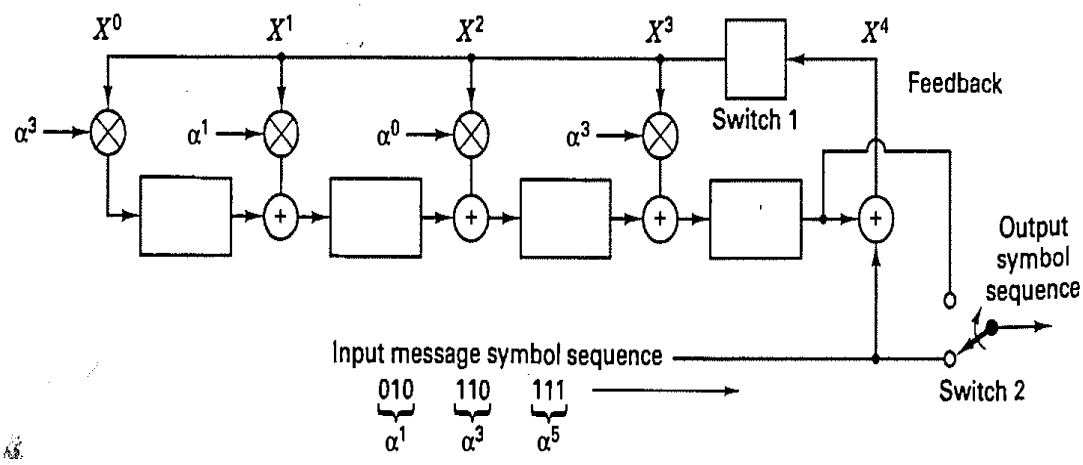
Όμως η κωδικοποιημένη λέξη $u(x)$ πρέπει να επαληθεύει και την εξίσωση

$$u(x) = m(x)g(x) \qquad (3\text{-}23)$$

εφόσον οι ρίζες του πολυωνύμου γεννήτριας $g(x)$ πρέπει να είναι ρίζες και τη κωδικοποιημένης λέξης $u(x)$. Η επαλήθευση του πορίσματος αυτού γίνεται και με την εύρεση της τιμής του πολυωνύμου $u(x)$ για όλες τις ρίζες του $g(x)$ όπου όπως προκύπτει είναι μηδενική. Για τις 4 ρίζες του $g(x)$ δηλαδή $a, a^2, a^3, a^4$ έχουμε με κατάλληλες πράξεις στο $GF(8)$ πεδίο

$$u(a) = u(a^2) = u(a^3) = u(a^4) = 0$$

όπου απεικονίζει το αναμενόμενο αποτέλεσμα πως οι ρίζες του $g(x)$ είναι ρίζες και της κωδικοποιημένης λέξης $u(x)$.




**Σχήμα 3.3** Αναπαράσταση Κωδικοποίησης $RS(7,3)$ κώδικα

### 3.2.2. Κωδικοποίηση $RS$ κωδίκων $C_{RS}(28,24)$, $C_{RS}(32,28)$ και $C_{RS}(255,251)$

Σε κανονική μορφή ένας $RS$ κώδικας σχεδιάζεται για επεξεργασία στο $GF(2^m)$ πεδίο και έχει μήκος $n = 2^m - 1$. Μια ενδιαφέρουσα εφαρμογή αυτών των κωδίκων είναι η σχεδίασή τους στο $GF(2^8)$ πεδίο, διότι κάθε στοιχείο δύναμης του $a$ ή αναπαράστασης διανύσματος, αποτελείται από 8 *bits* ή ένα *byte*. Ένας τέτοιος κώδικας σχεδιασμένος στο $GF(2^8)$ πεδίο και ικανός να διορθώσει $t = 2$ λάθη είναι ο $C_{RS}(255,251)$. Όμως γι' αυτόν τον κώδικα, ο πίνακας που περιέχει όλες τις δυνατές (έγκυρες) λέξεις θα είναι αρκετά μεγάλος. Εάν θεωρήσουμε ένα μικρότερο πίνακα όπου έχουμε διαγράψει έναν αριθμό $s_{RS}$ συμβόλων μηνύματος που τα θέτουμε μηδενικά $('0's)$ σε ένα ορισμένο πλήθος των δυνατών λέξεων και βέβαια ισχύει $1 \le s_{RS} \prec k$, τότε το μήκος του κώδικα είναι $n - s_{RS}$, ο αριθμός των συμβόλων μηνύματος είναι $k - s_{RS}$

και ο αριθμός των πλεοναζόντων συμβόλων είναι όπως και προηγουμένως $n-k$. Το πολυώνυμο γεννήτριας και η ικανότητα λάθους του περικομμένου κώδικα είναι το ίδιο με αυτού του αρχικού, αλλά ο νέος κώδικας δεν είναι κυκλικός διότι δεν ανήκουν όλες οι γραμμικές εναλλαγές *bits* συμβόλων του περικομμένου κώδικα στον περικομμένο κώδικα.

Από τα παραπάνω συμπεραίνουμε πως έχοντας ως δεδομένες τις ιδιότητες του κώδικα, δηλαδή αριθμό πλεοναζόντων συμβόλων και την ικανότητα διόρθωσης λαθών, μπορούμε να σχεδιάσουμε κώδικα χωρίς περιορισμό σε σταθερό μήκος κώδικα $n=2^m-1$. Οπότε ο αρχικός κύριος κώδικας $C_{RS}(28,24)$ και οι παράγωγοι αυτού $C_{RS}(32,28)$ και $C_{RS}(255,251)$, έχουν την ίδια ελάχιστη απόσταση $d_{\min}=5$ και μάλιστα οι παράγωγοι κώδικες χρησιμοποιούνται σε συστήματα κωδικοποίησης ελέγχου λάθους των $CD$. Η διαδικασία κωδικοποίησης των $CD$ χρησιμοποιεί έναν interleaver ανάμεσα στους δύο *shortened* κώδικες, που αναγκαστικά δημιουργεί καθυστέρηση μετάδοσης των *bytes* ενός δοσμένου συμβόλου. Με αυτόν τον τρόπο, το μη κωδικοποιημένο μήνυμα των 24-*bytes* κωδικοποιείται αρχικά από τον *shortened* κώδικα $C_{RS}(28,24)$ και παράγεται διάνυσμα 28-*bytes* , το οποίο ακολουθούμενο από την εφαρμογή του interleaver καταλήγει σε 28-*bytes* διάνυσμα που περιλαμβάνει στο εσωτερικό του *bytes* από προηγούμενη εφαρμογή κωδικοποίησης. Το τελευταίο διάνυσμα των 28-*bytes* είναι είσοδος σε κωδικοποιητή με κώδικα $C_{RS}(32,28)$ στο οποίο προστίθενται 4 *bytes* και έτσι δημιουργείται το τελικό διάνυσμα των 32-*bytes*.

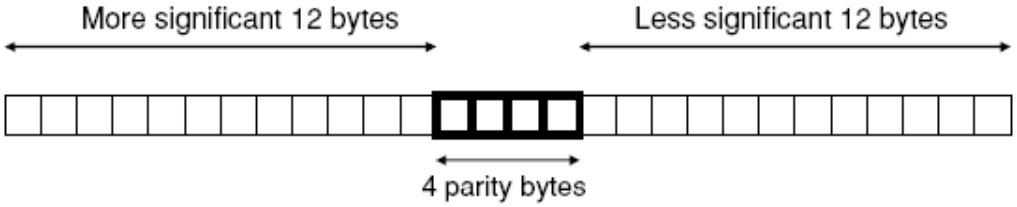
Ένα παράδειγμα κωδικοποίησης των *shortened* κωδίκων παρατίθεται, όπου με την διαδικασία της συστηματικής κωδικοποίησης που αναλύσαμε προηγουμένως, ένα διάνυσμα μηνύματος $m(x)$ πολλαπλασιάζεται με $x^{2t}=x^4$ δημιουργώντας το πολυώνυμο $x^4m(x)$ το οποίο στη συνέχεια διαιρείται με το πολυώνυμα γεννήτριας $g(x)$ του κώδικα. Για τον υπολογισμό του πολυωνύμου γεννήτριας , που είναι κοινό και για τους τρεις κώδικες, επεξεργαζόμαστε τα δεδομένα στο πεδίο του αρχικού $C_{RS}(255,251)$ κώδικα. Οπότε για δεδομένο τον αριθμό των πλεοναζόντων στοιχείων $n-k=4$ προκύπτουν 4 ρίζες διαδοχικών δυνάμεων του $a$ όπου με αναλυτικές πράξεις στο $GF(2^8)$ έχουμε :

$$g(x)=(x+a)(x+a^2)(x+a^3)(x+a^4) \qquad \text{(3-24)}$$

$$g(x)=a^{10}+a^{81}x+a^{251}x^2+a^{76}x^3+x^4 .$$

Το πολυώνυμο γεννήτριας που είναι κοινό και για τους 2 *shortened* κώδικες, $C_{RS}(28,24)$ και $C_{RS}(32,28)$, χρησιμοποιείται για τον υπολογισμό του πολυωνύμου υπολοίπου ή την εύρεση των πλεοναζόντων στοιχείων που προστίθενται κατά την διαδικασία κωδικοποίησης. Πράγματι, κατά την διαδικασία της κωδικοποίησης $C_{RS}(28,24)$ του αρχικού μηνύματος $m(x)$ των 24-*bytes* προκύπτει διάνυσμα των 28-*bytes* που όπως είναι γνωστό από την θεωρητική ανάλυσης τη συστηματικής κωδικοποίησης, το πολυώνυμο γεννήτριας $g(x)$ του κώδικα είναι παράγοντας του

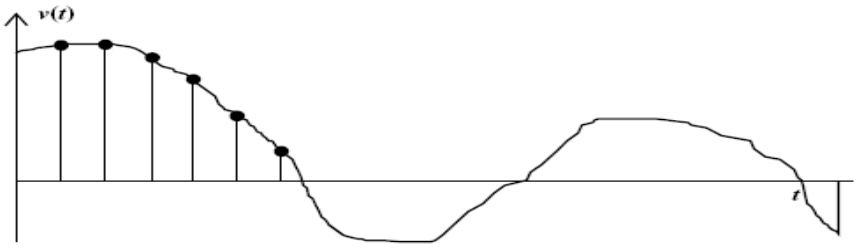
τελευταίου αυτού 28-*bytes* διανύσματος. Το μήνυμα όμως αυτό που αποτελεί και είσοδο της $C_{RS}(32,28)$ κωδικοποίησης μετακινείται προς τα δεξιά κατά 4 θέσεις και δημιουργεί την τελική κωδικοποιημένη λέξη των 32-*bytes*. Όμως επειδή και για τους δύο κώδικες το πολυώνυμο γεννήτριας είναι κοινό, παρατηρούμε πως οποιαδήποτε εισαγωγή 4 πλεοναζόντων συμβόλων το πολυώνυμο γεννήτριας είναι παράγοντας και του τελικού διανύσματος των 32-*bytes*. Άρα με μηδενικό διάνυσμα υπολοίπου η δεύτερη κωδικοποίηση είναι σαν να μην λαμβάνει καθόλου θέση στην συνολική διαδικασία και γι' αυτό μετά το τέλος της πρώτης κωδικοποίησης, τα 4 πλεονάζοντα εισάγονται στο μέσο του 28-*bytes* διανύσματος όπως εικονίζεται και στο **Σχήμα 3.4**.



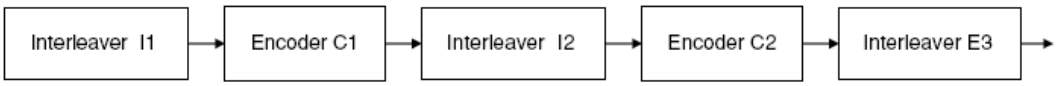

**Σχήμα 3.4** Διάνυσμα κωδικοποίησης παραγόμενο από τον $C_{RS}(28,24)$ κώδικα

### 3.2.3. Διαδικασία κωδικοποίησης των *CD* χρησιμοποιώντας *RS* κώδικες και interleavers

Το βασικό διάγραμμα βασίζεται στην δειγματοληψία ενός καναλιού ήχου και συγκεκριμένα παίρνουμε έξι δείγματα των 16 *bits* σε κάθε δεξί και αριστερό τμήμα του καναλιού. Το παρακάτω σχήμα παρουσιάζει την μια πλευρά του υπό δειγματοληψία σήμα εισόδου στην διαδικασία κωδικοποίησης των *CD*, η οποία απεικονίζεται στο **Σχήμα 3.6**.



**Σχήμα 3.5** Υπό δειγματοληψία σήμα ήχου που χρησιμοποιείται για την κωδικοποίηση $CD$.




**Σχήμα 3.6** Διαδικασία κωδικοποίησης των $CD$.

Κάθε διάνυσμα των 24-*bytes* αποτελείται από αναμειγμένα δείγματα (των 16 *bits* ή 2 *bytes*) του αριστερού (*right* : $R$) και δεξιού (*left* : $L$) τμήματος του σήματος που υπόκειται σε δειγματοληψία όπως απεικονίζεται στο **Σχήμα 3.6**. Το διάνυσμα αυτό των 24-*bytes* υπόκειται στην παρακάτω διαδικασία :

*Interleaver I1*: Διαχωρίζει το διάνυσμα των 24-*bytes* σε δύο διαφορετικές χρονικές θυρίδες ανάλογα με το αν τα *bytes* προέρχονται από το αριστερό ή δεξί τμήμα του διαγράμματος του ηχητικού σήματος. Τα κενά που δημιουργούνται στα νέα διανύσματα-χρονοθυρίδες καλύπτονται με δεδομένα από προηγούμενη επεξεργασία.

*Κωδικοποιητής* $C1$ : Είναι *shortened* έκδοση $C_{RS}(28,24)$ του $C_{RS}(255,251)$ κώδικα που προσθέτει 4 *bytes* και δημιουργεί κωδικοποιημένο 28-*bytes* διάνυσμα.

*Interleaver I2* : Δημιουργεί μια σταθερή καθυστέρηση μεταξύ διαδοχικών θέσεων των *bytes*, χρήσιμο όπως επισημάναμε και στην ενότητα 1.2.3 ώστε να μην προκύψει μηδενικό πολυώνυμο υπολοίπου κατά την εφαρμογή του κωδικοποιητή $C2$.

*Κωδικοποιητής* $C2$ : Είναι *shortened* έκδοση $C_{RS}(32,28)$ του $C_{RS}(255,251)$ κώδικα που προσθέτει 4 *bytes* και δημιουργεί κωδικοποιημένο 32-*bytes* διάνυσμα.

*Interleaver I3: Δημιουργεί* καθυστερήσεις και εναλλαγές στοιχείων για την διευκόλυνση χειρισμού διαδικασίας παρεμβολής που παίρνει μέρος μετά την αποκωδικοποίηση και που και που κάνει ανεπαίσθητη οποιαδήποτε παρουσία λάθους που προστέθηκε κατά την αντιστροφή του διανύσματος σε αναλογικό σήμα ήχου.

## 3.3. Αποκωδικοποίηση RS

### 3.3.1. Αρχιτεκτονική αποκωδικοποιητή.

Η βασική ιδέα αλγορίθμων $RS$ αποκωδικοποιητή είναι ίδια με αυτή των δυαδικών $BCH$ κωδίκων. Η μόνη διαφορά τους είναι ότι οι τιμές λάθους , $e_{jl}$, $1 \le l \le v$ με $v \le t_d$, σε αντίθεση με τους δυαδικούς κώδικες που είναι 0 ή 1 αντίθετα με την κωδικοποιημένη πληροφορία στην θέση λάθους, εδώ πρέπει να υπολογιστούν. Γενικότερα, για τον υπολογισμό αυτό χρησιμοποιείται ο αλγόριθμος $Forney$, όπου αν υποθέσουμε πως οι ρίζες του πολυωνύμου γεννήτριας και συνεπώς και της κωδικοποιημένης λέξεις είναι το σύνολο $2t_d$ -συμβόλων : $\{a^b, a^{b+1}, ..., a^{b+2t_d-1}\}$ τότε:

$$e_{jl} = \frac{(a^{jl})^{2-b}\Omega(a^{-jl})}{\Lambda'(a^{-jl})}. \qquad (3\text{-}25)$$

Όπου $\Lambda'(x)$ είναι η παράγωγος του πολυωνύμου θέσης λάθους $\Lambda(x)$ όπως υπολογίζεται στη συνέχεια και $\Omega(x)$ είναι πολυώνυμο εκτίμησης λάθους που ορίζεται ως εξής:

$$\Omega(x) = \sigma(x)S(x) \bmod x^{2t_d+1}. \qquad (3\text{-}26)$$

Όσον αφορά τον υπολογισμό του συνδρόμου $S(x)$, διευκολύνεται με την παράθεση του υπολογισμού του στην περίπτωση απλών δυαδικών κωδίκων και στην ακόλουθη επέκταση και γενίκευση της θεωρίας αυτής για τους $RS$ κώδικες που εξετάζουμε. Για τους γραμμικούς αρχικά κώδικες, το σύνδρομο υπολογίζεται από το γινόμενο $S = rH$ ,όπου για κάθε τιμή του συνδρόμου αντιστοιχεί σε ένα πρότυπο μήνυμα λάθους μέσα από τους κατάλληλους πίνακες και με την πρόσθεση του λάθους στο λαμβανόμενο μήνυμα μπορούμε να εξάγουμε το αρχικά απεσταλμένο μήνυμα.
Για τους κυκλικούς κώδικες, σχεδιάζουμε κύκλωμα καταχωρητή ανατροφοδότησης, όμοιο με αυτό του στάδιο κωδικοποίησης και με έκφραση του λαμβανόμενου μηνύματος σε μορφή πολυωνύμου $r(x)$ ακολουθούν οι υπολογισμοί:

$$\begin{aligned} & r(x) = u(x) + e(x) \\ & \rightarrow r(x) = m(x)g(x) + e(x) \end{aligned} \qquad (3\text{-}27)$$

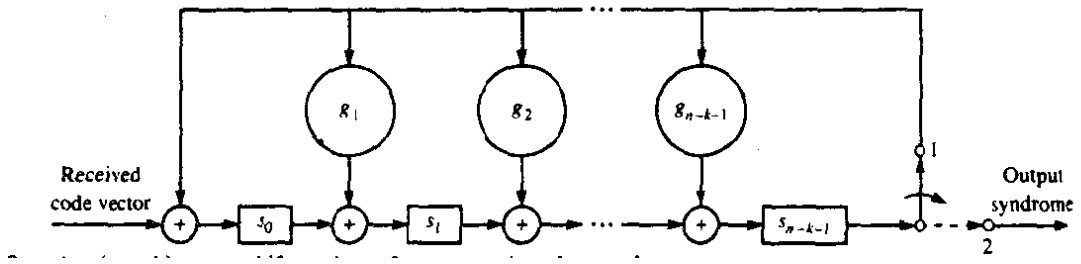
Υποθέτοντας την διαίρεση του $r(x)$ με το $g(x)$ έστω ότι λαμβάνουμε υπόλοιπο $R(x)$ βαθμού μικρότερου ή ίσου του $n-k-1$ με

$$r(x) = Q(x)g(x) + R(x) \qquad (3\text{-}28)$$

και σύμφωνα με την προηγούμενη ισότητα, προκύπτει:

$$e(x) = (m(x) + Q(x))g(x) + R(x) \qquad (3\text{-}29)$$

Το υπόλοιπο της διαίρεσης του $r(x)$ με το $g(x)$ βαθμού μικρότερου ή ίσου του $n-k-1$ αποτελεί το σύνδρομο $S(x)$ και θα μπορούσε να υπολογισθεί μέσα από ένα καταχωρητή όπως παρουσιάζεται στο **Σχήμα 3.7**



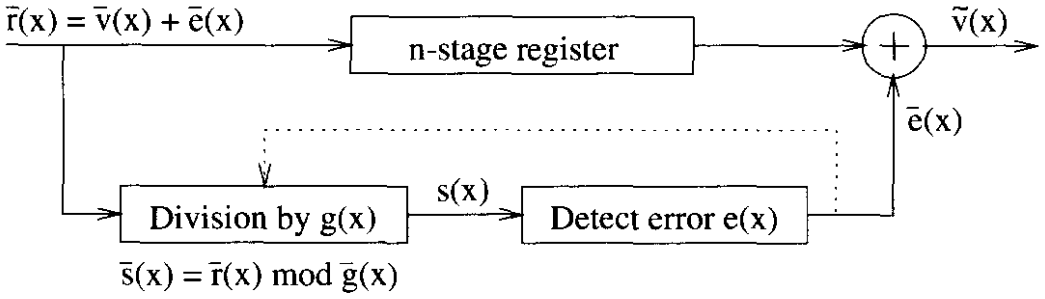

**Σχήμα 3.7** Ένας $n-k$ καταχωρητής ανατροφοδότησης για τον υπολογισμό του συνδρόμου $S(x)$

Εάν το πολυώνυμο γεννήτριας $g(x)$ διαιρεί ακριβώς το ληφθέν σήμα $r(x)$ τότε το υπόλοιπο της διαίρεσης $S(x)$ είναι μηδενικό και δεν έχει παρεμβληθεί κανένα λάθος κατά την διαδικασία της κωδικοποίησης. Αρχικά οι θέσεις του καταχωρητή είναι μηδενικές και ο διακόπτης είναι κλειστός στην θέση 1.Στη συνέχεια εισάγοντας στον καταχωρητή τα $n$ *bits* του μηνύματος $r(x)$ ,προκύπτουν οι $n-k$ θέσεις του καταχωρητή με τις τιμές του συνδρόμου $\{S_0, S_1, ..., S_{n-k-1}\}$ όπως εικονίζονται στο **Σχήμα 3.7**.Κλείνοντας τον διακόπτη στην θέση 2 και σε διάρκεια $n-k$ κύκλων ρολογιού παραλαμβάνουμε στην έξοδο του καταχωρητή τιμές του συνδρόμου.

**Παράδειγμα 3.5**: Για ένα απλό δυαδικό κυκλικό κώδικα παραθέτουμε την διαδικασία διόρθωσης λάθους *bit*. Υποθέτουμε ότι ένα λάθος σημειώνεται στην θέση που αντιστοιχεί στο $x^{n-1}$, $\overline{e(x)} = x^{n-1}$, που σημαίνει πως λάθος είναι το πρώτο *bit* της λαμβανόμενης λέξης. Από τον υπολογισμό του συνδρόμου $S(x)$ και μέσω του πίνακα αντιστοίχησης βρίσκουμε το αντίστοιχο λάθος $\overline{e(x)} = x^{n-1}$ για την συγκεκριμένη τιμή του συνδρόμου. Όμως η παρουσία και ανίχνευση ενός λάθους δεν μποδίζει την ανίχνευση τυχών άλλων λαθών. Λόγω της κυκλικότητας του κώδικα ισχύει προσεγγιστικά η παρακάτω σχέση :

$$S(x) = \overline{e(x)} \bmod g(x). \qquad (3\text{-}30)$$

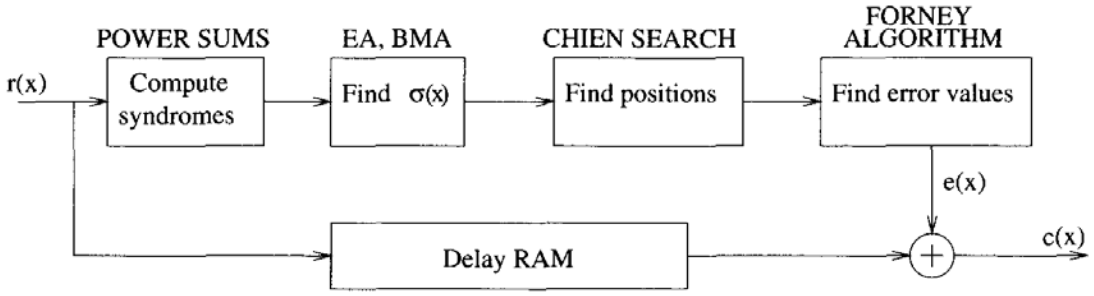
Από την κυκλική εναλλαγή των τιμών $\{S_0, S_1, ..., S_{n-k-1}\}$ του συνδρόμου μπορούμε να καταλήξουμε πως όλα τα λάθη έχουν ανιχνευτεί και διορθωθεί όταν όλες οι τιμές του συνδρόμου είναι μηδενικές. Η παραπάνω ανάλυση του αποκωδικοποιητή κυκλικού κώδικα αναπαρίσταται σε μορφή κυκλώματος στο **Σχήμα 3.8**.




**Σχήμα 3.8** Αρχιτεκτονική αποκωδικοποιητή κυκλικού κώδικα

### 3.3.2. Βασική ιδέα αποκωδικοποίησης *RS* κώδικα.

Το διάγραμμα ενός *RS* αποκωδικοποιητή, που αποτελείται από ψηφιακά κυκλώματα και διαδικασίες επεξεργασίας των συμβόλων παρατίθεται στο **Σχήμα 3.9**.




**Σχήμα 3.9** Αρχιτεκτονική $RS$ κωδικοποιητή με επεξεργασία συμβόλων στο $GF(2^m)$ πεδίο

Σύμφωνα με το παραπάνω διάγραμμα εξάγουμε πως για τον υπολογισμό του ορθού κωδικοποιημένου διανύσματος πρέπει να ακολουθήσουμε πέντε βασικά βήματα:

*1. Υπολογισμός των τιμών του συνδρόμου*

Οι τιμές του συνδρόμου καθορίζουν εάν το ληφθέν σήμα $r(x)$ ανήκει στο σύνολο των έγκυρων κωδικοποιημένων λέξεων. Επειδή το πολυώνυμο $g(x)$ αποτελεί παράγοντα του κωδικοποιημένου μηνύματος $(U(x) = m(x)g(x))$ οι ρίζες του $g(x)$ μηδενίζουν το $U(x)$. Εάν το ληφθέν μήνυμα $r(x)$ ανήκει στις έγκυρες κωδικοποιημένες λέξεις, τότε θα

πρέπει να μηδενίζεται και το $r(x)$ για τις ρίζες του $g(x)$ $(r(x) = U(x) + g(x))$. Ο υπολογισμός του συνδρόμου περιγράφεται από την σχέση:

$$S_i = r(x)_{x=a^i} = r(a^i), \quad i = 1, ..., n-k \qquad (3\text{-}31)$$

Όπου οποιοδήποτε μη μηδενικό αποτέλεσμα είναι ένδειξη παρουσίας λάθους.

Η βασική ιδέα κωδικοποίησης $RS$ κώδικα έγκειται στην εισαγωγή των στοιχείων β $\in GF(2^m)$ για την αρίθμηση των θέσεων μιας κωδικοποιημένης λέξης ή ισοδύναμα της σειράς των συντελεστών του αντίστοιχου πολυωνύμου. Η αρίθμηση αυτή απεικονίζεται στο **Σχήμα 3.10** για ένα διάνυσμα $\{r_0, r_1, ..., r_{n-1}\}$ του αντίστοιχου $r(x)$ πολυωνύμου.

| values | $r_0$ | $r_1$ | $r_{n-1}$ |
|---|---|---|---|
| positions | 1 | $\alpha$ | $\alpha^{n-1}$ |

**Σχήμα 3.10** Αρίθμηση θέσεων του κωδικοποιημένου διανύσματος $r(x)$ χρησιμοποιώντας στοιχεία του $GF(2^m)$ πεδίου

Υποθέτουμε ότι έχουν προκύψει ν λάθη στις θέσεις $X^{j_1}, X^{j_2}, ..., X^{j_v}$. Τότε το πολυώνυμο λάθους γράφεται ως εξής;

$$e(X) = e_{j_1} X^{j_1} + e_{j_2} X^{j_2} + ... + e_{j_v} X^{j_v} \qquad (3\text{-}32)$$

Οι δείκτες 1,2,…,ν αναφέρονται στο $1^{st}, 2^{nd}, ..., v^{th}$ λάθος και ο δείκτης $j$ αναφέρεται στη θέση λάθους. Προκειμένου να διορθώσουμε την αλλοιωμένη κωδικοποιημένη λέξη θα πρέπει να καθορίσουμε την θέση του λάθους $X^{j_l}$ και την αντίστοιχη τιμή του $e_{j_l}$.Οι θέσεις λάθους μπορούν να βρεθούν από τις ρίζες του $g(x)$ $a^{j_l}$ , όπου $\beta_l = a^{j_l}$ με $b \le j \le b + 2t_d - 1$ , και $v \le t_d$ είναι ο αριθμός των λαθών.

Οι τιμές των συνδρόμων είναι η εκτίμηση του πολυωνύμου $r(x)$ για κάθε μηδενικό $a^{j_l}$ του κώδικα :

$$S_j = \sum_{l=1}^{v} e_{i_l} (\alpha^j)^{i_l} = \sum_{l=1}^{v} e_{i_l} (\alpha^{i_l})^j.$$

(3-33)

$$
\begin{aligned}
S_1 &= r(\alpha^b) = e_{j_1}\alpha^{bj_1} + \cdots + e_{j_\nu}\alpha^{bj_\nu} \\
S_2 &= r(\alpha^{b+1}) = e_{j_1}\alpha^{(b+1)j_1} + \cdots + e_{j_\nu}\alpha^{(b+1)j_\nu} \\
&\vdots \qquad\qquad \vdots \\
S_{2t_d} &= r(\alpha^{b+2t_d-1}) = e_{j_1}\alpha^{(b+2t_d-1)j_1} + \cdots + e_{j_\nu}\alpha^{(b+2t_d-1)j_\nu}
\end{aligned}
$$

Προκύπτουν $2t$ εξισώσεις με $\nu \le t$ άγνωστες τιμές θέσεις λαθών και $\nu$ άγνωστες εκτιμήσεις των λαθών στις θέσεις αυτές. Ι επίλυση των εξισώσεων αυτών δεν είναι άμεση λόγω της μη γραμμικότητας του συστήματος καθώς υπάρχουν άγνωστοι υψωμένοι σε δυνάμεις. Οι τεχνικές που χρησιμοποιούνται με σκοπό την επίλυση των παραπάνω εξισώσεων ονομάζονται αλγόριθμοι Re*ed – Solomon* αποκωδικοποίησης.
Την προσπάθεια επίλυσης των εξισώσεων διευκολύνει η εισαγωγή ενός νέου πολυωνύμου, το πολυώνυμο θέσης λάθους, που ορίζεται από την σχέση:

$$\Lambda(x) = \prod_{l=1}^{\nu}\left(1 - X_l x\right) = \Lambda_\nu x^\nu + \Lambda_{\nu-1}x^{\nu-1} + \ldots + \Lambda_1 x + \Lambda_0 \qquad (3\text{-}34)$$

Όπου $\Lambda_0$=1. Με τον ορισμό αυτό εάν $x = X_l^{-1}$ τότε $\Lambda(x)$=0. Έπεται λοιπόν πως οι ρίζες του πολυωνύμου θέσης λάθους είναι οι αντίστροφοι (υπολογισμένοι στο $GF(2^m)$ πεδίο ) των άγνωστων θέσεων λάθους.

*2. Υπολογισμός του πολυωνύμου* $\Lambda(x)$ *και επίλυση των αλγορίθμων BMA* και *EA.*

Ενώ υπάρχει μη γραμμική εξάρτηση των συντελεστών του πολυωνύμου θέσης λάθους και των θέσεων των λαθών, ανιχνεύεται γραμμική σχέση μεταξύ των τιμών του συνδρόμου $\{S_0, S_1, \ldots, S_{n-k-1}\}$ και των συντελεστών του πολυωνύμου ένδειξης λάθους $\Lambda(x)$.
Η συσχέτιση αυτή περιγράφεται από τις παρακάτω εξισώσεις:

$$S_k + \Lambda_1 S_{k-1} + \ldots + \Lambda_{k-1}S_1 + k\Lambda_k = 0 \qquad 1 \le k \le \nu$$

$$S_k + \Lambda_1 S_{k-1} + \ldots + \Lambda_{\nu-1}S_{k-\nu+1} + \Lambda_\nu S_{k-\nu} = 0 \qquad k \succ \nu$$

Συνεπώς:

$$
\begin{aligned}
k = 1: &\quad S_1 + \Lambda_1 = 0 \\
k = 2: &\quad S_2 + \Lambda_1 S_1 + 2\Lambda_2 = 0 \\
&\quad \vdots \\
k = \nu: &\quad S_\nu + \Lambda_1 S_{\nu-1} + \Lambda_2 S_{\nu-2} + \cdots + \Lambda_{\nu-1} S_1 + \nu\Lambda_\nu = 0 \\
\\
k = \nu+1: &\quad S_{\nu+1} + \Lambda_1 S_\nu + \Lambda_2 S_{\nu-1} + \cdots + \Lambda_\nu S_1 = 0 \\
k = \nu+2: &\quad S_{\nu+2} + \Lambda_1 S_{\nu+1} + \Lambda_2 S_\nu + \cdots + \Lambda_\nu S_2 = 0 \\
&\quad \vdots \\
k = 2t: &\quad S_{2t} + \Lambda_1 S_{2t-1} + \Lambda_2 S_{2t-2} + \cdots + \Lambda_\nu S_{2t-\nu} = 0.
\end{aligned}
$$

Για $k \succ \nu$ υπάρχει σχέση γραμμικής ανατροφοδότησης μεταξύ των τιμών του συνδρόμου και των συντελεστών του πολυωνύμου $\Lambda(x)$.

$$S_j = -\sum_{i=1}^{\nu} \Lambda_i S_{j-i}.$$

Ισοδύναμα, εκφρασμένη η εξίσωση σε μορφή πίνακα :

$$
\begin{bmatrix}
S_1 & S_2 & \cdots & S_\nu \\
S_2 & S_3 & \cdots & S_{\nu+1} \\
S_3 & S_4 & \cdots & S_{\nu+2} \\
\vdots & & & \\
S_\nu & S_{\nu+1} & \cdots & S_{2\nu-1}
\end{bmatrix}
\begin{bmatrix}
\Lambda_\nu \\ \Lambda_{\nu-1} \\ \Lambda_{\nu-2} \\ \vdots \\ \Lambda_1
\end{bmatrix}
= -
\begin{bmatrix}
S_{\nu+1} \\ S_{\nu+2} \\ \vdots \\ S_{2\nu}
\end{bmatrix}.
$$

Ο $\nu \times \nu$ πίνακας που τον ορίζουμε ως $M_\nu$ είναι ένας *Toeplitz* πίνακας με σταθερά τα στοιχεία της διαγωνίου. Ο $\nu$ αριθμός των αγνώστων δεν είναι εκ των προτέρων γνωστός και πρέπει να καθοριστεί. Για τον υπολογισμό των συντελεστών του πολυωνύμου θέσης λάθους μέσω της σχέσης μεταξύ πινάκων ακολουθούμε τα παρακάτω βήματα μέσω του *Peterson – Gorenstein – Zierler* αποκωδικοποιητή.

- Θέτουμε $\nu = t$
- Σχηματίζουμε τον πίνακα $M_\nu$ και υπολογίζουμε την ορίζουσα $\det(M_\nu)$. Εάν ο πίνακας είναι μη αντιστρέψιμος ($\det(M_\nu)$) τότε θέτουμε $\nu = \nu - 1$ και επαναλαμβάνουμε το βήμα αυτό.
- Εάν ο πίνακας $M_\nu$ είναι αντιστρέψιμος λύνουμε ως προς τους συντελεστές $\{\Lambda_\nu, \Lambda_{\nu-1}, ..., \Lambda_1\}$ και υπολογίζουμε τις τιμές τους.

Οι πιο συνηθισμένοι μέθοδοι επίλυσης της παραπάνω εξίσωσης είναι οι ακόλουθοι:

*Berlekamp – Massey Αλγόριθμος (BMA)*

Ο Αλγόριθμος αυτός ανακαλύφθηκε από τους *Berlekamp* και *Massey*. Είναι υπολογιστικά αποτελεσματικός τρόπος επίλυσης της παραπάνω εξίσωσης, μέσα από το πρίσμα των διαδικασιών που απαιτούνται στο $GF(2^m)$ πεδίο. Είναι δημοφιλής επιλογή για την προσομοίωση και εφαρμογή των *RS* αποκωδικοποιητών στο λογισμικό.

1. Αρχικοποιούμε τον αλγόριθμο με $\Lambda(x)=1$, όπου $\Lambda(x)$ το έχουμε ήδη αναφέρει ως πολυώνυμο σύνδεσης για γραμμικούς καταχωρητές ανατροφοδότησης κατάστασης (*LFSR*). Όρος διόρθωσης $\rho(x)$ με $\rho(x)=x$, μετρητής συνδρόμου $i$ με $i=1$, μήκος καταχωρητή $l$ με $l=0$.

2. Υπολογίζουμε την τιμή του συνδρόμου $S_{i+1}$ και την τιμή της ασυμφωνίας $d$

$$d = S_i + \sum_{j=1}^{l} \Lambda_j S_{i-j} \qquad (3\text{-}35)$$

3. Εξετάζουμε την τιμή της ασυμφωνίας $d$, εάν $d=0$ εκτελούμε το βήμα 8.

4. Σχηματίζουμε νέο πολυώνυμο συνδέσμου
$\Lambda_{new}(x) = \Lambda(x) - d\,\rho(x)$ (3-36)

5. εξετάζουμε το μήκος του καταχωρητή, εάν $2l \geq i$ εκτελούμε το βήμα 7

6. Μεταβάλλουμε το μήκος του καταχωρητή σε $l = i - l$ και τον όρο διόρθωσης σε
$\rho(x) = \Lambda(x)/d$. (3-37)

7. Τα πολυώνυμο συνδέσμου παίρνει την τιμή $\Lambda(x) = \Lambda_{new}(x)$.

8. Ο όρος διόρθωσης έχει νέα τιμή: $\rho(x) = x\rho(x)$

9. Ο μετρητής συνδρόμου αυξάνεται κατά μια μονάδα : $i = i+1$

10. Συνθήκη τέλους. Εάν $i \prec d$ εκτέλεσε το βήμα 2, αλλιώς τέλος.

**Παράδειγμα 3.6**: Υποθέτουμε κώδικα $RS(7,3,5)$ με $r(x) = ax^2 + a^5x^4$ το ληφθέν μήνυμα. Οι τιμές του συνδρόμου για $n-k=7-3=4$ ρίζες του κώδικα είναι :
$S_1 = r(1) = a^6$, $S_2 = r(a) = a^5$, $S_3 = r(a^2) = a$, $S_4 = r(a^3) = a$
Συνεπώς:

$$\begin{bmatrix} a^6 & a^5 \\ a^5 & a \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} a \\ a \end{bmatrix}$$

Σύμφωνα με τον *Berlekamp – Massey* (BMA) αλγόριθμο προκύπτει :

$i=0:\Lambda(x)=1,l=0,\rho(x)=x.$

$i=1:d=S_1=a^6,$

$\Lambda_{new}(x)=\Lambda(x)+d\rho(x)=1+a^6x$

$2l=0\prec i,l=i-l=1,$

$\rho(x)=\Lambda(x)/d=a^{-6}=a,$

$\rho(x)=x\rho(x)=ax,\quad \Lambda(x)=\Lambda_{new}(x).$

$i=2:d=S_2+\sum_{j=1}^{l}\Lambda_j S_{2-j}=a^5+a^6a^6=0$

$\rho(x)=x\rho(x)=ax^2.$

$i=3:d=S_3+\sum_{j=1}^{l}\Lambda_j S_{3-j}=a+a^6a^5=a^2$

$\Lambda_{new}(x)=\Lambda(x)+d\rho(x)=1+a^6x+a^3x^2,$

$2l=2\prec i,l=i-l=2,$

$\rho(x)=\Lambda(x)/d=a^5+a^4x$

$\rho(x)=x\rho(x)=a^5x+a^4x^2\ ,\quad \Lambda(x)=\Lambda_{new}(x)$

$i=4:d=S_4+\sum_{j=1}^{l}\Lambda_j S_{4-j}=a+a^6a+a^3a^5=1$

$\Lambda_{new}(x)=\Lambda(x)+d\rho(x)=1+a^6x+a^3x^2+(1)(a^5x+a^4x^2)$

$=1+ax+a^6x^2$

$2l\geq 4$

$\rho(x)=x\rho(x)=a^5x^2+a^4x^3,\Lambda(x)=\Lambda_{new}(x)$

$i=5>d.Stop$

*Euclidean Αλγόριθμος (ΕΑ)*

Ο αλγόριθμος αυτός είναι μέθοδος επίλυσης της παραπάνω εξίσωσης μεταξύ πινάκων σε πολυωνυμική πλέον μορφή και χρησιμοποιείται ευρύτατα σε εφαρμογές υλικού ( *hardware* ) που σχετίζονται με αποκωδικοποιητές *RS* και *BCH* . Ο αλγόριθμος αυτός χρησιμοποιεί εκτός από το πολυώνυμο θέσης και άλλα ενδιάμεσα πολυώνυμα όπως είναι το πολυώνυμο εκτίμησης λάθους $\Omega(x)$ .

$$\Omega(x)=S(x)\Lambda(x)\bmod x^{2t} \qquad (3\text{-}38)$$

Όπου χρησιμοποιώντας πολυώνυμο $\Theta(x)$ έχουμε ισοδύναμα:

$$\Theta(x)x^{2t}+S(x)\Lambda(x)=\Omega(x) \qquad (3\text{-}39)$$

Ο επεκταμένος ευκλείδειος αλγόριθμος χρησιμοποιώντας την αρχή του μέγιστου βαθμού κοινού διαιρέτη $GCD(Greatest-Common-Division)$ μεταξύ δύο πολυωνύμων έτσι ώστε βαθμός του $\Omega(x) < t$, ακολουθεί τα παρακάτω βήματα:

1. Έστω $as + bt = c$ , όπου $c$ είναι το $GCD$ μεταξύ του $a$ και $b$. Υπολογίζουμε τις τιμές των συνδρόμων και δημιουργούμε το πολυώνυμο συνδρόμου $S(x) = S_1 + S_2 x + ... + S_{2t} x^{2t-1}$

2. Θέτουμε $a(x) = x^{2t}$ και $a(x) = x^{2t}$. Τρέχουμε την παραπάνω εξίσωση μέχρι $\deg(r_i(x)) < t$ και στη συνέχεια θέτουμε $\Omega(x) = r_i(x)$ και $\Lambda(x) = t_i(x)$.

3. Αναζητούμε τις ρίζες του $\Lambda(x)$. Θέσεις λάθους είναι τα αντίστοιχα $x_i$.

**Παράδειγμα 3.7**: Υποθέτουμε κώδικα $RS(7,3,5)$ όμοιο με αυτό του παραδείγματος 1.6, με ληφθέν μήνυμα $r(x) = ax^2 + a^5 x^4$

Σύμφωνα με τον $Euclidean$ αλγόριθμο έχουμε:

$j = 1$

$r_0(x) = x^5,$

$r_1(x) = S(x) = 1 + a^6 x + a^5 x^2 + ax^3 + ax^4,$

$b_0(x) = 0,$

$b_1(x) = 1$

$j = 2$

$x^5 = (1 + a^6 x + a^5 x^2 + ax^3 + ax^4)(a^6 x + a^6) + a^5 x^3 + x^2 + ax + a^6.$

$r_2(x) = a^5 x^3 + x^2 + ax + a^6,$

$q_2(x) = a^6 x + a^6,$

$b_2(x) = 0 + (a^6 x + a^6)(1) = a^6 x + a^6.$

$j = 3$

$1 + a^6 x + a^5 x^2 + ax^3 + ax^4 = (a^5 x^3 + x^2 + ax + a^6)(a^3 x + a^2) + a^6 x^2 + ax + a^3.$

$r_3(x) = a^6 x^2 + ax + a^3,$

$q_3(x) = a^3 x + a^2,$

$b_3(x) = 1 + (a^3 x + a^2)(a^6 x + a^6) = a^3 + a^4 x + a^2 x^2.$

$\deg(r_3(x)) = 2 = t.$

*Direct Αλγόριθμος (Άμεση λύση)*

Η μέθοδος αυτή προτάθηκε πρώτα από τον *Peterson* όπου ο υπολογισμών των ριζών του πολυωνύμου θέσης λάθους $\Lambda(x)$ γίνεται άμεσα με την επίλυση ενός συνόλου γραμμικών εξισώσεων. Ο όρος *PGZ* (*Peterson – Gorenstein – Zierler*) αποκωδικοποιητής χρησιμοποιείται συχνά στην βιβλιογραφία διότι η μέθοδος αυτή εφαρμόζεται για την αποκωδικοποίηση μη δυαδικών *BCH* και *RS* κωδίκων. Επειδή η πολυπλοκότητα αντιστροφής πίνακα αυξάνεται με την τρίτη δύναμη του $t_d$ (ικανότητα διόρθωσης λαθών), η μέθοδος αυτή χρησιμοποιείται μόνο στις εφαρμογές όπου $t_d$ μικρό. Αντίθετα αν ο αριθμός των λαθών είναι μεγάλος τότε η επίλυση είναι πιο περίπλοκη και ο αλγόριθμος αυτός είναι μη αποτελεσματικός.

Σύμφωνα με τον άμεσο τρόπο αποκωδικοποίησης θα πρέπει να υπολογίσουμε τον αντίστροφο του $\mathrm{M}_\nu$ *Toeplitz* πίνακα που χρησιμοποιείται κατά την επίλυση

Του συστήματος ως προς τις τιμές $\{\Lambda_\nu, \Lambda_{\nu-1}, ..., \Lambda_1\}$. Το πρόβλημα που αντιμετωπίζουμε είναι η άγνωστη τιμή των πραγματικών λαθών που προέκυψαν.

Υποθέτουμε λοιπόν πως ο αριθμός αυτός των λαθών είναι $\nu$ και ακολουθούμε τα παρακάτω βήματα:

1. Υπολογίζουμε τις τιμές των συνδρόμων $S_i$ $1 \le i \le 2t$, θεωρούμε $\nu_{\max} = t_d$ και ελέγχουμε για $i = \nu_{\max} = t_d$ την τιμή της ορίζουσας $\Delta_i$

$$\Delta_i = \det \begin{pmatrix} S_1 & S_2 & \cdots & S_i \\ S_2 & S_3 & \cdots & S_{i+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_i & S_{i+1} & \cdots & S_{2i-1} \end{pmatrix}$$

2. Εάν $\Delta_i = 0$ τότε μικρότερος αριθμός λαθών προέκυψε κατά την μετάδοση του μηνύματος και ελαχιστοποιούμε την τιμή του $i$ μέχρι $i = 1$

3. Εάν $\Delta_i \neq 0$ τότε υπολογίζουμε τον αντίστροφο του πίνακα $\mathrm{M}_\nu$ θέτοντας $\nu = i$.

4. Εάν $\Delta_i = 0$ για $1 \le i \le t_d$ τότε η αποκωδικοποίηση δεν είναι επιτυχής και έχει ανιχνευτεί κάποιο λάθος που δεν είναι δυνατή η ορθή του διόρθωση ώστε να παραχθεί το πραγματικά απεσταλμένο μήνυμα.

**Παράδειγμα 3.8**: Υποθέτουμε κώδικα $RS(7,3,5)$ όμοιο με αυτό του παραδείγματος 1.6 και 1.7 και τον επιλύουμε με τον άμεσο τρόπο λύσης.

$t = (n-k)/2 = 2$

Έστω $i = \nu_{\max} = t_d = 2$,

$$\Delta_2 = \begin{vmatrix} a^6 & a^5 \\ a^5 & a \end{vmatrix} = a^7 + a^{10} = 1 + a^3 = a \neq 0$$

$$\begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix}^{-1} \begin{bmatrix} S_3 \\ S_4 \end{bmatrix} = \begin{bmatrix} a^6 & a^5 \\ a^5 & a \end{bmatrix}^{-1} \begin{bmatrix} a \\ a \end{bmatrix} = \begin{bmatrix} a^6 \\ a \end{bmatrix}$$

$$\Rightarrow \begin{matrix} \Lambda_2 = a^6 \\ \Lambda_1 = a \end{matrix}$$

$$\Lambda(x) = 1 + ax + a^6 x^2 = (1 + a^2 x)(1 + a^4 x)$$

*3. Υπολογισμός των θέσεων λάθους (Chien Search)*

Όπως είναι φανερό και από τα παραπάνω παραδείγματα, το πολυώνυμο θέσεις $\Lambda(x)$ που προκύπτει από τον ΕΑ αλγόριθμο θα διαφέρει από τους *BMA* και *direct* αλγορίθμους κατά ένα σταθερό παράγοντα. Όμως τον ενδιαφέρον στην αποκωδικοποίηση παρουσιάζεται στις ρίζες του πολυωνύμου, όπου οι αντίστροφες τιμές των ριζών αποτελούν ένδειξη των θέσεων των λαθών, και είναι ίδιες και στις τρεις περιπτώσεις. Συγκεκριμένα $\Lambda(x) = (1 + a^2 x)(1 + a^4 x)$ και οι θέσεις των λαθών είναι $j_1 = 2$ και $j_2 = 4$.

Σε μια πιο πολύπλοκη μορφή πολυωνύμου $\Lambda(x)$ γίνεται χρήση μιας απλής διαδικασίας δοκιμών *(Chien Search)*. Εξετάζουμε όλα τα στοιχεία του πεδίου του κώδικα $gf(q^m)$, εάν αποτελούν ρίζες του πολυωνύμου. Υποθέτουμε πως για $\nu = t = 3$ προκύπτει:

$$\Lambda(x) = \Lambda_0 + \Lambda_1 x + \Lambda_2 x^2 + \Lambda_3 x^3$$

Υπολογίζοντας την τιμή του πολυωνύμου για κάθε μη μηδενικό στοιχείο του πεδίου ($x = \{1, a, a^2, ..., a^{q^m - 2}\}$) και στη συνέχεια διαμορφώνουμε το πολυώνυμο λάθους. Για παράδειγμα, εάν οι ρίζες του $\Lambda(x)$ είναι $x = a^3$ και $x = a^4$ τότε το πολυώνυμο λάθους (εκτιμώμενου) προκύπτει:

$$\hat{e}(x) = e_{j1} x^{j1} + e_{j2} x^{j2}$$
$$\hat{e}(x) = e_{j1} x^3 + e_{j2} x^4$$

Χρησιμοποιώντας την εξίσωση (3-33) και επιλέγοντας έναν από τους παραπάνω τρόπους επίλυσης, επιλύουμε ως προς $e_{ji}$ για τον υπολογισμό των συντελεστών του πολυωνύμου λάθους $\hat{e}(x)$. Στη συνέχεια, από την εξίσωση:

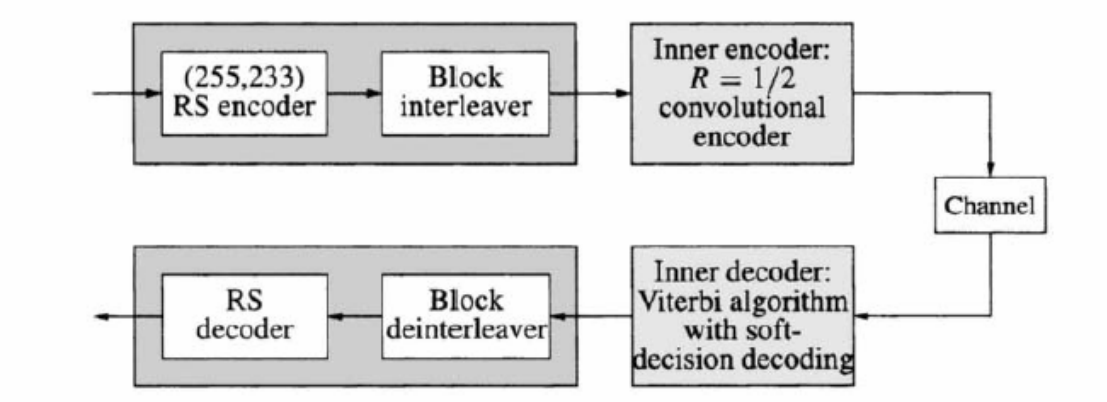
$$\hat{u}(x) = r(x) + \hat{e}(x)$$

$$\hat{u}(x) = u(x) + e(x) + \hat{e}(x)$$

προκύπτει το αποκωδικοποιημένο μήνυμα, που ανάλογα με το εκτιμώμενο λάθος παίρνει διάφορες τιμές. Σωστή αποκωδικοποίηση έχουμε στην περίπτωση που το εκτιμώμενο λάθος ισούται με το πραγματικό λάθος και το αποκωδικοποιημένο μήνυμα ισούται με το αρχικό (πριν την κωδικοποίηση) μήνυμα. Όμως σε πραγματικές συνθήκες εφαρμογής του κώδικα, ανάλογα με τις ιδιότητες του κώδικα και συγκεκριμένα με την ικανότητα διόρθωσης λαθών, προκύπτει διαφορετικό μήνυμα στην έξοδο του αποκωδικοποιητή. Η σύγκριση των bits των δύο μηνυμάτων παρέχει το ποσοστό λάθους (Bit Error Rate) ή την απόδοση του αποκωδικοποιητή.

## 3.4. Concatenated RS με εφαρμογή του Αλγορίθμου Viterbi

Οι Concatenated κώδικες είχαν προταθεί από τον Forney ως μέσο για την απόκτηση μεγάλων κωδίκων (όπως απαιτούσε το θεώρημα Shannonγια την επίτευξη βέλτιστης ικανότητας διόρθωσης) με μέτρια πολυπλοκότητα [3]. Το βασικό σύστημα concatenation κωδικοποίησης παρουσιάζεται στο παρακάτω σχήμα. Ο εσωτερικός κώδικας είναι συνήθως ένας δυαδικός κώδικας. Ο εξωτερικός κώδικας τυπικά είναι $(n_2, k_2)$ Reed-Solomon κώδικα στο πεδίο $GF(2^k)$. Κατά την κωδικοποίηση, ο εξωτερικός κώδικας λαμβάνει $k * k_2$ bits ως $k_2$ σύμβολα των $k$ bits και τα κωδικοποιεί σαν Reed-Solomon κωδικοποιημένες λέξεις $(c_0, c_1, ..., c_{n_2})$. Αυτά τα σύμβολα στη συνέχεια κωδικοποιούνται από το εσωτερικό κωδικοποιητή ως δυαδική ακολουθία όπου αυτή η δυαδική πλέον μορφή θα μεταδοθεί μέσω του καναλιού.




**Σχήμα 3.11** Σύστημα Συναλυσώμενης Κωδικοποίησης

Ο εσωτερικός κώδικας είναι συνήθως ένας συνελικτικού κώδικα. Ο σκοπός του εσωτερικού κώδικα είναι να βελτιώσει την ποιότητα του μηνύματος εξόδου έτσι ώστε ο RS κώδικας να μπορεί να χρησιμοποιηθεί πολύ αποτελεσματικά. Όταν ο Viterbi αποκωδικοποιητής (στο εσωτερικό) παράγει σφάλμα αποκωδικοποίησης, συνήθως

περιλαμβάνει μερικά διαδοχικά στάδια της αποκωδικοποίησης trellis, γεγονός που οδηγεί σε μια σύντομη έκρηξη σφαλμάτων. Η εξάπλωσής των λάθος bit τα οποία τείνουν να έχουν παραχθεί από το κέντρο της αποκωδικοποίησης αντιμετωπίζονται από τον RS αποκωδικοποιητή βασιζόμενος στην εγγενή ικανότητα του, διόρθωση εκρήξεων σφαλμάτων.

Προκειμένου να προβλεφθεί η πιθανότητα εκρηκτικής ακολουθίας λαθών της αποκωδικοποιημένης πληροφορίας να υπερβαίνει τα 16 x 8 = 128 bits, ένα σύμβολο interleaver τοποθετείται μεταξύ της RS κωδικοποιητή και του συνελικτικού κωδικοποιητή. Επειδή εισάγουμε ένα μόνο σύμβολο interleaver, τα bits λάθους τα οποία καταλαμβάνουν ένα μόνο byte εξακολουθούν να είναι συγκεντρωμένα μαζί. Αλλά πολλές εκρήξεις από bytes λαθών τυχαιοποιήθηκαν. Block interleavers μήκους από 2 έως 8 Reed-Solomon κωδικοποιημένες λέξεις έχουν εφαρμοστεί. Με τις μελέτες προσομοίωσης, αποδεικνύεται ότι για την επίτευξη ενός ποσοστό λάθους bit (BER) $10^{-5}$ με interleavers μεγέθους 2,4, και 8, απαιτείται ο όρος $E_b / N_0$ να παίρνει αντίστοιχες τιμές 2,6 dB, 2,45 dB, και 2,35 dB. Χωρίς την BPSK κωδικοποίηση, θα απαιτούνταν 9,6 dB. Τέλος χρησιμοποιώντας μόνο συνελικτικό κώδικα με ρυθμό κωδικοποίησης 1/2 θα απαιτούν 5,1 dB, και συνεπώς καταλήγουμε στο ότι οι Concatenated κώδικες παρέχουν περίπου 2,5 dB του κέρδους σε σύγκριση με το συνελικτικό κώδικα μόνο.

## 3.5. Εφαρμογές Κωδίκων RS σε Σύγχρονα και Μελλοντικά Συστήματα Ασυρμάτων Κινητών Επικοινωνιών

Οι Reed-Solomon κώδικες, έχουν ένα ευρύ φάσμα εφαρμογών στους τομείς των ψηφιακών επικοινωνιών και αποθήκευσης. Χρησιμοποιούνται στην διόρθωση λαθών, σε πολλά συστήματα συμπεριλαμβανομένων:

- Συσκευές αποθήκευσης (συμπεριλαμβανομένων ταινία, Compact Disk, DVD, barcodes, κλπ)
- Ασύρματες ή κινητές επικοινωνίες (συμπεριλαμβανομένων κινητά τηλέφωνα, μικροκυματικές ζεύξεις, κ.λπ.)
- Δορυφορικές επικοινωνίες
- Η ψηφιακή τηλεόραση / DVB
- Υψηλής ταχύτητας μόντεμ όπως ADSL, xDSL, κλπ.

Ο Reed-Solomon αποκωδικοποιητής επεξεργάζεται κάθε μπλοκ και προσπαθεί να διορθώσει τα λάθη και να ανακτήσει τα αρχικά δεδομένα. Ο αριθμός και το είδος των λαθών που μπορούν να διορθωθούν εξαρτάται από τα χαρακτηριστικά του Reed-Solomon κώδικα.

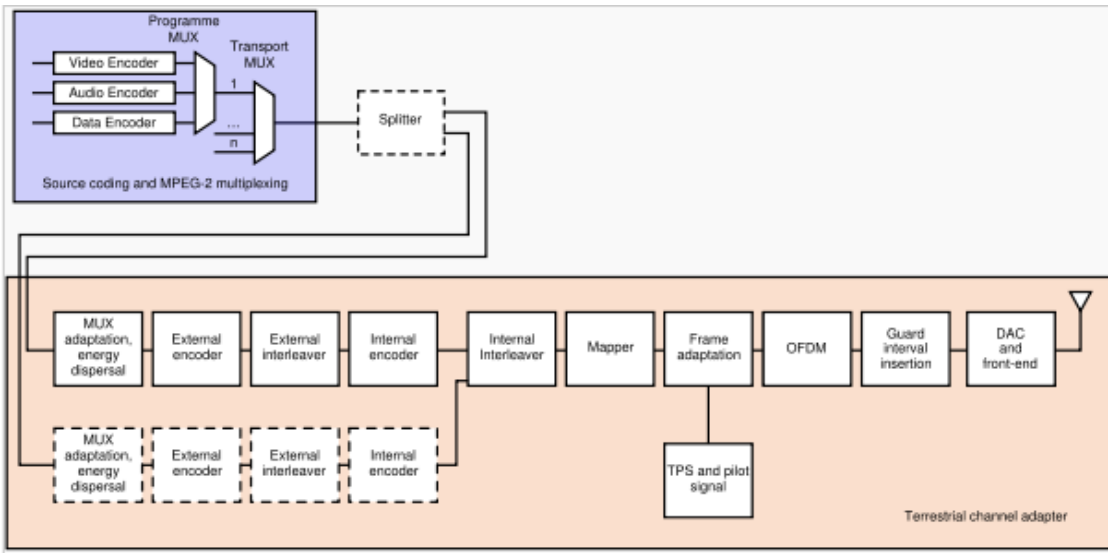
### 3.5.1. DVB

Το DVB (Digital Video Broadcast) είναι ένα σύνολο προδιαγραφών για την ψηφιακή μετάδοση εικόνας και ήχου, καθώς επίσης και μετάδοση δεδομένων. Οι DVB προδιαγραφές διατηρούνται μέσα από το DVB Project, που είναι μια κοινοπραξία υπό την ηγεσία πάνω από 260 ραδιοτηλεοπτικών φορέων, κατασκευαστών, φορείς εκμετάλλευσης δικτύων, λογισμικού, ρυθμιστικών φορέων και άλλων σε περισσότερες από 35 χώρες. Οι προδιαγραφές αυτές καθορίζουν το φυσικό στρώμα στρώμα και το στρώμα συνδέσμου δεδομένων του συστήματος διανομής. Συσκευές αλληλεπιδρούν με το φυσικό στρώμα μέσω μιας σύγχρονης παράλληλης διεπαφή (SPI), ή σύγχρονης σειριακής διεπαφή (SSI), ή ασύγχρονης σειριακής διεπαφή (ASI). Όλα τα δεδομένα μεταδίδονται σε MPEG-2 ρεύματα μεταφοράς με κάποιους επιπλέον περιορισμούς (DVB-MPEG). Οι DVB προδιαγραφές έχουν εφαρμοστεί στα παρακάτω πεδία:

- DVB-S (Satellite) ETS 300 421 (Συστήματα ψηφιακής δορυφορικής μετάδοσης)
- DVB-T (Terrestrial) ETS 300 744 (Συστήματα Ψηφιακής επίγειας μετάδοσης)
- Διεπαφές σε Plesiochronous Digital Hierarchy (PDH) networks (Prets 300 813).
- Διεπαφές σε Synchronous Digital Hierarchy (SDH) networks (Prets 300 814).
- Διεπαφές σε Asynchronous Transfer Mode (ATM) δίκτυα (Prets 300 815).
- Διεπαφές για CATV/SMATV Headends και παρόμοιο επαγγελματικό εξοπλισμό(EN50083-9)

Ένα παράδειγμα προτύπου DVB μετάδοσης που χρησιμοποιεί τον Reed-Solomon κώδικα για βελτιωμένη διόρθωση λαθών είναι το DVB-T. Το DVB-T ή Σύστημα Επίγειας Ψηφιακής Μετάδοσης είναι το πιο ευρέως χρησιμοποιούμενο πρότυπο ψηφιακής τηλεόρασης σε όλο τον πλανήτη για επίγειας τηλεοπτικές μεταδόσεις. Παρέχει πολλές εγκαταστάσεις και επιτρέπει μια πολύ πιο αποτελεσματική χρήση των διαθέσιμων συχνοτήτων ραδιοφωνικού φάσματος σε σχέση με τις παλαιότερες αναλογικές μεταδόσεις. Το DVB-T εκδόθηκε για πρώτη φορά το 1997 και από τότε έχει γίνει το πιο ευρέως χρησιμοποιούμενο πρότυπο στην την ψηφιακή μετάδοση παγκοσμίως. Μέχρι το 2008, ήταν το πρότυπο που εκδόθηκε σε περισσότερες από 35 χώρες και πάνω από 60 εκατομμύρια δέκτες έχουν αναπτυχθεί και το χρησιμοποιούν.

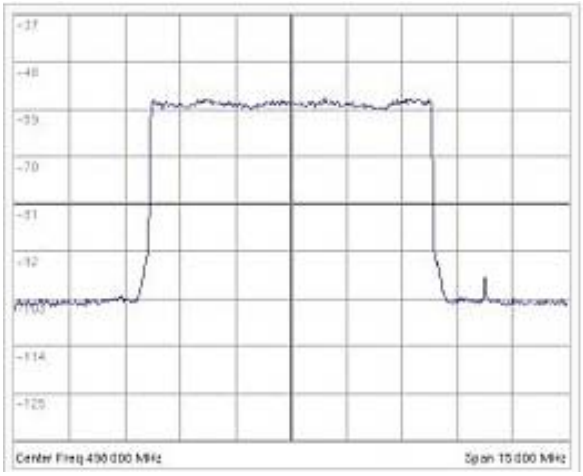
Όσον αφορά την τεχνική περιγραφή DVB-T μετάδοσης, το Σχήμα 3.12 υποδεικνύει αναλυτικά τα μπλοκ όπου διέρχεται το σήμα κατά τη διέλευση του από τον πομπό στον δέκτη. Οι διεργασίες από τις οποίες διέρχεται το σήμα κατά τη διέλευση από το πομπό στο δέκτη αναλύονται παρακάτω σύμφωνα με την αναφορά [11].




**Σχήμα 3.12** Αρχιτεκτονική Συστήματος Ψηφιακής επίγειας μετάδοσης

- **Πηγή κωδικοποίησης MPEG-2 και πολυπλεξίας (MUX):** Πληροφορία όπως συμπιεσμένο βίντεο, συμπιεσμένος ήχος και ροές δεδομένων πολυπλέκονται σε σε PSS (προγραμματισμένες ροές). Ένα ή περισσότερα PSS ενώνονται σε MPEG-2 TS (MPEG-2 Transport Stream). Αυτό είναι το βασικό ψηφιακό ρεύμα το οποίο διαβιβάζεται και παραλαμβάνεται από Set Top Boxes (STB).
- **Splitter:** δύο διαφορετικά TSS μπορούν να μεταδοθούν ταυτόχρονα, χρησιμοποιώντας μια τεχνική που ονομάζεται *Hierarchical Transmission*. Είναι δυνατόν να χρησιμοποιείται για τη διαβίβαση, για παράδειγμα, μιας τυπικής ευκρίνειας SDTV σήμα και μιας υψηλής ευκρίνειας HDTV σήμα, ο ίδιος μεταφορέας. Γενικά, το SDTV σήμα είναι πιο ισχυρό από το HDTV. Στο δέκτη, ανάλογα με την ποιότητα του σήματος που λαμβάνεται, το STB μπορεί να είναι σε θέση να αποκωδικοποιεί HDTV ρεύματα ή, αν η ισχύς του σήματος είναι μικρή, μπορεί να στραφεί σε SDTV σήματα.
- **MUX προσαρμογή και διασπορά ενέργειας:** το MPEG-2 TS χαρακτηρίζεται ως μια ακολουθία πακέτων δεδομένων, σταθερού μήκους (188 bytes). Με μια τεχνική που ονομάζεται διασπορά ενέργειας, η ακολουθία byte αποσυσχετίζεται.
- **Εξωτερικός κωδικοποιητής:** Το πρώτο επίπεδο διόρθωσης σφαλμάτων εφαρμόζεται στα δεδομένα που διαβιβάζονται. Χρησιμοποιώντας ένα μη δυαδικό block ***Reed-Solomon RS (204, 188)*** κώδικα επιτρέπεται η διόρθωση ενός ανώτατου ορίου 8 bytes για κάθε 188 bytes πακέτων.
- **Εξωτερικός interleaver:** ***συνελικτικός interleaving*** χρησιμοποιείται για την αναδιαμόρφωση της σειράς δεδομένων που μεταδίδονται, με τέτοιο τρόπο ώστε να διασπόνται μεγάλες ακολουθίες σφαλμάτων.
- **Εσωτερική κωδικοποιητή:** ένα δεύτερο επίπεδο διόρθωσης λάθους δίνεται από ένα συνελικτικό κώδικα, ο οποίο συχνά αποδίδεται ως FEC (Forward διόρθωση σφαλμάτων) κώδικας. Υπάρχουν πέντε έγκυροι ρυθμοί κωδικοποίησης: 1 / 2, 2 / 3, 3 / 4, 5 / 6, και 7 / 8.

- **Εσωτερικός interleaver:** Ακολουθία δεδομένων υπόκειται αναδιάταξη και πάλι, με στόχο να μειωθεί η επιρροή των εκρήξεων (μεγάλης ακολουθίας) λαθών. Αυτή τη φορά χρησιμοποιείται μπλοκ interleaving τεχνική με σύστημα ψευδο-τυχαίας εκχώρησης (δύο ξεχωριστές interleaving διαδικασίες, η μία εκ των οποίων εφαρμόζεται σε bits και η άλλη σε ομάδες από bits).
- **Mapper:** Ψηφιακή ακολουθία bit αντιστοιχίζεται σε μια σειρά συμβόλων διαμόρφωσης βασικής ζώνης (base band modulated). Υπάρχουν τρία έγκυρα συστήματα διαμόρφωσης: QPSK, 16-QAM, 64-QAM.
- **Προσαρμογή Πλαισίου:** Τα σύμβολα ομαδοποιούνται σε block σταθερού μήκους (1512, 3024, ή 6.048 σύμβολα ανά block). Δημιουργούνται πλαίσια μήκους 68 blocks, και ένα υπερπλαίσιο αποτελείται από 4 πλαίσια.
- **Πιλοτικά και TPS μηνύματα:** Προκειμένου να απλοποιηθεί η διαδικασία λήψης του μεταδιδόμενου σήματος στο επίγειο ραδιοφωνικό σταθμό, επιπλέον σήματα προστίθενται σε κάθε block. Πιλοτικά σήματα χρησιμοποιούνται κατά τη διάρκεια του συγχρονισμού και εξισορρόπησης φάση, ενώ TPS σήματα (Transmission Parameters Signalling) αποστέλλουν παραμέτρους των σημάτων που μεταδίδονται ώστε να προσδιορισθούν σαφώς τα μεταδιδόμενα πλαίσια. Ο δέκτης πρέπει να γνωρίζει την απαραίτητη πληροφορία για την αποκωδικοποίηση κάθε σήματος και τα TPS δεδομένα χρησιμοποιούνται μόνο σε ειδικές περιπτώσεις, όπως αλλαγές στις παραμέτρους, επανάληψη συγχρονισμού (resynchronization), κλπ.
- **OFDM Modulation:** η σειρά των μπλοκ διαμορφώνεται σύμφωνα με την OFDM τεχνική με τη χρήση 2048, 4096, 8192 φορέων (2k, 4k, 8k mode, αντίστοιχα). Η αύξηση του αριθμού των φορέων δεν επηρεάζει το ωφέλιμο ρυθμό bits, το οποίο παραμένει σταθερό.



**Σχήμα 3.13** Φασματική Περιγραφή Συστήματος Ψηφιακής Μετάδοσης (8k mode)

- **Προσθήκη διαστήματος ασφαλείας:** Προκειμένου την μείωση της πολυπλοκότητας στον δέκτη, κάθε OFDM block επεκτείνεται, κυκλικό πρόθεμα ως αντιγραφή μπροστά από το τέλος κάθε block. Το πλάτος των εν λόγω διαστήματος ασφαλείας μπορεί να είναι 1 / 32, 1 / 16, 1 / 8 ή 1 / 4 ποσοστό του

αρχικού μήκους μπλοκ. Το κυκλικό πρόθεμα είναι αναγκαίο στην λειτουργία δικτύων σε μια μόνο συχνότητα, όπου μπορεί να υφίστανται συνεχείς παρεμβολές που προέρχονται από πολλές τοποθεσίες που μεταδίδουν πρόγραμμα με την ίδια συχνότητα.

- **DAC και front-end:** Το ψηφιακό σήμα μετατρέπεται σε αναλογικό σήμα, με ψηφιακό σε αναλογικό μετατροπέα (DAC), και στη συνέχεια διαμορφώνεται σε ραδιοφωνική συχνότητα (VHF, UHF) από το RF front-end. Το εύρος ζώνης που καταλαμβάνεται είναι σχεδιασμένο να παρέχει σε κάθε συνιστώσα DVB-T σήματος 5, 6, 7 ή 8 MHz εύρος καναλιού. Ο ρυθμός βασικής ζώνης (base band sample rate ) που παρέχεται στην είσοδο του DAC εξαρτάται από το εύρος ζώνης του καναλιού: Προκύπτει: $f_s = \frac{8}{7}B$ δείγματα / s, όπου B είναι το εύρος ζώνης καναλιού που εκφράζεται σε Hz.

### 3.5.2. WiMax

Βασισμένη στο πρότυπο IEEE 802,16, η τεχνολογία WiMAX (Worldwide Interoperability Microwave Access) έρχεται να γεφυρώσει το κενό ανάμεσα στις ασύρματες συνδέσεις και τις μεγάλες ταχύτητες των ενσύμαρτων συνδέσεων όπως παρουσιάζεται στην αναφορά [12]. Συχνά αποκαλείται Wireless Broadband γιατί υπόσχεται ταχύτητες αντίστοιχες με ADSL, Cable και T1 χωρίς καλώδια, σε μακρινές αποστάσεις.

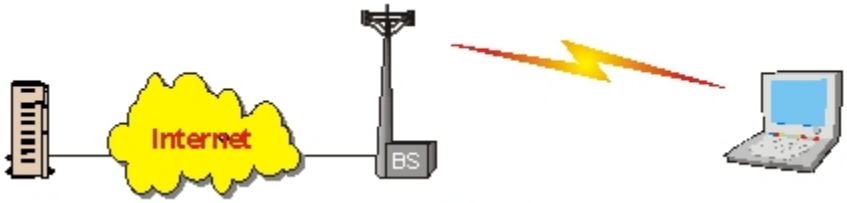
Μέχρι την τελική κάλυψη πόλεων, απομακρυσμένων τοποθεσιών ή περιοχών με γεωγραφικές ιδιαιτερότητες, το WiMAX υπόσχεται να καλύψει τα κενά που ήδη υπάρχουν στις ενσύρματες συνδέσεις αλλά και στα δίκτυα κινητής τηλεφωνίας. Χαρακτηριστικό της τεχνολογίας είναι ότι μόνο ένας σταθμός (κεραία) WiMAX μπορεί να προσφέρει ταυτόχρονα συνδέσεις σε αρκετούς συνδρομητές ή επιχειρήσεις με ταχύτητες T1 ενώ παράλληλα να εξυπηρετεί πλήθος συνδέσεων με ταχύτητες ADSL ή Cable. Το πρακτικό αποτέλεσμα της δυνατότητας αυτής, είναι η διάθεση πολλών ασύρματων συνδέσεων υψηλών ταχυτήτων με μειωμένο κόστος εγκατάστασης και συντήρησης σε σύγκριση με τις υπάρχουσες τεχνολογίες.

Η δοκιμαστική εγκατάσταση σταθμών WiMAX έχει ξεκινήσει ήδη και αναμένεται η επέκταση των δοκιμών και της διάδοση της τεχνολογίας. Οι αισιόδοξοι κάνουν αναφορές για το 2006 και πέρα αλλά φαίνεται πως θα χρειαστούν τουλάχιστον 2

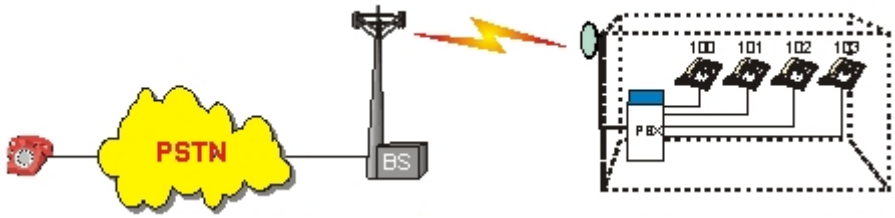
χρόνια μέχρι την εκτεταμένη διάδοση του προτύπου και το άνοιγμα στην αγορά. Τα μελλοντικά laptop (Intel Centrino WiMAX) και πιθανώς τα κινητά τηλέφωνα και PDA θα υποστηρίζουν την τεχνολογία WiMAX βασισμένα σε συμβατό με το πρότυπο IEEE 802.16e hardware, όπως ακριβώς συμβαίνει τώρα με το WiFi και το πρότυπο 802.11.


**Σχήμα 3.14** Βασικοί Σταθμοί WiMAX Συστήματος

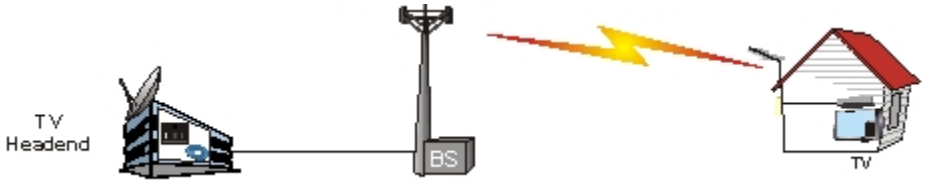
Τα παρακάτω σχήματα παρουσιάζουν ορισμένες από τις εφαρμογές όπου τα WiMAX συστήματα μπορούν να χρησιμοποιηθούν. Όπως παρουσιάζεται, τα WiMAX συστήματα μπορούν να προσφέρουν ασύρματη ευρυζωνική πρόσβαση στο Internet, τηλεφωνικές υπηρεσίες πρόσβασης, ή τηλεοπτική υπηρεσία πρόσβασης και υπηρεσίες κινητής τηλεφωνίας



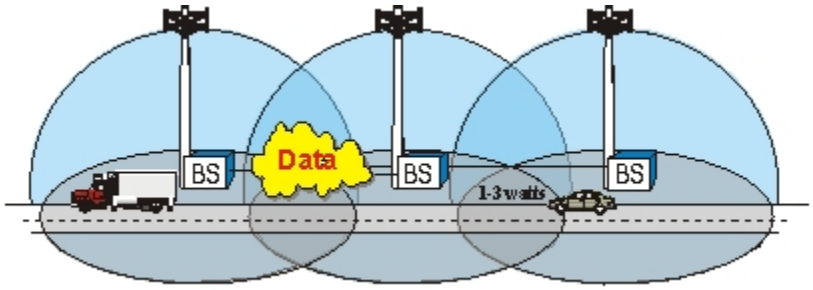

**Σχήμα 3.15** Ασύρματη Μετάδοση




**Σχήμα 3.16** Τηλεφωνική Επικοινωνία




**Σχήμα 3.17** Ψηφιακή Τηλεόραση




**Σχήμα 3.18** Μετάδοση Δεδομένων Κινητής Τηλεπικοινωνίας

Όσον αφορά το φυσικό (PHY) στρώμα μεταφοράς, αυτό καταλαμβάνει εύρος συχνοτήτων 10 - 66 GHz και βασίζεται στη διαμόρφωση ενός μόνο φορέα με προσαρμογή εκρηκτικού προσθετικού θορύβου. Αυτό σημαίνει ότι οι διεργασίες διαμόρφωσης και κωδικοποίησης μπορούν να εφαρμοστούν ξεχωριστά για κάθε μία συσκευή με ρυθμό πλαίσιο ανά πλαίσιο. Οι προδιαγραφές επίσης καθορίζουν τόσο TDD όσο και FDD λειτουργικά συστήματα.

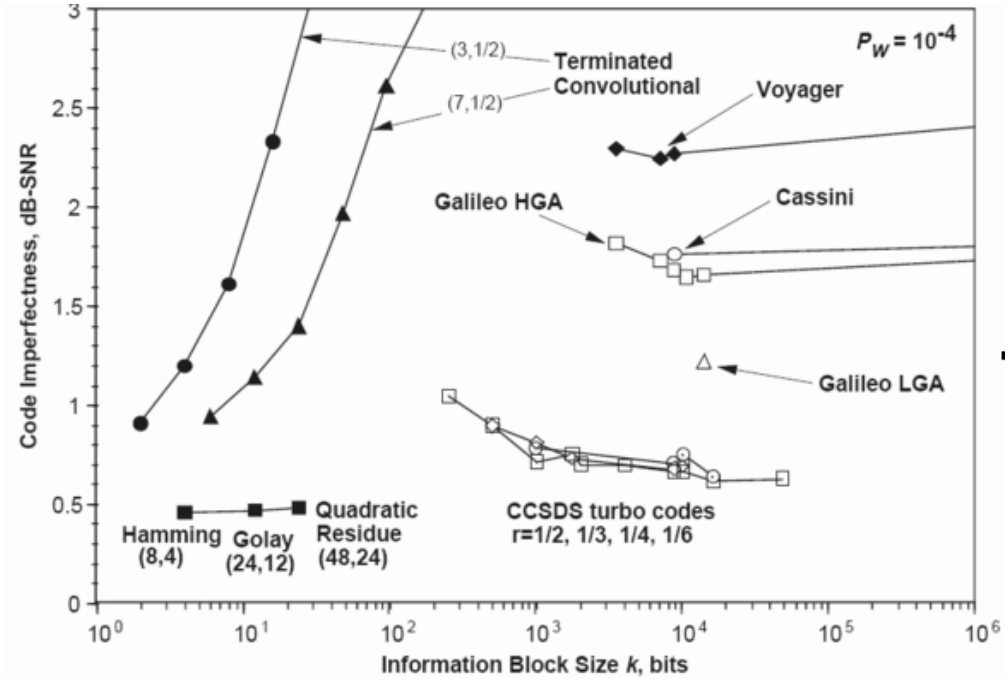
Στις Ηνωμένες Πολιτείες οι προδιαγραφές προσδιορίζουν κανάλια 20 - 25 MHz, ενώ τα κανάλια των 26 MHz έχουν οριστεί για Ευρωπαϊκούς φορείς. Οι Forward Error Correction (FEC) διεργασίες βασίζονται στο ***Reed-Solomon GF (256)*** κώδικα με αυτή τη μεταβλητή ως το μήκος του μπλοκ. Ο κώδικας αυτός συνδυάζεται με συνελικτικό κώδικα για να εξασφαλιστεί η διαβίβαση του περιεχομένου κρίσιμων δεδομένων, όπως οι αρχικές παράμετροι του πλαισίου ελέγχου πρόσβασης. Διεργασίες FEC συνδυάζονται με Quadrature Phase Sift Keying (QPSK) 16 -state τετραγωνικής διαμόρφωσης πλάτους (16QAM) και 64 -state (64QAM) για τον έλεγχο της αποτελεσματικότητας της μετάδοσης. Εάν το τελικό FEC Block δεν έχει συμπληρωθεί κατά το χρονικό όριο της

μεταφοράς, τότε αυτό μπορεί να μειωθεί από το σταθμό βάσης τόσο στην κατεύθυνση μετάδοσης προς και αντίθετα προς τον δορυφόρο. Εάν συμβεί αυτή η επέμβαση, άμεσα γνωστοποιείται τόσο στο χάρτη καταγραφής κατεύθυνσης μετάδοσης προς (UL-Map) και αντίθετα προς (DL-MAP) το δορυφόρο.

Το φυσικό στρώμα 802,16 καθορίζει το πλαίσιο μετάδοσης διάρκειας 0,5 1 ή 2 ms. Χωρίζεται σε timeslots για τις κατανομές και τις αναγνωρίσεις του εύρος ζώνης στο φυσικό (PHY) στρώμα μεταφοράς. Ένα timeslot ορίζεται από τέσσερα σύμβολα QAM. Στην περίπτωση TDD συστημάτων, υποπλαίσια μετάδοσης προς το δορυφόρο ακολουθούν τα υποπλαίσια μετάδοσης κατεύθυνσης αντίθετης προς τον δορυφόρο για την ίδια συχνότητα. Στα FDD συστήματα, τα υποπλαίσια ανεξαρτήτως κατεύθυνσης μετάδοσης συμπίπτουν χρονικά αλλά χρησιμοποιούν διαφορετικές συχνότητες μεταφοράς.

### 3.5.3. Satellite Transmission

Μια επίσης σημαντική εφαρμογή της Reed-Solomon κωδικοποίησης ήταν η κωδικοποίηση ψηφιακών φωτογραφιών που στέλνονται πίσω από το διαστημικό μηχάνημα Voyager. Το Voyager εισήγαγε Reed-Solomon κωδικοποίηση σε συνδυασμό με ML συνελικτικοί κώδικες, μια πρακτική που έγινε πολύ διαδεδομένη στο βαθύ διάστημα και στις δορυφορικές επικοινωνίες (π.χ. άμεσες ψηφιακές εκπομπές).




**Σχήμα 3.19** Κώδικες που χρησιμοποιούνται στη δορυφορική επικοινωνία

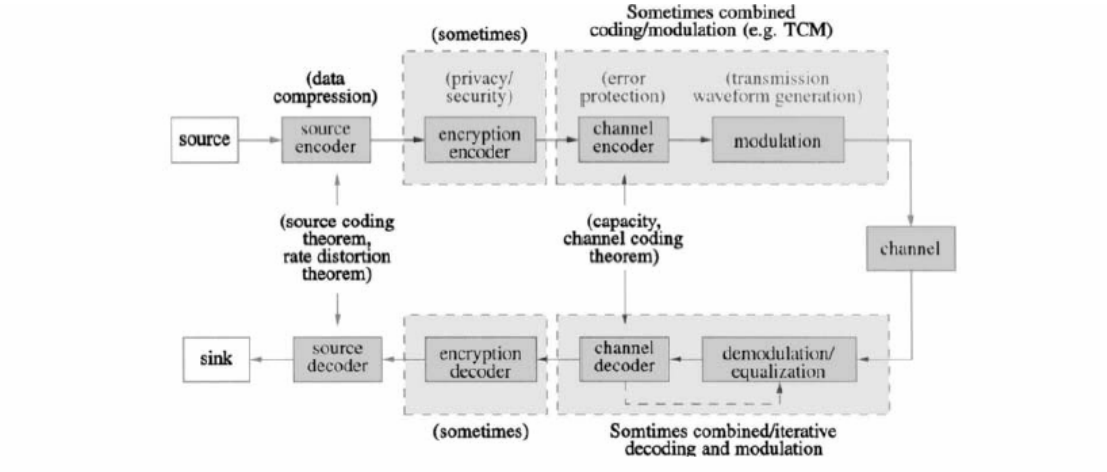
Αποκωδικοποιητές Viterbi τείνουν να εξάγουν μικρών εκρήξεων λάθη. Η διόρθωση αυτών των λαθών επιτυγχάνεται καλύτερα από περικομμένους ή απλοποιημένους Reed-Solomon κώδικες. Σύγχρονες εκδόσεις concatenated Reed-Solomon/Viterbi-decoded κωδικοποίησης χρησιμοποιούνται για στις *Mars Pathfinder*, *Galileo*, *Mars Exploration Rover* και *Cassini* αποστολές, όπου εφαρμόζονται εντός περίπου 1-1,5 dB του τελικό ορίου που επιβάλλει η θεωρία Shannon. Αυτοί οι concatenated κώδικες τώρα τείνουν να αντικατασταθούν από πιο ισχυρούς κωδικούς turbo όπου τα μεταδιδόμενα δεδομένα δεν χρειάζεται να αποκωδικοποιούνται αμέσως.

# 4. Υλοποίηση του Συστήματος με χρήση του Προγραμματιστικού Περιβάλλοντος Matlab

Υπάρχουν πολλοί λόγοι για την επιλογή του ***MATLAB*** ως προγραμματιστικό περιβάλλον. Αρχικά, το MATLAB χρησιμοποιείται ευρέως στην κοινωνία των μηχανικών. Επιπλέον το MATLAB συνδυάζει εξαιρετική υπολογιστική με άριστες και εύκολες στη χρήση γραφικές δυνατότητες. Το MATLAB περιέχει πλούσια βιβλιοθήκη των προγραμματιστικών λειτουργιών (m-files) για την παραγωγή, ανάλυση, επεξεργασία και παρουσίαση σημάτων. Επιπρόσθετες βιβλιοθήκες (toolboxes) επιτρέπουν τη βασική MATLAB βιβλιοθήκη να συμπληρωθεί με m-αρχεία σημαντικά για συγκεκριμένες εφαρμοσμένες περιοχές. Όσο για τον MATLAB κώδικα, είναι πολύ συνοπτικός, με αποτέλεσμα να κάνει εφικτή την επεξεργασία πολύπλοκων σημάτων (προσομοίωση σημάτων) με χρήση αλγορίθμων σε πολύ λίγες γραμμές. Για την προσομοίωση των επιμέρους διεργασιών κατά την διέλευση του σήματος από τον πομπό στον δέκτη, έχω αναπτύξει κώδικα MATLAB όπως αναλύεται στις παρακάτω υποενότητες.

## 4.1. Δομή Συστήματος Ψηφιακών Επικοινωνιών

Ένα ψηφιακό σύστημα επικοινωνίας ενσωματώνει λειτουργίες για την εκτέλεση πραγματικών δραστηριοτήτων που αφορούν μετάδοση πληροφορίας. Το παρακάτω Σχήμα απεικονίζει ένα γενικό πλαίσιο ενός ενιαίου ψηφιακού επικοινωνιακού συνδέσμου. Σε αυτόν το σύνδεσμο, ψηφιακά δεδομένα από μια πηγή κωδικοποιούνται και διαμορφώνονται (και ενδεχομένως κρυπτογραφούνται) για μετάδοση μέσω ενός καναλιού επικοινωνίας. Στην άλλη άκρη του καναλιού, τα δεδομένα αποδιαμορφώνονται , αποκωδικοποιούνται (και ενδεχομένως αποκρυπτογραφούνται), και αποστέλλονται σε έναν προορισμό. Όλα τα στοιχεία του συνδέσμου έχουν μαθηματικές περιγραφές και θεωρήματα από την θεωρία της πληροφορίας που διέπουν τις εφαρμογές και επιδόσεις τους. Υπάρχουν πολλά είδη κωδίκων που έχουν εφαρμογή σε ένα σύστημα επικοινωνίας. Στην παρακάτω αναφορά επισημαίνονται οι επιδόσεις των κωδίκων διόρθωσης λάθους που ασχολούμαστε: Reed-Solomon καθώς και ο συνδυασμός αυτού με τον Viterbi που αποτελεί αλγόριθμο μέγιστης απόδοσης των συνελικτικών κωδίκων.




**Σχήμα 4.1** Δομή Συστήματος Ψηφιακών Επικοινωνιών

**Η πηγή** είναι τα στοιχεία που πρόκειται να μεταδοθούν, όπως ένα ηλεκτρονικό αρχείο, ένα βίντεο, ή μια τηλεφωνική συνομιλία. Για τους σκοπούς μας, η πληροφορία παρουσιάζεται σε ψηφιακή μορφή, ίσως ως αποτέλεσμα της μετατροπής από αναλογική σε ψηφιακή μορφή. Η πηγή (θεωρητικά πληροφορία) προβάλλεται ως ρεύματα τυχαίων αριθμών που διέπονται από ορισμένες κατανομές πιθανοτήτων. Κάθε πηγή δεδομένων έχει ένα μέτρο πληροφορίας που αντιπροσωπεύει, το οποίο μπορεί να ποσοτικοποιηθεί με ακρίβεια μέσω του ορισμού της εντροπίας.

**Ο κωδικοποιητής πηγής** εκτελεί συμπίεση δεδομένων με την άρση των πλεοναζόντων ψηφίων. Ο αριθμός των bits που χρησιμοποιούνται για την αποθήκευση της πληροφορίας από τη πηγή μπορεί να υπερβαίνει τον αριθμό των bits του πραγματικού περιεχομένου της πληροφορίας. Το πόσο μια συγκεκριμένη πηγή δεδομένων μπορεί να συμπιέζεται χωρίς καμία απώλεια πληροφοριών (lossless συμπίεση) θεωρητικά διέπεται από το θεώρημα κωδικοποίησης πηγής της θεωρίας της πληροφορίας, η οποία αναφέρει ότι μια πηγή πληροφοριών μπορεί να εκπροσωπείται χωρίς καμία απώλεια δεδομένων κατά τέτοιο τρόπο ώστε το ύψος της αποθήκευσης που απαιτείται (σε bits) να είναι ίσο με το ποσό του περιεχομένου της πληροφορίας - η εντροπία - ή σε bits Shannons. Για να επιτευχθεί αυτό το χαμηλότερο όριο, μπορεί να είναι απαραίτητο τα μεγάλα block δεδομένων να κωδικοποιούνται από κοινού.

**Ο encrypter (Κρυπτογράφηση)** αποκρύπτει ή ανακατανέμει τη πληροφορία έτσι ώστε τυχαίοι προορισμοί να μην είναι σε θέση να διακρίνουν το περιεχόμενο της πληροφορίας. Οι κώδικες που χρησιμοποιούνται για κρυπτογράφηση είναι γενικά διαφορετικοί από τους κωδικούς που χρησιμοποιούνται για τη διόρθωση σφαλμάτων. Η Κρυπτογράφηση είναι συχνά για ένα μη ειδήμονα συνώνυμο του όρου "κωδικοποίηση," αλλά παρατηρούμε πως υπάρχουν πολλά άλλα διαφορετικά είδη κωδίκων.

**Ο Κωδικοποιητής καναλιού** είναι το πρώτο βήμα προς τη διαδικασία διόρθωσης ή ανίχνευσης λάθους. Ο Κωδικοποιητής καναλιού προσθέτει πλεονάζουσα πληροφορία στο ρεύμα εισόδου των συμβόλων με τρόπο που πετυχαίνει λάθη που έχουν εισαχθεί στο κανάλι να διορθώνονται. Το βήμα αυτό σε συνδυασμό με τον αποκωδικοποιητή καναλιού καλύπτουν ολόκληρη σχεδόν την μελέτη που αφιερώνει η εργασία για τους

κώδικες διόρθωσης λάθους. Μπορεί να φαίνεται περίεργο, αρχικά άρση πλεοναζόντων με τον κωδικοποιητή πηγής, και στη συνέχεια προσθήκη πλεοναζόντων με το κωδικοποιητή καναλιού. Ωστόσο, τα πλεονάζοντα ψηφία στην πηγή συνήθως εξαρτώνται από μια προσπάθεια κατανόησης του περιεχομένου της πηγής, ενώ τα πλεονάζοντα από τον κωδικοποιητή καναλιού εισάγονται με ένα δομημένο τρόπο, ακριβώς να παρέχουν δυνατότητα ελέγχου σφάλματος. Μεταχειρίζοντας τα προβλήματα της συμπίεσης δεδομένων και διόρθωσης λαθών ξεχωριστά, αντί της επίλυσης από κοινού βέλτιστης κωδικοποίησης πηγής / διαύλου, είναι αυτό που προτείνεται (ιδιαίτερα για μεγάλου μεγέθους block). Το γεγονός αυτό είναι γνωστό ως θεώρημα διαχωρισμού πηγής / διαύλου στην θεωρία της πληροφορίας. Συχνά, ο κωδικοποιητής καναλιού λειτουργεί λαμβάνοντας ως είσοδο ένα μπλοκ $k$ συμβόλων και παράγοντας ως έξοδο ένα μπλοκ $n$ συμβόλων, με $n > k$. Ο ρυθμός κωδικοποίησης είναι:

$$R = \frac{k}{n} \quad , \; R < 1$$

Η είσοδος στον κωδικοποιητή καναλιού αναφέρεται ως σύμβολα (ή bits στην περίπτωση της δυαδικών κωδίκων) μηνύματος ή πληροφορίας.

**Ο Διαμορφωτής** μετατρέπει αλληλουχίες συμβόλων από τον κωδικοποιητή καναλιού σε κατάλληλα σήματα για μετάδοση πάνω από το κανάλι. Πολλά κανάλια απαιτούν τα σήματα να αποστέλλονται ως τάση συνεχούς χρόνου, ή ως ηλεκτρομαγνητική κυματομορφή σε μια καθορισμένη συχνότητα. Έτσι ο Διαμορφωτής παρέχει μορφή σήματος κατάλληλη-σύμφωνη με το κανάλι. Ορισμένοι Διαμορφωτές διαθέτουν μηχανισμούς για να εξασφαλιστεί ότι το μήνυμα καταλαμβάνει ένα ευρύ εύρος ζώνης. Η εξάπλωση αυτή του φάσματος μπορεί να χρησιμεύσει για να παρέχει πρόσβαση πολλαπλών χρηστών, μεγαλύτερη αντοχή στις παρεμβολές, χαμηλή πιθανότητα ανίχνευσης, καθώς και άλλα πλεονεκτήματα [3].

**Το κανάλι** είναι το μέσο πάνω από το οποίο μεταδίδονται οι πληροφορίες. Παραδείγματα καναλιών αποτελούν οι τηλεφωνικές γραμμές, καλώδια Internet, γραμμές οπτικών ινών, μικροκυματικά κανάλια, κανάλια υψηλών συχνοτήτων, κανάλια κινητού τηλεφώνου, κλπ. Αυτά είναι κανάλια στα οποία η πληροφορία μεταδίδεται μεταξύ δύο διακριτών θέσεων. Η πληροφορία μπορεί επίσης να μεταφέρεται μεταξύ δύο διαφορετικών χρονικών στιγμών. Για παράδειγμα, αποθηκεύοντας πληροφορία πάνω σε ένα δίσκο του υπολογιστή και ανακτώντας την σε μεταγενέστερο χρόνο.

Καθώς τα σήματα ταξιδεύουν μέσω ενός καναλιού υπόκεινται σε αλλοίωση. Για παράδειγμα, ένα μήνυμα μπορεί να υποστεί προσθήκη θορύβου, χρονική καθυστέρηση ή εξασθένηση οφειλόμενη στο φαινόμενο πολλαπλών διαδρομών. Επίσης υπόκειται σε ακούσιες παρεμβολές από άλλους σταθμούς, ή εκούσια συμφόρηση παρεμβολών. Αυτές οι πηγές διαφθοράς σε πολλές περιπτώσεις μπορεί να συμβαίνουν όλες ταυτόχρονα.

Για τους σκοπούς της ανάλυσης, οι δίαυλοι συχνά χαρακτηρίζονται από μαθηματικά μοντέλα, τα οποία είναι αρκετά ακριβή ώστε να αντιπροσωπεύουν τα χαρακτηριστικά πραγματικών καναλιών. Ένα μοντέλο καναλιού που θα ασχοληθούμε είναι το AWGN κανάλι (white Gaussian noise channel) που αποτελεί σταθμός για την

εξοικείωση με πιο σύνθετα κανάλια όπως είναι ο δίαυλος διαλείψεων (Fading Channel): Rayleigh/Ricean.

Τα κανάλια έχουν διαφορετικές ικανότητες όσον αφορά τη μεταφορά πληροφορίας. Για παράδειγμα, μια γραμμή οπτικών ινών είναι ικανή να μεταφέρει περισσότερες πληροφορίες από ένα απλό ζεύγος χάλκινων καλωδίων για τηλεφωνικούς τηλεπικοινωνιακούς σκοπούς. Το ποσό που συνδέεται με κάθε κανάλι και δείχνει πόση πληροφορία μπορεί να μεταφέρει αξιόπιστα είναι γνωστό ως χωρητικότητα καναλιού $C$.

Η πληροφορία που μπορεί να μεταφέρει αξιόπιστα ένα κανάλι συνδέεται στενά με τη χρήση κώδικα διόρθωσης λάθους. Το κυρίαρχο θεώρημα από τη θεωρία της πληροφορίας είναι το **θεώρημα κωδικοποίησης διαύλου του Shannon**, το οποίο ορίζει ουσιαστικά το εξής: Υπό την προϋπόθεση ότι ο ρυθμός κωδικοποίησης $R$ είναι μικρότερος από την χωρητικότητα καναλιού $C$, υπάρχει ένας κώδικας τέτοιος ώστε η πιθανότητα σφάλματος μπορεί να είναι αυθαίρετα μικρή.

**Ο αποδιαμορφωτής / Ισοσταθμιστής** λαμβάνει το σήμα από το κανάλι και το μετατρέπει σε μια ακολουθία συμβόλων. Αυτό συνήθως περιλαμβάνει πολλές λειτουργίες, όπως φιλτράρισμα, αποδιαμόρφωση, συγχρονισμό μεταφοράς, χρονική εκτίμηση μεταφοράς συμβόλου, συγχρονισμό πλαισίου, αντιστοίχηση φιλτραρίσματος, ακολουθούμενες από ένα βήμα ανίχνευσης σχετικά με αποφάσεις για τη μεταφορά των μεταδιδόμενων συμβόλων.

**Ο αποκωδικοποιητής καναλιού** χρησιμοποιεί τα πλεονάζοντα ψηφία που εισάγονται από τον κωδικοποιητή για να διορθώσει τυχόν λάθη που μπορεί να έχουν εισαχθεί. Όπως παρουσιάζεται και στο παραπάνω σχήμα, ο αποδιαμορφωτής, ισοσταθμιστής και αποκωδικοποίησης είναι δυνατόν να συνδυάζονται, γεγονός που συναντάται ευρέως στους turbo αντισταθμιστές.

**Η decrypter** αίρει κάθε κρυπτογράφηση

.

**Η αποκωδικοποιητής πηγής** παρέχει μια ασυμπίεστη μορφή των δεδομένων.

**Ο προορισμός** είναι η κατάληξη των δεδομένων.

## *4.2. Κωδικοποίηση - Αποκωδικοποίηση*

Αρχικά η είσοδος δεδομένων στο πρόγραμμα Matlab γίνεται με την εκχώρηση τιμών από τον χρήστη με τις παρακάτω εντολές:

```
m=input('number of bits per symbol "m" :')
% Αριθμός bits ανά symbol
k=input('number of symbols per word "k" :')
%μήκος λέξης πριν την κωδικοποίηση(αριθμός από symbols)
noPackets=input('noPackets :')
% Συνολικός αριθμός πακέτων προς επεξεργασία
```

Όπως επισημαίναμε στην ενότητα που εξετάσαμε την Reed – Solomon κωδικοποίηση, το μήκος κωδικοποιημένης λέξης προκύπτει:

```
n=2^m-1
% μήκος λέξης
```

Υπάρχουν δύο βασικές προσεγγίσεις για την δημιουργία της ακολουθίας bit στη προσομοίωση. Ένας τρόπος είναι η δημιουργία κάθε σήματος ξεχωριστά και η κατά ακολουθία επιμέρους επεξεργασία κάθε σήματος. Δεύτερος τρόπος είναι η δημιουργία και η αποθήκευση μιας μεγάλης ποικιλίας από bits επεξεργάζοντάς τα όλα μαζί.

```
messageColumn = randint(noPackets*k*m,1)
% Τυχαία ακολουθία αποτελούμενη από: 0, 1
```

Αυτή η μέθοδος είναι αποτελεσματική στη γλώσσα Matlab που χρησιμοποιούμε όπου οι πράξεις μεταξύ πινάκων είναι ταχύτερες σε σύγκριση με την χρησιμοποίηση βρόχων. Ένας επιπλέον λόγος που με οδήγησε στην δημιουργία μιας μεγάλης εισροής από bits και η επεξεργασία όλων των πακέτων μαζί είναι εξαγωγή ακριβότερων αποτελεσμάτων. Εφόσον για κάθε πακέτο ξεχωριστά όσο η τιμή της πιθανότητας λάθους προκύπτει με περισσότερα δεκαδικά ψηφία και πλησιάζει το μηδέν, τότε στο Matlab η τιμή αυτόματα απλοποιείται σε μηδέν και οι ακόλουθες πράξεις δίνουν γινόμενο μηδέν.

Χρησιμοποιώ την συνάρτηση **code = rsenc (words, n, k)** από την βιβλιοθήκη του Matlab για την διαδικασία της **κωδικοποίησης**. Η τυχαία ακολουθία του εισερχόμενου μηνύματος πρέπει να έχει τη μορφή πίνακα με σύμβολα των m bits μεταφρασμένα στο Galoi πεδίο, και με κάθε γραμμή του πίνακα να αποτελεί ένα πακέτο των k συμβόλων. Αυτό επιτυγχάνεται με τις παρακάτω εντολές:

```
message=reshape(messageColumn,noPackets*k,m)
symbolMessage=bi2de(message,'left-msb')
% converts it to decimal system
words=reshape(symbolMessage,noPackets,k)
% gf is matlab function (all operations in galoi field)
words=gf(words,m)
code = rsenc(words,n,k)
```

Η συνάρτηση rsenc μετατρέπει τον πίνακα: words με διαστάσεις (noPackets,k) σε πίνακα με διαστάσεις (noPackets, n), δηλαδή συμπληρώνει κάθε γραμμή του πίνακα με n-k parity check symbols όπως προκύπτει και από την εξίσωση

$u(X) = X^{n-k} \bullet m(X) + p(X) = X^2 + X^5 + X^3$

$X^5 + X^3 \quad | X^3 + X + 1$

$X^5 + X^3 + X^2 \quad X^2$

$X^2$

$p(X) = X^2$

$$u(X) = X^{n-k} \bullet m(X) + p(X)$$
$$= p_0 + p_1 \bullet X + \cdots + p_{n-k-1}X^{n-k-1} + m_0 X^{n-k} + m_1 X^{n-k+1} + \cdots + m_{k-1}X^{n-1} \quad (3\text{-}7)$$

. Κατά την επιλογή της συγκεκριμένης συνάρτησης για την RS κωδικοποίηση, χρησιμοποιούμε άμεσα το default πολυώνυμο γεννήτριας που προτείνει το Matlab. Σύμφωνα με την εξίσωση $g(x) = (x-a)(x-a^2)(x-a^3)(x-a^4)$ (3-10), το πολυώνυμο γεννήτριας που χρησιμοποιείται είναι της μορφής $g(x) = (x-A^1)(x-A^2)...(x-A^{2t})$ όπου A είναι ρίζα του default primitive πολυωνύμου και για το galoi field GF(n+1) και t αποτελεί την ικανότητα διόρθωσης λάθους του κώδικα όπως δόθηκε στην εξίσωση $t=\left\lfloor \frac{1}{2}(d_{\min}-1) \right\rfloor$ (3-14).

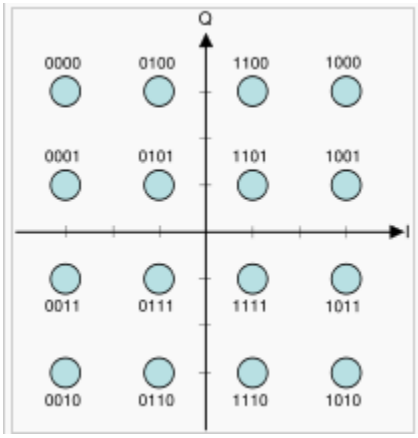
Κατά την αντίστροφη διαδικασία της **αποκωδικοποίησης**, χρησιμοποιώ την συνάρτηση **decoded = rsdec(rxWords,n,k)** του Matlab, όπου επιστρέφει σε κάθε γραμμή του πίνακα rxWords το αποκωδικοποιημένο μήνυμα μήκους k, όσο και το μήκος του αρχικού μηνύματος πριν την κωδικοποίηση. Αποτυχία αποκωδικοποίησης συμβαίνει όταν η ικανότητα διόρθωσης λάθους του κώδικα είναι μικρότερη του αριθμού των λάθος συμβόλων που προέκυψαν κατά την διέλευση του κωδικοποιημένου μηνύματος από το κανάλι. Στις περιπτώσεις αυτές όπου υπάρχει διαφορά συγκρίνοντας τα k σύμβολα κάθε πλαισίου του αρχικού μηνύματος με το αποκωδικοποιημένο μήνυμα, προκύπτει το Output Bit Error Rate, όπου είναι και το μέτρο που μας ενδιαφέρει για την μελέτη του κώδικα σε διάφορες συνθήκες περιβάλλοντος ή υπό διαφορετικές παραμέτρους του κώδικα.

## *4.3. Διαμόρφωση*

Στην είσοδο του διαμορφωτή καταφθάνουν κατά σειρά οι κωδικοποιημένες λέξεις από την έξοδο του κωδικοποιητή. Κάθε σχήμα διαμόρφωσης αντιστοιχεί τα εισερχόμενα δυαδικά ψηφία σε σύμβολα – παλμούς προς μετάδοση. Το απλούστερο σχήμα διαμόρφωσης που θα χρησιμοποιήσουμε είναι η διαμόρφωση BPSK, όπου το ψηφίο 1 αντιστοιχίζεται σε παλμό πλάτους +A και το ψηφίο 0 αντιστοιχίζεται σε παλμό πλάτους –A. Εμείς χρησιμοποιούμε την κανονικοποιημένη τιμή +1 και –1. Άλλο σχήμα διαμόρφωσης είναι η διαμόρφωση M-QAM όπου αντιστοιχίζεται ένα σύμβολο ανά log2M δυαδικά ψηφία. Για παράδειγμα, στη διαμόρφωση 4-QAM ο διαμορφωτής αντιστοιχεί ανά δύο ψηφία (00, 01, 10, ή 11) και ένα σύμβολο οπότε προκύπτουν συνολικά 4 διαφορετικά σύμβολα.

Κατά την QAM (QUADRATURE AMPLITUDE MODULATION) διαμόρφωση, μεταφέρονται δεδομένα με αλλαγή (αυξομείωση) του πλάτους των δύο φερουσών κυμάτων. Αυτά τα δύο κύματα, συνήθως ημιτονοειδή, είναι εκτός φάσης μεταξύ τους 90

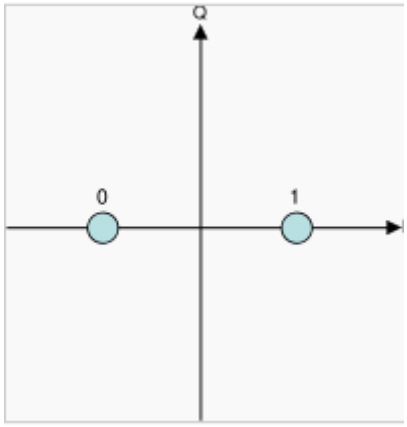
μοιρών και για το λόγο αυτό η διαμόρφωση καλείται τετραγωνική (Quadrature). Ο πρώτος ορθογώνιος QAM αστερισμός που απαντάται συνήθως είναι ο 16-QAM . Μια σύντομη εξήγηση που αποκαλύπτει το γεγονός αυτό είναι ότι οι διαμορφώσεις 2-QAM και 4-QAM είναι στην πραγματικότητα διαμορφώσεις BPSK και QPSK αντίστοιχα. Όσον αφορά την επίπτωση του αριθμού των σημείων στον αστερισμό (όπως παρουσιάζεται στο παρακάτω σχήμα)στο ποσοστό λάθους (Bit Error Rate), περισσότερα σημεία οδηγούν σε υψηλότερη τιμή ποσοστό λάθους. Μια σύντομη εξήγηση για το τελευταίο είναι ότι εφόσον η μέση τιμή της ενέργειας του αστερισμού απαιτείται να διατηρείται σταθερή, τα περισσότερα σημεία χρειάζεται να έρθουν πιο κοντά μεταξύ τους, με συνέπεια να είναι πιο επιρρεπή στα φαινόμενα θορύβου και ακολούθως να δίνουν υψηλότερες τιμές ποσοστού λάθους [11].



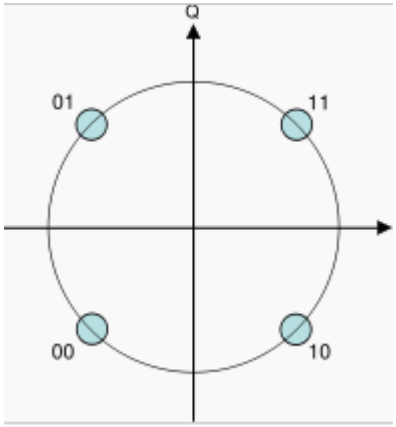

**Σχήμα 4.2** Αστερισμός for rectangular 16-QAM

Συστήματα διαμόρφωσης φάσης (αναλογική PM και ψηφιακή PSK), μπορεί να θεωρηθούν ως ειδική περίπτωση QAM διαμόρφωσης, όπου το μέγεθος του διαμορφωμένου σήματος είναι σταθερό, με μόνη αλλαγή στη φάση του σήματος. Αυτό μπορεί επίσης να επεκταθεί και στην διαμόρφωση συχνότητας (FM) και (FSK), όπου μπορούν να θεωρηθούν μια ειδική περίπτωση της διαμόρφωσης φάσης.

Στην PSK, ο αστερισμός των σημείων είναι η τοποθέτησή τους με ομοιόμορφη γωνιακή απόσταση γύρω από έναν κύκλο (όπως απεικονίζονται: Σχήμα 4.3 Αστερισμός Διαμόρφωσης BPSK Σχήμα 4.4). Η τοποθέτηση αυτή δίνει μέγιστη φάση διαχωρισμού μεταξύ των γειτονικών σημείων και, επομένως, καλύτερη συμπεριφορά στην επίδραση θορύβου. Τα σημεία είναι τοποθετημένα σε έναν κύκλο, ώστε να μπορούν όλα να μεταδοθούν με την ίδια ενέργεια, με συνέπεια να απαιτείται ίδιο πλάτος από όλα στην ημιτονοειδή αναπαράσταση. Δεδομένου ότι τα στοιχεία που θα μεταφέρονται είναι συνήθως δυαδικής μορφής, στην PSK διαμόρφωση διατίθενται συνήθως σημεία που ο συνολικός τους αριθμός είναι βάση του 2 [11].

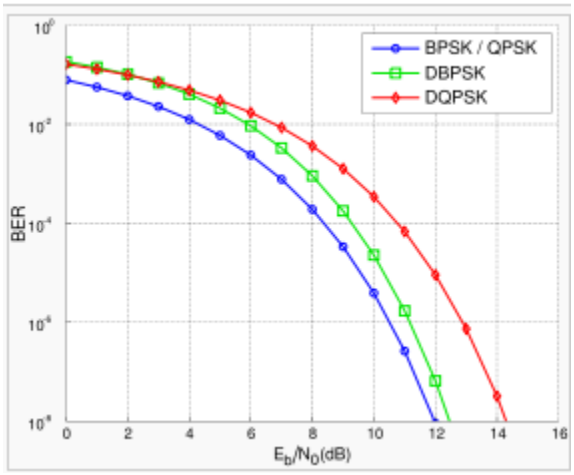



**Σχήμα 4.3** Αστερισμός Διαμόρφωσης BPSK




**Σχήμα 4.4** Αστερισμός Διαμόρφωσης QPSK

Στην περίπτωση της *DPSK διαμόρφωσης*, σε αντίθεση με *PSK διαμόρφωση*, δεν απαιτείται η γνώση της φάσης του εισερχόμενου σήματος αλλά η διαφορά των φάσεων δύο διαδοχικών κωδικοποιημένων σημάτων, και για το λόγο αυτό ονομάζεται *διαφορική κωδικοποίηση*. Μη σύμφωνη αποδιαμόρφωση (*No coherent Demodulation*) αναφέρεται σε συστήματα που απασχολούν αποδιαμορφωτές που είναι σχεδιασμένοι να λειτουργούν χωρίς τη γνώση της απόλυτης τιμής της φάσης του εισερχόμενου μηνύματος. Ως εκ τούτου, δεν απαιτείται εκτίμηση φάσης. Έτσι το πλεονέκτημα της μη σύμφωνης έναντι της σύμφωνης αποδιαμόρφωσης είναι η μειωμένη πολυπλοκότητα, με βάρος την αυξημένη πιθανότητα λάθους (BER). Η ανάλυση δείχνει ότι η διαφορική κωδικοποίηση περίπου διπλασιάζεται το ποσοστό λάθους σε σύγκριση με τη συνήθη M-PSK. Επιπλέον, η ανάλυση αυτή βασίζεται σε ένα σύστημα όπου η διαφθορά είναι μόνο πρόσθετος λευκός Gaussian θόρυβος.




**Σχήμα 4.5** Σύγκριση απόδοσης DBPSK και DQPSK διαμόρφωσης, καθώς και της μη διαφορικής τους μορφής, σε Gaussian Δίαυλο

Ωστόσο, θα υπάρχει επίσης ένα φυσικό κανάλι μεταξύ του πομπού και του δέκτη στο σύστημα επικοινωνίας. Αυτό το κανάλι θα είναι, γενικά, θα εισάγει μια άγνωστη στροφή φάσης στο PSK σήμα. **Σε αυτές τις περιπτώσεις η διαφορική DPSK διαμόρφωση παράγει καλύτερη απόδοση σφάλματος στην έξοδο από τα συνήθη συστήματα που βασίζονται σε ακριβείς πληροφορίες φάσης.** Όταν το κανάλι συνδυάζει διαλείψεις (fading) μαζί με θόρυβο AWGN, η προτεινόμενη από το Matlab ακολουθία είναι η χρήση φίλτρου, δηλαδή της συνάρτησης filter όπως παρουσιάζεται και στο κώδικα πριν την συνάρτηση awgn. Επίσης η προτεινόμενη από το **Matlab** διαμόρφωση σήματος είναι η DBPSK, δηλαδή η διαφορική κωδικοποίηση με την παράμετρο M = 2. Επειδή η απόδοση του συστήματος μελετάται για διάφορα κανάλια (AWGN, Rayleigh, Rician), κρατάμε την παράμετρο της διαμόρφωσης σταθερή: DBPSK. Ο κώδικας Matlab που αντιπροσωπεύει την διαδικασία της διαμόρφωσης δίνεται παρακάτω:

```
M = 2;    % DBPSK signal
dcode = double(code.x)
%converts to double system in order operations be available
dcodebi = de2bi(dcode,'left-msb')
% converts integers to bits.
dcodebi = reshape(dcodebi,noPackets*n*m,1)

dpskSig = dpskmod(dcodebi,M)
%modulation of the message signal dcodebi using differential phase
shift keying modulation
%The message signal must consist of integers between 0 and M-1 (0,1).
%If x is a matrix with multiple rows and columns, the function
processes the columns independently.
metritis = 1
for SNR = 0:2:20
% Range of SNR values, in dB.
```

```
    rxSig = awgn(dpskSig,SNR,'measured',[], 'dB');
    % Add Gaussian noise.
    rx =dpskdemod(rxSig,M);
    % Demodulate.
end
```

## 4.4. Δίαυλος

### 4.4.1. Δίαυλος AWGN

Υποθέτοντας πιθανότητα μετάδοσης λάθους συμβόλου p, η οποία ισούται με τη πιθανότητα μετάδοσης λάθους bit (BER) στην περίπτωση των δυαδικών κωδίκων, για AWGN κανάλι έχουμε:

$$p = Q(\sqrt{\frac{2E_b}{N_0}})$$

Όπου

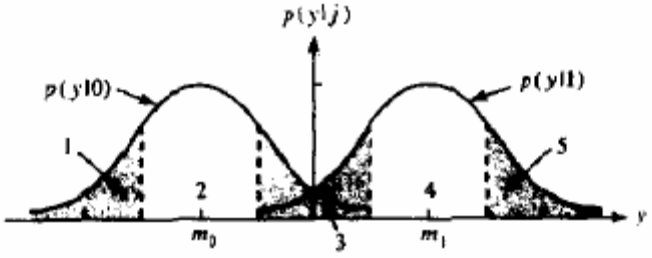
$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{z^2}{2}} dz \qquad , x \geq 0$$

Όπου $Q(x)$ αποτελεί την Gaussian συνάρτηση .

Η πιθανότητα μετάδοσης $P(i/j)$ εξαρτάται από τα χαρακτηριστικά του καναλιού και συγκεκριμένα για AWGN κανάλι η έξοδος του αποδιαμορφωτή κατά τη δειγματοληψία μπορεί να είναι στιγμιαία εκφρασμένη ως

$$p(y \mid j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-m_j)^2/2\sigma^2}, \qquad j = 0, 1$$

Όπου $m_o = \sqrt{E}$ , $m_1 = \sqrt{E}$ και $\sigma^2 = \frac{1}{2}N_0$. Οι δύο περιοχές για $j = \{0,1\}$ παρουσιάζονται στο παρακάτω σχήμα




**Σχήμα 4.** Πιθανότητα Μεταφοράς στην Έξοδο του Αποδιαμορφωτή.

Αναλύοντας προγραμματιστικά την επίδραση AWGN καναλιού στο σήμα εισόδου χρησιμοποιούμε την συνάρτηση awgn από την βιβλιοθήκη συναρτήσεων του matlab. Όταν το σήμα εισόδου ανήκει στο σύνολο των πραγματικών αριθμών, αυτή η συνάρτηση προσθέτει πραγματικό Gaussian θόρυβο και παράγει ένα πραγματικό σήμα εξόδου. Όταν το σήμα εισόδου στο σύνολο των μιγαδικών αριθμών, τότε αντίστοιχα παράγεται ένα μιγαδικό σήμα εξόδου. Συγκεκριμένα για την παραγωγή της πιθανότητας λάθους στην έξοδο του συστήματος, χρησιμοποιήσαμε μεταβλητή παράμετρο τον όρο του σηματοθορυβικού θορύβου SNR υπολογισμένο σε dB. Παρακάτω δίνεται το κομμάτι του κώδικα που αντιπροσωπεύει την είσοδο λευκού προσθετικού θορύβου στο σήμα:

```
M = 2;    % DPSK signal
dpskSig = dpskmod(dcodebi,M)

metritis = 1
for SNR = 0:2:20
    % Range of SNR values, in dB.
    rxSig = awgn(dpskSig,SNR,'measured',[], 'dB');
    % Add Gaussian noise.
    rx =dpskdemod(rxSig,M);
```

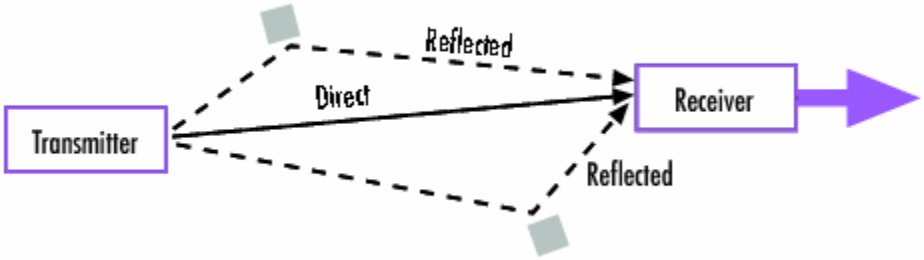
Όπου SNR είναι ο λόγος ισχύς σήματος προς ισχύ θορύβου εκφρασμένος σε dB. Εκφράζοντας τον όρο αυτό στην αντίστοιχη τιμή της ενέργειας bit προς την φασματική πυκνότητα ισχύος θορύβου BER χρησιμοποιούμε τις παρακάτω εντολές στο matlab.

```
EbNo = 10; % In dB
snr = EbNo + 10*log10(k) - 10*log10(nsamp);
ynoisy = awgn(ytx,snr,'measured');
 %% Received Signal
```

Όπου k είναι ο αριθμός της πληροφορίας bits ανά σύμβολο και nsamp είναι παράγοντας που χρησιμοποιείται για την μετατροπή αυτή με default τιμή τη μονάδα. Επιπλέον να σημειώσουμε πως η awgn συνάρτηση χρησιμοποιεί την Ziggurat μέθοδο για την παραγωγή τυχαίων αριθμών, όμοιας λειτουργίας με την γνωστή randn μέθοδο του matlab.

### 4.4.2. Δίαυλος Διαλείψεων (Fading Channel): Rayleigh/Rice

Στην ενότητα αυτή ασχολούμαστε με τη μοντελοποίηση και προσομοίωση εξασθένησης του σήματος οφειλόμενης στο φαινόμενο των πολλαπλών διαδρομών που αποτελούν το βασικότερο αίτιο εξασθένησης στον τομέα των ασύρματων επικοινωνιών. Σε κάθε ασύρματο κανάλι επικοινωνίας μπορεί να υπάρχουν περισσότερες από μία διαδρομές μέσω των οποίων το σήμα μπορεί να ταξιδέψει μεταξύ του πομπού και του δέκτη της κεραίας. Η παρουσία πολλαπλών διαδρομών μπορεί να οφείλεται στην ατμοσφαιρική αντανάκλαση ή διάθλαση, ή αντανακλάσεις από τα κτίρια και άλλα αντικείμενα. Τα αποτελέσματα αυτά, παρατηρήθηκαν και αναλύθηκαν αρχικά για τα HF troposcatter συστήματα κατά τη δεκαετία του 1950 και 1960 [5]. Μεγάλο ενδιαφέρον παρουσιάζεται στη μοντελοποίηση και προσομοίωση εξασθένησης πολλαπλών διαδρομών στην κινητή και ασύρματη επικοινωνία εσωτερικών χώρων στο 1 - 60 GHz εύρος συχνοτήτων.




**Σχήμα 4.6** Πολλαπλές Διαδρομές σε Ασύρματο Περιβάλλον Διάδοσης

Επιπλέον, κάθε συνιστώσα πολλαπλών διαδρομών ή ακτινών είναι δυνατόν να υποβληθεί σε τοπική σκέδαση , λόγω της παρουσίας αντικειμένων, όπως πινακίδες, οδόστρωμα, και δέντρα που βρίσκονται κοντά στο κινητό δέκτη. Το συνολικό μήνυμα που φτάνει στο δέκτη αποτελείται από το άθροισμα ενός μεγάλου αριθμού διάσπαρτων συστατικών που προστίθενται με τυχαίες φάσεις και, ως εκ τούτου, το αποτέλεσμα που προκύπτει μπορεί να μοντελοποιηθεί ως μια σύνθετη Gaussian διαδικασία . Κίνηση για μικρές αποστάσεις της τάξης του λ / 2 (περίπου 15 cm σε 1 GHz) μπορεί να οδηγήσει σε σημαντικές μεταβολές στην φάση των διάσπαρτων συνιστωσών έτσι ώστε ενώ προηγουμένως προσθέτονταν εποικοδομητικά σε μια τοποθεσία, τώρα προσθέτονται καταστροφικά σε μικρή απόσταση από την προηγούμενη. Αυτό έχει ως αποτέλεσμα ταχείς διακυμάνσεις στο λαμβανόμενο σήμα (πλάτους και ισχύς) το οποίο ονομάζεται φαινόμενο **μικρής κλίμακας ή γρήγορης εξασθένησης.**

Θα πρέπει να σημειωθεί ότι η μικρής κλίμακας εξασθένηση προκαλείται από αλλαγές στη φάση και όχι από εξασθένηση μήκους διαδρομής, δεδομένου ότι το μήκος αλλάζει κατά ένα πολύ μικρό ποσοστό όταν μελετούμε μικρές αποστάσεις. Από την άλλη πλευρά, εάν ο κινητός δέκτης κινείται σε μεγάλες αποστάσεις και η διαδρομή αυξάνεται από 1 χλμ. σε 2 χλμ., η ισχύς του λαμβανόμενου σήματος θα μειωθεί, καθώς η

εξασθένηση θα μεταβληθεί σημαντικά. Κίνηση σε μεγάλες αποστάσεις ($\gg \lambda$) και αλλαγές στα χαρακτηριστικά του εδάφους επηρεάζουν την εξασθένηση και την ισχύ του λαμβανόμενου σήματος . Το φαινόμενο αυτό ονομάζεται ***μεγάλης κλίμακας ή αργή εξασθένηση***.

Σε κανάλι γρήγορης εξασθένησης, ο πομπός μπορεί να επωφεληθεί από τις διακυμάνσεις στο κανάλι χρησιμοποιώντας '***time diversity***' για να βελτιώσει την επίδοση του καναλιού σε προσωρινή βαθιά εξασθένηση. Αν και στην περίπτωση της γρήγορης εξασθένησης είναι πολύ πιθανό να διαγραφούν πολλά bits πληροφορίας, ο συνδυασμός χρησιμοποίησης κώδικα διόρθωσης λαθών και αναδιάταξης των bits στο χρόνο (***interleaving***) μπορεί να αποτρέψει την διαγραφή των λανθασμένων αυτών bits. Αντίθετα σε κανάλι αργής εξασθένησης, δεν είναι δυνατή η αναδιάταξη των bits με βάση το χρόνο διότι ο πομπός βλέπει κάθε φορά μια μόνο υλοποίηση του καναλιού εντός του περιορισμένου παραθύρου καθυστέρησης και συνεπώς δεν μπορεί να μετριάσει την βαθιά αργή εξασθένηση που βρίσκεται σε όλο το μήκος του καναλιού παρά τη χρήση κωδικοποίησης [8].

Η συνοχή του χρόνου στο κανάλι συνδέεται με μια ποσότητα γνωστή ως εξάπλωση Doppler (***Doppler spread***) του καναλιού. Όταν ένας χρήστης (ή οι ανακλάσεις στο περιβάλλον του), είναι σε κίνηση, η ταχύτητα της κίνησης προκαλεί μια αλλαγή στην συχνότητα του σήματος που μεταδίδεται κατά μήκος κάθε διαδρομής. Το φαινόμενο αυτό είναι γνωστό ως μετατόπιση Doppler (***Doppler shift***). Τα σήματα κατά μήκος διαφορετικών διαδρομών μπορούν να έχουν διαφορετικές μετατοπίσεις Doppler, που αντιστοιχούν σε διαφορετικά ποσοστά μεταβολής στη φάση τους. Η διαφορά στην μετατόπιση Doppler μεταξύ των διάφορων συνιστωσών σημάτων, συμβάλλουν στην συνολική εξασθένηση (fading) που υπόκειται το κανάλι, φαινόμενο γνωστό ως εξάπλωση Doppler. Τα κανάλια με μεγάλη εξάπλωση Doppler έχουν συνιστώσες σημάτων που έχουν ανεξάρτητη αλλαγή στη φάση με την πάροδο του χρόνου. Από το γεγονός ότι η εξασθένηση εξαρτάται από την προσθήκη (εποικοδομητική ή καταστροφική) των διάφορων συνιστωσών σημάτων, τέτοια κανάλια έχουν πολύ μικρό χρόνο συνοχής [8].

Το περίπλοκο αποτέλεσμα μεγάλου αριθμού συνιστωσών εισροών στο δέκτη είναι μια πολύπλοκη διαδικασία Gaussian. Για την υπόθεση στην οποία η διαδικασία αυτή έχει μηδενική μέση τιμή, ο τύπος της διαδικασίας είναι **Rayleigh.** Εάν μία line-of-sight (LOS) συνιστώσα είναι παρούσα, η διαδικασία γίνεται **Ricean.**

Παρακάτω δίνονται και προγραμματιστικά η εφαρμογή των παραπάνω δύο διαφορετικών τύπων εξασθένησης.

```
chan = rayleighchan(ts,fd);   /    chan = ricianchan (ts,fd, K)
    % Generate data and apply fading channel.
M = 2; % DBPSK modulation order
dpskSig = dpskmod(dcodebi,M)  % DPSK signal
fadedSig = filter(chan,dpskSig) % Effect of channel


SNR = 0:2:20; % Range of SNR values, in dB.
for metritis = 1:length(SNR)
        % Add Gaussian noise.
    rxSig = awgn(fadedSig,SNR(metritis));
        % Demodulate.
```

```
rx = dpskdemod(rxSig,M);
```

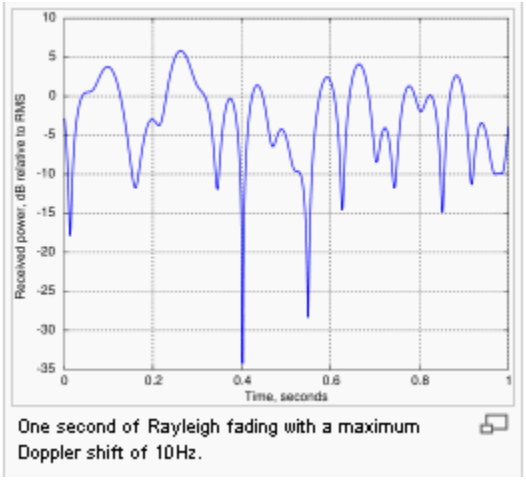
Η συνάρτηση chan = rayleighchan(ts,fd) κατασκευάζει κανάλι εξασθένησης Rayleigh όπου έχουμε μία ή περισσότερες κύριες διαδρομές προκαλούμενες από αντανάκλαση. Όπου ts είναι ο χρόνος δείγματος σήματος εισόδου στο κανάλι σε seconds και fd είναι η μέγιστη μετατόπιση Doppler σε Hertz . Την επίδραση του καναλιού στο εισερχόμενο σήμα, την μοντελοποιούμε με χρήση φίλτρου χρησιμοποιώντας την συνάρτηση fadedSig = filter(chan,dpskSig). Όπως συμβαίνει και στο πραγματικό περιβάλλον, το κανάλι εξασθένησης συνοδεύεται από απλό προσθετικό θόρυβο με την εφαρμογή της συνάρτησης awgn που χρησιμοποιήσαμε και στην προηγούμενη ενότητα.

Η συνάρτηση chan = ricianchan (ts, fd, K) κατασκευάζει κανάλι εξασθένησης Rician όπου έχουμε μια άμεση διαδρομή οπτικής επαφής και πιθανότητα συνδυασμό από μία ή περισσότερες διαδρομές προκαλούμενες από αντανάκλαση. Οι παράμετροι ts,fd, της συνάρτησης ricianchan λαμβάνουν αντίστοιχες τιμές με αυτές της συνάρτησης rayleighchan που εξετάσαμε παραπάνω. Πρόσθετα, ο Rician K παράγοντας είναι μονοδιάστατος παράγοντας και αντιπροσωπεύει την ισχύ της άμεσης διαδρομής.

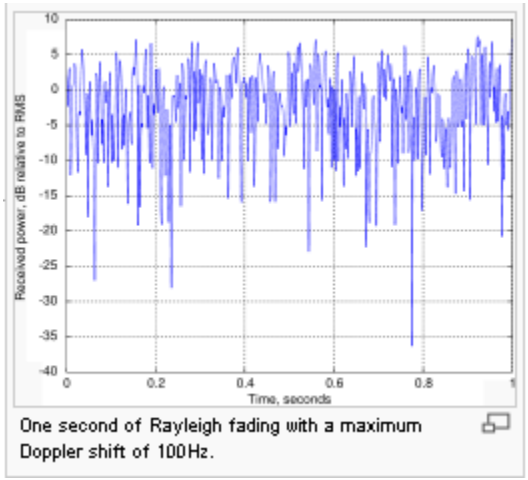
$$K = \frac{\text{ΙΣΧΥΣ ΑΠΕΥΘΕΙΣ ΚΥΜΑΤΟΣ}}{\text{ΙΣΧΥΣ ΣΚΕΔΑΣΜΕΝΩΝ ΚΥΜΑΤΩΝ}}$$

Εάν K=0, το κανάλι Rician συμπεριφέρεται ακριβώς όπως ένα Rayleigh κανάλι όπου δεν υπάρχει άμεση οπτική διαδρομή. Αντίθετα, όσο μεγαλύτερη τιμή έχει η μεταβλητή K, τόσο μικρότερη είναι η επίδραση της εξασθένησης στην πιθανότητα λάθους στην έξοδο του σήματος, και το κανάλι πλησιάζει τη συμπεριφορά AWGN καναλιού.

Στις παραμέτρους των συναρτήσεων rayleighchan και `ricianchan` δηλώνουμε τιμές που σύμφωνα με αυτές χαρακτηρίζουμε ένα κανάλι ως "αργό" (αργή εξασθένηση) ή "γρήγορο" (γρήγορη εξασθένηση). Συγκεκριμένα αν το γινόμενο fd*ts, είναι << 1, π.χ. 10^-4 ή 10^-3 τότε χαρακτηρίζουμε το κανάλι ως "αργό" . Διαφορετικά αν προσεγγίζει ή ξεπερνάει τη μονάδα χαρακτηρίζουμε το κανάλι ως "γρήγορο". Συνεπώς προσδιορίζουμε το πόσο γρήγορα αλλάζει το κανάλι ως προς το σήμα, και παίρνοντας δείγματα από διαφορετικές τιμές της ποσότητας αυτής (`data rate`) έχουμε διαφορετικά αποτελέσματα στην πιθανότητα λάθους στην έξοδο του συστήματος.




**Σχήμα 4.7** Δείγματα σε Δίαυλο Διαλείψεων με μέγιστη συχνότητα Doppler fd=10Hz




**Σχήμα 4.8** Δείγματα σε Δίαυλο Διαλείψεων με μέγιστη συχνότητα Doppler fd=100Hz

Οι μετατοπίσεις Doppler fd=10Hz και fd=100Hz των παραπάνω σχημάτων αντιστοιχούν σε ταχύτητες 6km/h (4 μίλι/ ώρα) και 60km/h (40 μίλι/ ώρα) αντίστοιχα στα 1800 MHz, μία από τις συχνότητες λειτουργίας για τα κινητά τηλέφωνα GSM. Να σημειώσουμε ότι στις βαθιές εξασθενήσεις 'deep fades', η ισχύς του σήματος μειώνεται κατά ένα παράγοντα 30-40 dB.

# 5. Αποτελέσματα Προσομοίωσης Τεχνικών Κωδικοποίησης για την Εκτίμηση της Επίδοσής τους σε Δίαυλο Κινητών Επικοινωνιών.

Ενώ είναι δυνατόν για τους Reed-Solomon κωδικούς να υπολογίσουμε την απόδοσή τους με ακριβείς καμπύλες, αξίζει να εξετάσουμε πώς η απόδοση τους μπορεί να επηρεαστεί από διαφορετικά αρχικά δεδομένα και να προσομοιώσουμε τα αποτελέσματα. Το πρόγραμμα που δίνεται στο παράρτημα υπολογίζει την πιθανότητα λάθους Bit Error Rate (BER) στην έξοδο του αποκωδικοποιητή με δεδομένα τα χαρακτηριστικά του κώδικα $(n, k, \tau)$ ως προς κάποιο χαρακτηριστικό του καναλιού που μπορεί να είναι η ενέργεια του $(E_b / N_0)$, ή το ποσοστό εισαγόμενων λαθών ανά bit - Bit Error Rate ή το ποσοστό εισαγόμενων λαθών ανά σύμβολο - Symbol Error Rate. Επίσης το πρόγραμμα υπολογίζει την πιθανότητα λάθους στην έξοδο του αποκωδικοποιητή με δεδομένα σταθερά χαρακτηριστικά του καναλιού και μεταβλητά κάθε φορά κάποιο από τα χαρακτηριστικά του κώδικα.

## 5.1. Reed-Solomon σε AWGN Δίαυλο

Ο συνήθης τρόπος με τον οποίο εξασφαλίζουμε καλύτερη απόδοση σφάλματος είναι η εισαγωγή κώδικα διόρθωσης λαθών όπως είναι ο Reed Solomon κώδικας με τον οποίο ασχολούμαστε. Κατά την διαδικασία της κωδικοποίησης εισάγονται parity check bits, που χρησιμοποιούνται για τον έλεγχο της λανθασμένης πληροφορίας. Η εισαγωγή αυτών των επιπλέον bits υπαγορεύει ταχύτερο ρυθμό μετάδοσης. που φυσικά σημαίνει μεγαλύτερο εύρος ζώνης. Αυτό είναι και το αντίτιμο της βελτιωμένης συμπεριφοράς στο θόρυβο του κωδικοποιημένου μηνύματος έναντι του μη κωδικοποιημένου. Όπως παρατηρούμε και στο Σχήμα 5.1, για την ίδια τιμή σηματοθορυβικού λόγου του καναλιού, το ποσοστό λάθους στην έξοδο του συστήματος είναι μικρότερο στην περίπτωση του κωδικοποιημένου μηνύματος για όλες σχεδόν τις τιμές του SNR. Εξαίρεση στο συμπέρασμα αυτό παρατηρούμε γα τις πολύ χαμηλές τιμές του SNR (0 – 4 dB), όπου επειδή η ικανότητα διόρθωσης λάθους του κώδικα είναι μικρότερη του αριθμού των λαθών και η ισχύς του σήματος είναι μικρή σχετικά με το θόρυβο, το σύστημα υπερχειλίζεται από λάθη που εισήχθησαν στο κανάλι.

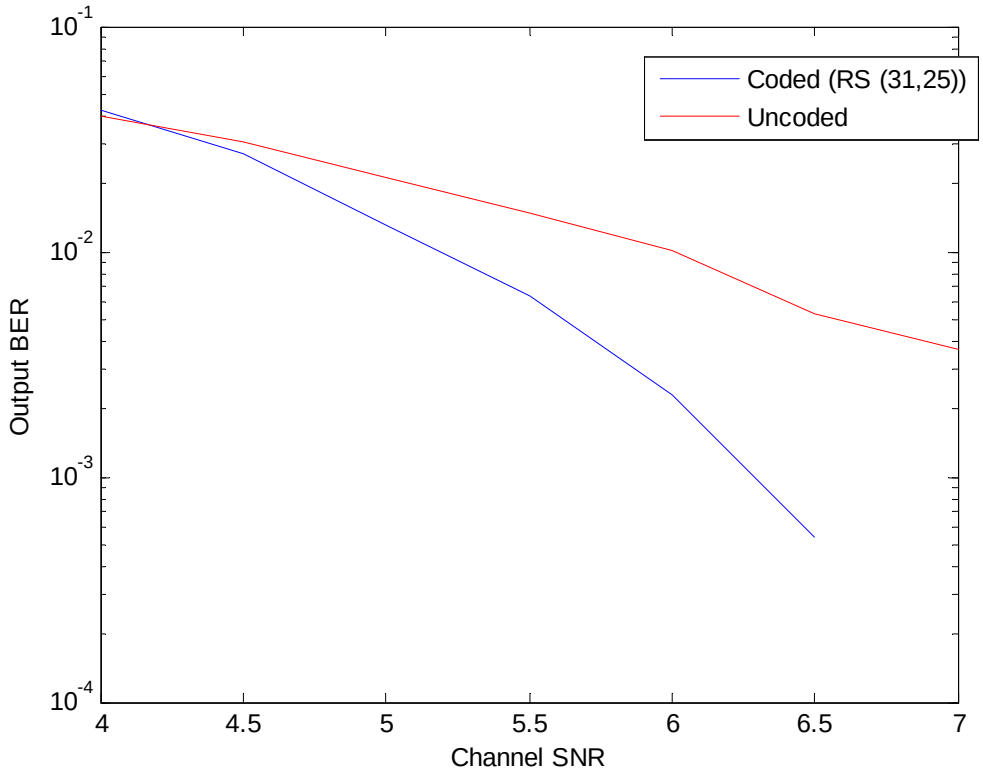
Στο σημείο αυτό μπορούμε να ορίσουμε και το κέρδος κωδικοποίησης ως την διαφορά του λόγου $E_b / N_0$ που απαιτείται από το κανάλι για την παραγωγή της ίδιας τιμής ποσοστού λάθους (BER) στην έξοδο του συστήματος.

$$G(db) = (\frac{E_b}{N_0})_u(db) - (\frac{E_b}{N_0})_c(db)$$

Όπου $(\frac{E_b}{N_0})_u$ και $(\frac{E_b}{N_0})_c$ αντιπροσωπεύουν αντίστοιχα το λόγο $\frac{E_b}{N_0}$ σε $(db)$ για το uncoded και coded μήνυμα αντίστοιχα.

**Πίνακας 5.1** Σύγκριση απόδοσης (BER) κωδικοποιημένου με μη κωδικοποιημένο μήνυμα

| **Coding Data** | **Channel SNR (dB)** | | | | | | |
|---|---|---|---|---|---|---|---|
| m = 7<br>k = 117 | 4 | 4.5 | 5 | 5.5 | 6 | 6.5 | 7 |
| | **Output BER** | | | | | | |
| Coded | 0.0425 | 0.0270 | 0.0131 | 0.0064 | 0.0023 | 0.0005 | 0 |
| Uncoded | 0.0402 | 0.0306 | 0.0212 | 0.0148 | 0.0101 | 0.0053 | 0.0037 |

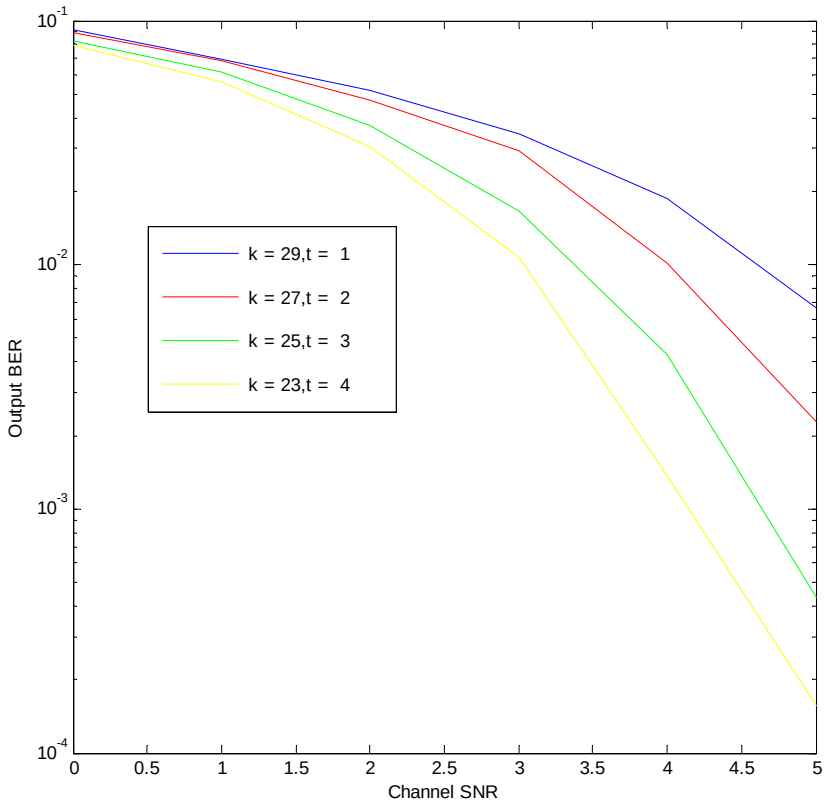



**Σχήμα 5.1** Σύγκριση απόδοσης (BER) κωδικοποιημένου με μη κωδικοποιημένο μήνυμα

Η απόδοση ποσοστού λάθους του κώδικα σε AWGN κανάλι βελτιώνεται με την αύξηση της τιμής της ικανότητας διόρθωσης του κώδικα t. Για τις τιμές $t = 1$, $t = 2$, $t = 3$ και $t = 4$ που αντιστοιχούν σε R-S (31, 29), R-S (31, 27), R-S (31, 25) και R-S (31, 23) παρατηρούμε εμφανή βελτίωση της απόδοσης του συστήματος όπως απεικονίζεται στο Σχήμα 5.2 . Συγκεκριμένα, για μια τιμή πιθανότητας λάθους στην έξοδο του συστήματος (BER) απαιτείται λιγότερος σηματοθορυβικός λόγος (BER) για μεγαλύτερες τιμές της ικανότητας διόρθωσης λάθους του κώδικα t.

**Πίνακας 5.2** Πιθανότητα λάθους στην έξοδο του συστήματος σαν συνάρτηση της ικανότητας διόρθωσης λάθους του Reed-Solomon κώδικα

| **Coding Data** | **Channel SNR (dB)** | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| | **Output BER** | | | | | |
| k = 29<br>t = 1 | 0.0920 | 0.0697 | 0.0520 | 0.0343 | 0.0187 | 0.0066 |
| k = 27<br>t = 2 | 0.0897 | 0.0689 | 0.0474 | 0.0294 | 0.0102 | 0.0023 |
| k = 25<br>t = 3 | 0.0825 | 0.0617 | 0.0371 | 0.0166 | 0.0043 | 0.0004 |
| k = 23<br>t = 4 | 0.0796 | 0.0561 | 0.0306 | 0.0107 | 0.0014 | 0.0002 |



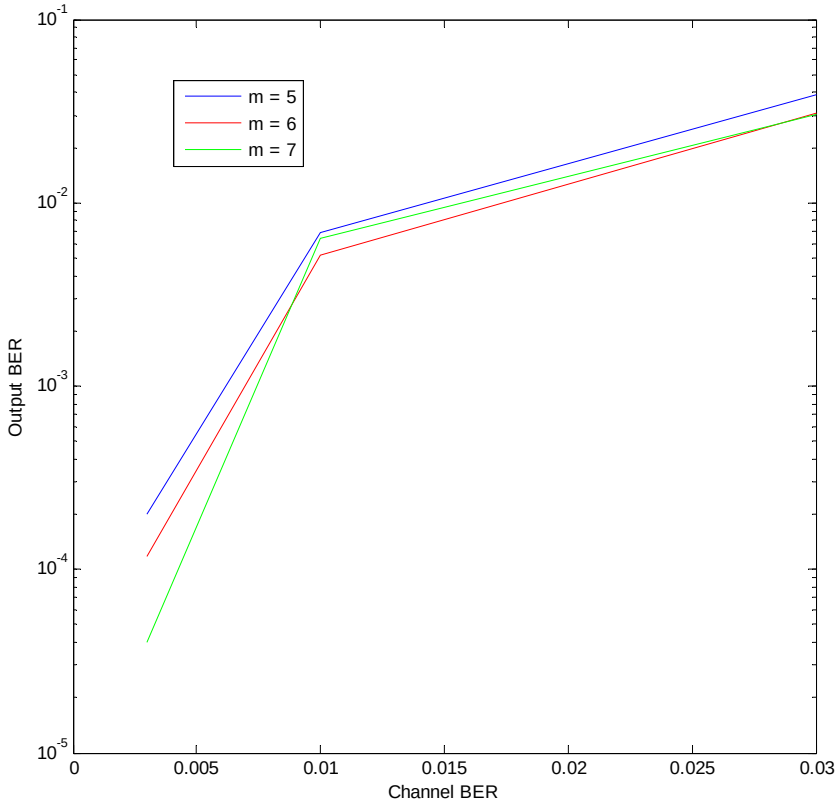

**Σχήμα 5.2** Πιθανότητα λάθους στην έξοδο του συστήματος σαν συνάρτηση της ικανότητας διόρθωσης λάθους του Reed-Solomon κώδικα

Η απόδοση ποσοστού λάθους του κώδικα σε AWGN κανάλι με σταθερό ρυθμό κωδικοποίησης, βελτιώνεται με την αύξηση του μήκους του συμβόλου. Έμεσα μπορεί να εξηγηθεί πως για σταθερό λόγο k/n, με αυξανόμενο m έχουμε και αυξανόμενες τις παραμέτρους k, n και της διαφοράς τους (n-k) που προσδιορίζει ουσιαστικά την ικανότητα διόρθωσης του κώδικα. Συνεπώς Για τιμές m=5 , m=6 και m=7 που αντιστοιχούν σε R-S (31, 27), R-S (63, 55) και R-S (127, 111) παρατηρούμε βελτίωση της απόδοσης του συστήματος όπως απεικονίζεται στο Σχήμα 5.3. Συγκεκριμένα, για μια τιμή πιθανότητας λάθους στην έξοδο του συστήματος (BER) απαιτείται μικρότερη τιμή του ποσοστού λάθους που εισέρχεται στο κανάλι για μικρότερες τιμές του μήκους του συμβόλου m. Πρέπει ωστόσο να επισημάνουμε πως η συμπεριφορά αυτή μπορεί να μην επαληθεύεται για μεγάλες τιμές ποσοστού λάθους στο κανάλι όπου στην περίπτωση αυτή έχουμε υπερχειλισμό λαθών στην έξοδο του συστήματος.

**Πίνακας 5.3** Πιθανότητα λάθους στην έξοδο του συστήματος σαν συνάρτηση του μήκους συμβόλου m με σταθερό το ρυθμό κωδικοποίησης k/n

| **Coding Data** | **Channel BER** |
|---|---|

| (k/n = 0,87 σταθερό) | | 0.00001 | 0.00003 | 0.0001 | 0.0003 | 0.001 | 0.003 |
|---|---|---|---|---|---|---|---|
| | | **Output BER** | | | | | |
| m | 5 | 1.0e-003 * 0 | 1.0e-003 * 0 | 1.0e-003 * 0 | 1.0e-003 * 0 | 1.0e-003 * 0.0222 | 1.0e-003 *0.2444 |
| k | 27 | | | | | | |
| n | 31 | | | | | | |
| t | 2 | | | | | | |
| k/n | 0,87 | | | | | | |
| Packets | 1000 | | | | | | |
| m | 6 | 1.0e-003 * 0 | 1.0e-003 * 0 | 1.0e-003 * 0 | 1.0e-003 * 0 | 1.0e-003 * 0 | 1.0e-003 * 0.1121 |
| k | 55 | | | | | | |
| n | 63 | | | | | | |
| t | 4 | | | | | | |
| k/n | 0,87 | | | | | | |
| Packets | 1000 | | | | | | |
| m | 7 | 1.0e-005 * 0 | 1.0e-005 * 0 | 1.0e-005 * 0 | 1.0e-005 * 0 | 1.0e-005 * 0 | 1.0e-005 * 0.7722 |
| k | 111 | | | | | | |
| n | 127 | | | | | | |
| t | 8 | | | | | | |
| k/n | 0,87 | | | | | | |
| Packets | 1000 | | | | | | |

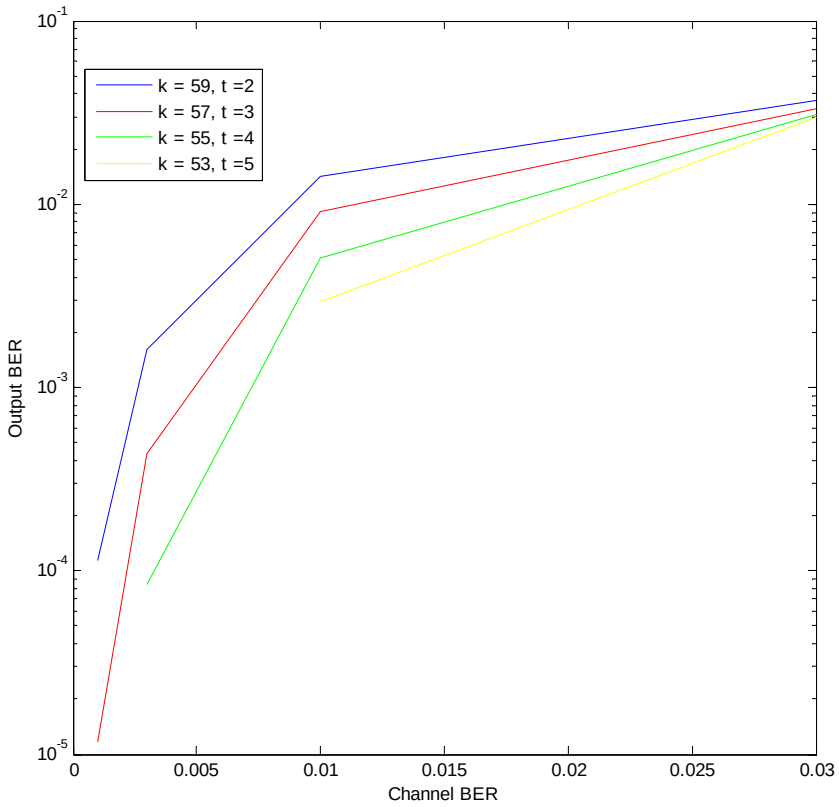



**Σχήμα 5.3** Πιθανότητα λάθους στην έξοδο του συστήματος σαν συνάρτηση του μήκους συμβόλου m με σταθερό το ρυθμό κωδικοποίησης k/n

Όπως εξετάσαμε και προηγουμένως, η απόδοση ποσοστού λάθους του κώδικα σε AWGN κανάλι βελτιώνεται με την αύξηση της τιμής της ικανότητας διόρθωσης του κώδικα t. Η μόνη διαφορά της παρούσας παράστασης είναι ότι τα αποτελέσματα εξάγονται για διάφορες τιμές ποσοστού λάθους που εισέρχεται στο κανάλι (Σχήμα 5.4) και όχι ως προς το σηματοθορυβικό λόγο του καναλιού (Σχήμα 5.2). Συγκεκριμένα, για τις τιμές $t = 2$, $t = 3$, $t = 4$ και $t = 5$ που αντιστοιχούν σε R-S (63, 59), R-S (63, 57) , R-S (63, 55) και R-S 63, 53) παρατηρούμε εμφανή βελτίωση της απόδοσης του συστήματος. Για μια τιμή πιθανότητας λάθους στην έξοδο του συστήματος (BER) απαιτείται μικρότερη τιμή του ποσοστού λάθους που εισέρχεται στο κανάλι για μικρότερες τιμές ικανότητας διόρθωσης λάθους του κώδικα.

**Πίνακας 5.4** Πιθανότητα λάθους στην έξοδο του συστήματος σαν συνάρτηση της ικανότητας διόρθωσης λάθους του Reed-Solomon κώδικα (είσοδος BER Διαύλου)

| **Coding Data** | **Channel BER** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| m = 6<br>n = 63 | 0.00001 | 0.00003 | 0.0001 | 0.0003 | 0.001 | 0.003 | 0.01 | 0.03 |
| | **Output BER** | | | | | | | |
| k = 59<br>t = 2 | 0 | 0 | 0 | 0 | 0.0000 | 0.0016 | 0.0140 | 0.0370 |
| k = 57<br>t = 3 | 0 | 0 | 0 | 0 | 0.0000 | 0.0002 | 0.0085 | 0.0330 |
| k = 55<br>t = 4 | 0 | 0 | 0 | 0 | 0 | 0.0001 | 0.0052 | 0.0313 |
| k = 53<br>t = 5 | 0 | 0 | 0 | 0 | 0 | 0.0000 | 0.0030 | 0.0296 |



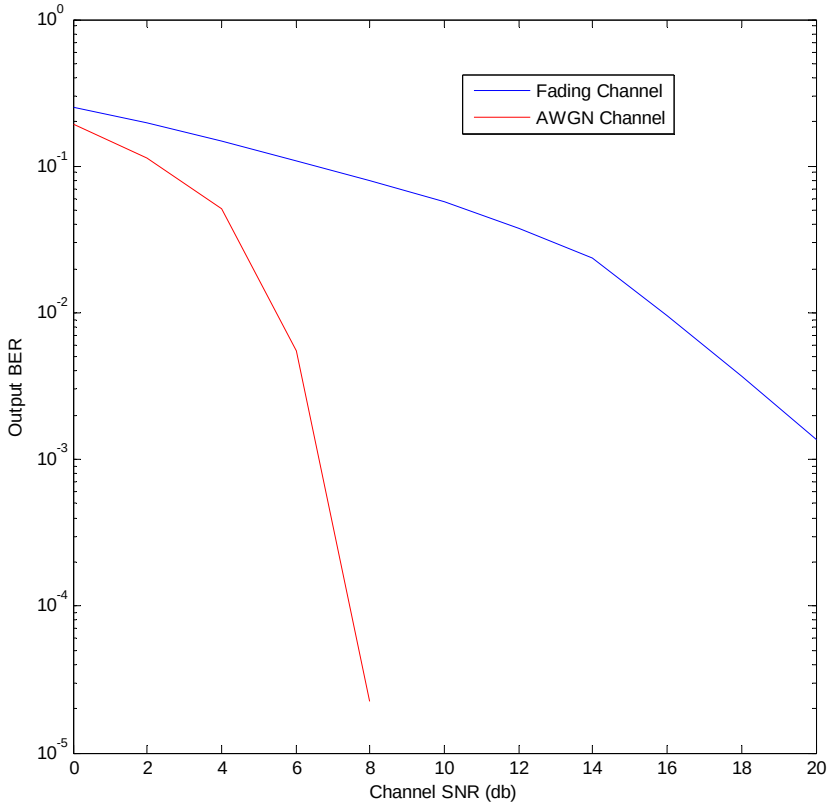

**Σχήμα 5.4** Πιθανότητα λάθους στην έξοδο του συστήματος σαν συνάρτηση της ικανότητας διόρθωσης λάθους του Reed-Solomon κώδικα (είσοδος BER Διαύλου)

## *5.2. Reed-Solomon σε Δίαυλο Διαλείψεων: Rayleigh/ Ricean*

Στην ενότητα αυτή εξετάζουμε την επίδοση του RS κώδικα υπό διαφορετικά κανάλια. Η απόδοση ποσοστού λάθους του κώδικα σε AWGN κανάλι συγκριτικά με ένα fading κανάλι είναι αρκετά βελτιωμένη. Με το γενικό όρο κανάλι fading, εννοούμε εξασθένηση Rayleigh, και στα αποτελέσματα της προσομοίωσης του καναλιού (Σχήμα 5.5) καταλήξαμε θεωρώντας μηδενική μετατόπιση Doppler. Όπως απεικονίζεται και στο παρακάτω σχήμα, για ένα R-S (31, 27) κώδικα, για μια τιμή πιθανότητας λάθους στην έξοδο του συστήματος (BER) απαιτείται μεγαλύτερος σηματοθορυβικός λόγος σε ένα Rayleigh κανάλι σχετικά με ένα κανάλι στο οποίο υπεισέρχεται μόνο λευκός Gaussian προσθετικός θόρυβος.

**Πίνακας 5.5** Σύγκριση απόδοσης (BER) AWGN διαύλου με Fading (Rayleigh) δίαυλο (χωρίς μετατόπιση Doppler, Rician παράγοντας K = 1)

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 5<br>k = 27 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| Fading (Rayleigh) | 0.2505 | 0.1986 | 0.1493 | 0.1094 | 0.0784 | 0.0556 | 0.0377 | 0.0200 | 0.0103 | 0.0041 |
| AWGN | 0.1933 | 0.1136 | 0.0515 | 0.0055 | 0.0000 | 0 | 0 | 0 | 0 | 0 |



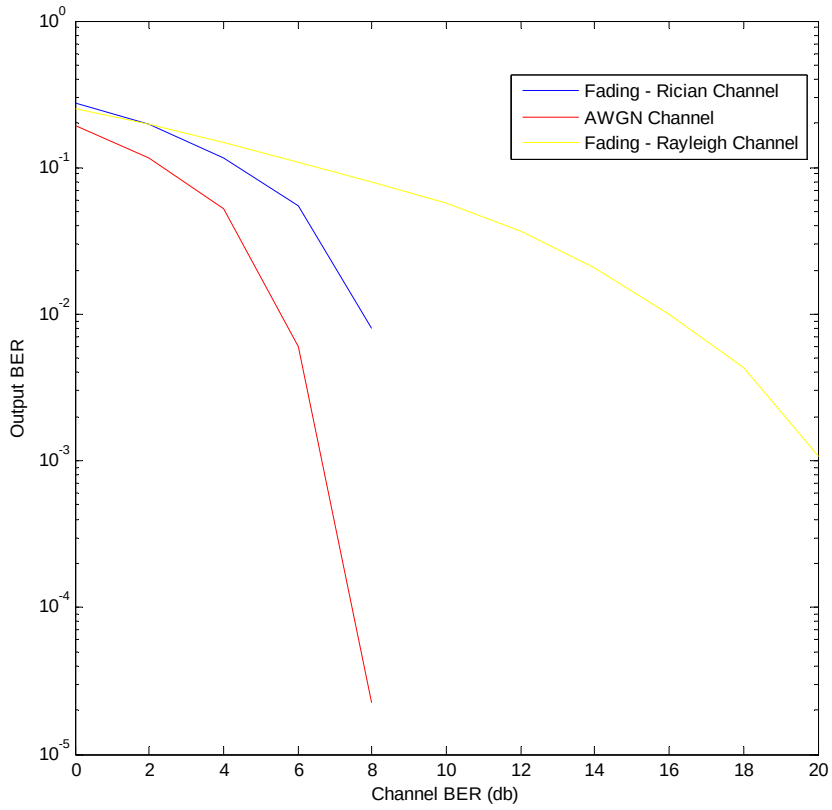

**Σχήμα 5.5** Σύγκριση απόδοσης (BER) AWGN διαύλου με Fading (Rayleigh) δίαυλο (χωρίς μετατόπιση Doppler, Rician παράγοντας K = 1)

Όπως προαναφέραμε, η απόδοση ποσοστού λάθους του κώδικα σε AWGN κανάλι συγκριτικά με ένα fading κανάλι είναι αρκετά βελτιωμένη. Στο Σχήμα 5.6 δίνουμε μεγαλύτερη έμφαση στην μορφή της εξασθένησης και συγκεκριμένα αναλύουμε την

απόδοση του συστήματος υπό Rayleigh και Rician εξασθένηση συγκριτικά με απλό AWGN κανάλι. Κατά την μοντελοποίηση της προσομοίωσης θεωρήσαμε μηδενική μετατόπιση Doppler, και καταλήξαμε όπως απεικονίζεται στο παρακάτω σχήμα πως η Rician εξασθένηση δίνει καλύτερη επίδοση του κώδικα σχετικά με την Rayleigh εξασθένηση, όπου στην τελευταία δεν υπάρχει άμεση οπτική διαδρομή μετάδοσης του σήματος.

**Πίνακας 5.6** Σύγκριση απόδοσης (BER) AWGN, Rayleigh και Rician διαύλου

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 5<br>k = 27 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| Rician<br>k=1<br>No Doppler | 0.2740 | 0.1975 | 0.1171 | 0.0541 | 0.0077 | 0 | 0 | 0 | 0 | 0 |
| AWGN | 0.1933 | 0.1136 | 0.0515 | 0.0055 | 0.0000 | 0 | 0 | 0 | 0 | 0 |
| Rayleigh<br>k=1<br>No Doppler | 0.2509 | 0.1969 | 0.1475 | 0.1079 | 0.0792 | 0.0575 | 0.0367 | 0.0208 | 0.0099 | 0.0043 |



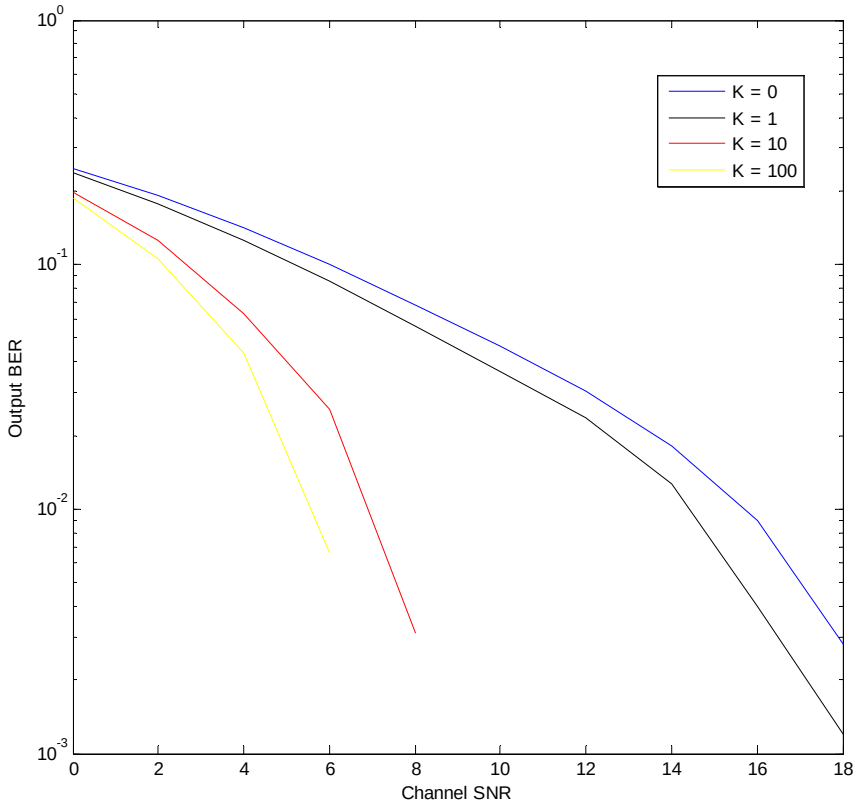

**Σχήμα 5.6** Σύγκριση απόδοσης (BER) AWGN, Rayleigh και Rician διαύλου

Με την παρακάτω προσομοίωση εξετάζουμε την επίδοση του RS κώδικα υπό Rician κανάλι εξασθένησης για διαφορετικές τιμές της μεταβλητής K. Για μεγαλύτερες τιμές της μεταβλητής K, έχουμε λιγότερη επίδραση της εξασθένησης στην πιθανότητα λάθους στην έξοδο του σήματος και το κανάλι πλησιάζει τη συμπεριφορά AWGN καναλιού. Αντίθετα, εάν K=0 τότε το Rician κανάλι αντιστοιχεί σε κανάλι Rayleigh με μηδενική άμεση οπτική διαδρομή.

**Πίνακας 5.7** Απόδοση Rician Διάλου σαν συνάρτηση της Rician παραμέτρου K

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 6<br>k = 55<br>ts = 1/10.000<br>fd = 100 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| Rician - k = 0 | 0.2484 | 0.1922 | 0.1407 | 0.0998 | 0.0686 | 0.0465 | 0.0304 | 0.0182 | 0.0090 | 0.0028 |
| Rician - k = 1 | 0.2364 | 0.1772 | 0.1259 | 0.0850 | 0.0559 | 0.0366 | 0.0235 | 0.0126 | 0.0040 | 0.0012 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Rician - k = 10 | 0.1982 | 0.1255 | 0.0627 | 0.0257 | 0.0031 | 0.0000 | 0 | 0 | 0 | 0 |
| Rician - k = 100 | 0.1873 | 0.1062 | 0.0437 | 0.0066 | 0 | 0 | 0 | 0 | 0 | 0 |

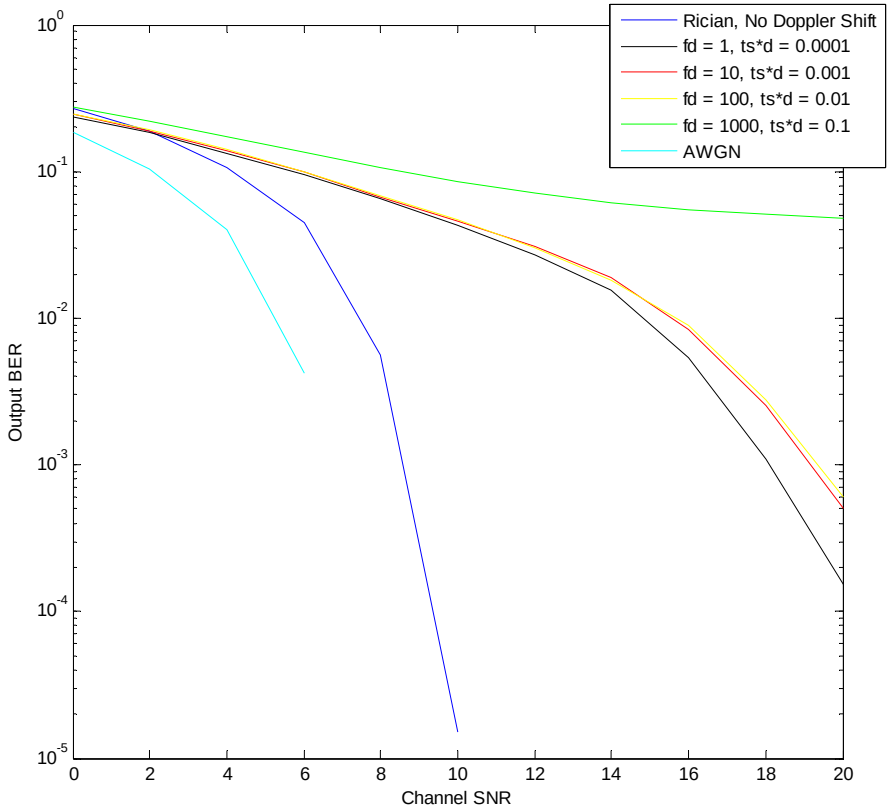



**Σχήμα 5.7** Απόδοση Rician Διάλου σαν συνάρτηση της Rician παραμέτρου K

Στα σχήματα: Σχήμα 5.8, Σχήμα 5.9, Σχήμα 5.10 και Σχήμα 5.11 μοντελοποιούμε την επίδοση του καναλιού ως προς τον παράγοντα Doppler μετατόπισης. Όπως παρατηρήσαμε στο προηγούμενο σχήμα προσομοίωσης, η επίδοση του κώδικα διαφέρει ανάλογα με την τιμή του Rician παράγοντα K. Οπότε τα σχήματα που ακολουθούν αντιστοιχούν σε προσομοιώσεις που ξεκινούν με χαμηλή τιμή K (K=0, Rayleigh εξασθένηση) και καταλήγουν σε μοντελοποίηση καναλιού με K=100 (Rician εξασθένηση με σημαντική επίδραση της άμεσης οπτικής διαδρομής).

**Πίνακας 5.8** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=0)

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 6<br>k = 55<br>ts = 1/10.000<br>K = 0 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| Rician - Fd =1 | 0.2366 | 0.1822 | 0.1339 | 0.0946 | 0.0640 | 0.0433 | 0.0277 | 0.0155 | 0.0053 | 0.0010 |
| Rician - Fd =10 | 0.2465 | 0.1881 | 0.1395 | 0.0992 | 0.0672 | 0.0456 | 0.0310 | 0.0189 | 0.0083 | 0.0025 |
| Rician - Fd =100 | 0.2484 | 0.1922 | 0.1407 | 0.0998 | 0.0686 | 0.0465 | 0.0304 | 0.0182 | 0.0090 | 0.0028 |
| Rician - Fd =1000 | 0.2718 | 0.2206 | 0.1733 | 0.1348 | 0.1059 | 0.0849 | 0.0712 | 0.0612 | 0.0549 | 0.0509 |
| Rician – NO DOPPLER | 0.2710 | 0.1914 | 0.1085 | 0.0441 | 0.0059 | 0 | 0 | 0 | 0 | 0 |
| AWGN Channel | 0.1836 | 0.1030 | 0.0399 | 0.0042 | 0 | 0 | 0 | 0 | 0 | 0 |

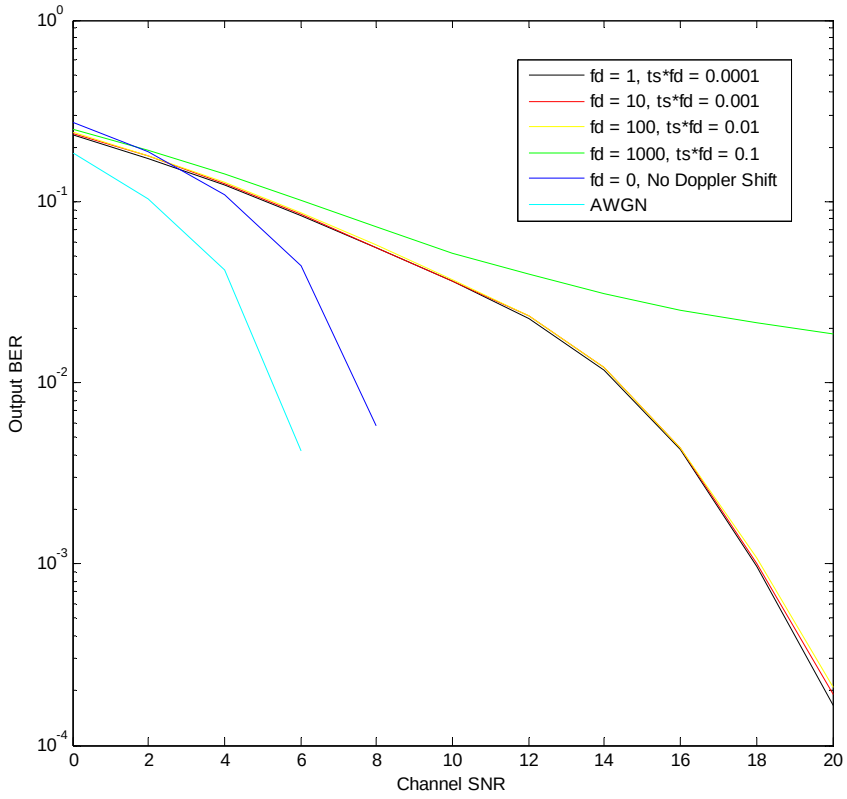



**Σχήμα 5.8** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=0)

**Πίνακας 5.9** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=1)

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 6<br>k = 55<br>ts = 1/10.000<br>K = 1 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| Rician - Fd =1 | 0.2328 | 0.1735 | 0.1225 | 0.0835 | 0.0551 | 0.0362 | 0.0227 | 0.0118 | 0.0042 | 0.0010 |
| Rician - Fd =10 | 0.2368 | 0.1771 | 0.1253 | 0.0848 | 0.0560 | 0.0366 | 0.0233 | 0.0121 | 0.0044 | 0.0010 |
| Rician - Fd =100 | 0.2391 | 0.1799 | 0.1273 | 0.0861 | 0.0572 | 0.0372 | 0.0233 | 0.0121 | 0.0044 | 0.0011 |
| Rician - Fd =1000 | 0.2493 | 0.1922 | 0.1414 | 0.1010 | 0.0721 | 0.0522 | 0.0394 | 0.0309 | 0.0309 | 0.0251 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Rician – NO DOPPLER | 0.2716 | 0.1897 | 0.1081 | 0.0442 | 0.0058 | 0 | 0 | 0 | 0 | 0 |
| AWGN Channel | 0.1849 | 0.1038 | 0.0417 | 0.0042 | 0 | 0 | 0 | 0 | 0 | 0 |

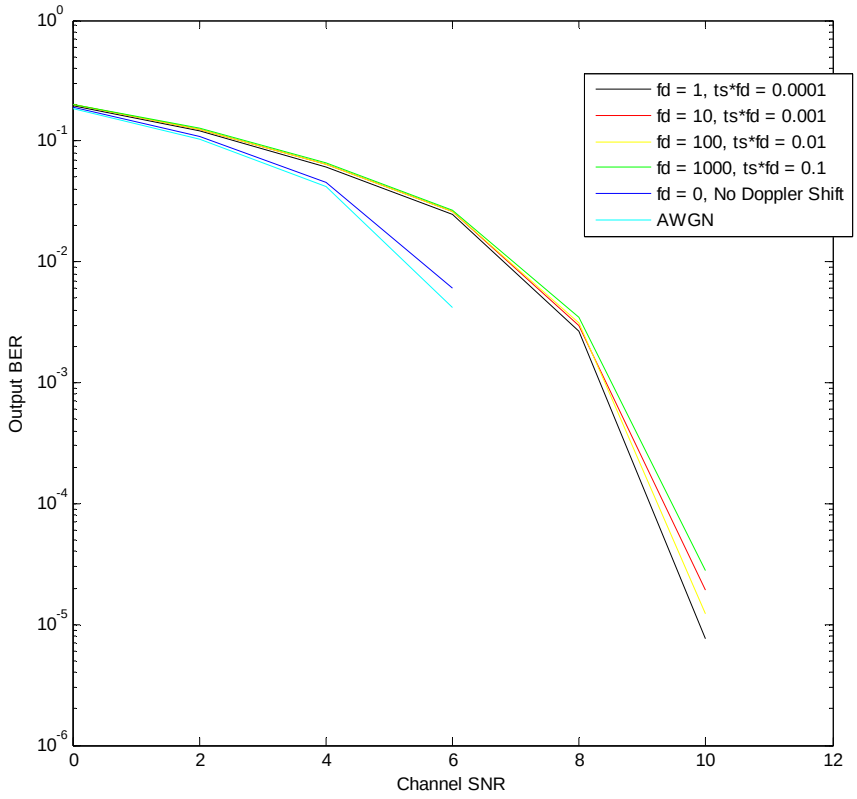



**Σχήμα 5.9** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=1)

**Πίνακας 5.10** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=10)

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 6<br>k = 55<br>ts = 1/10.000<br>K = 10 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |

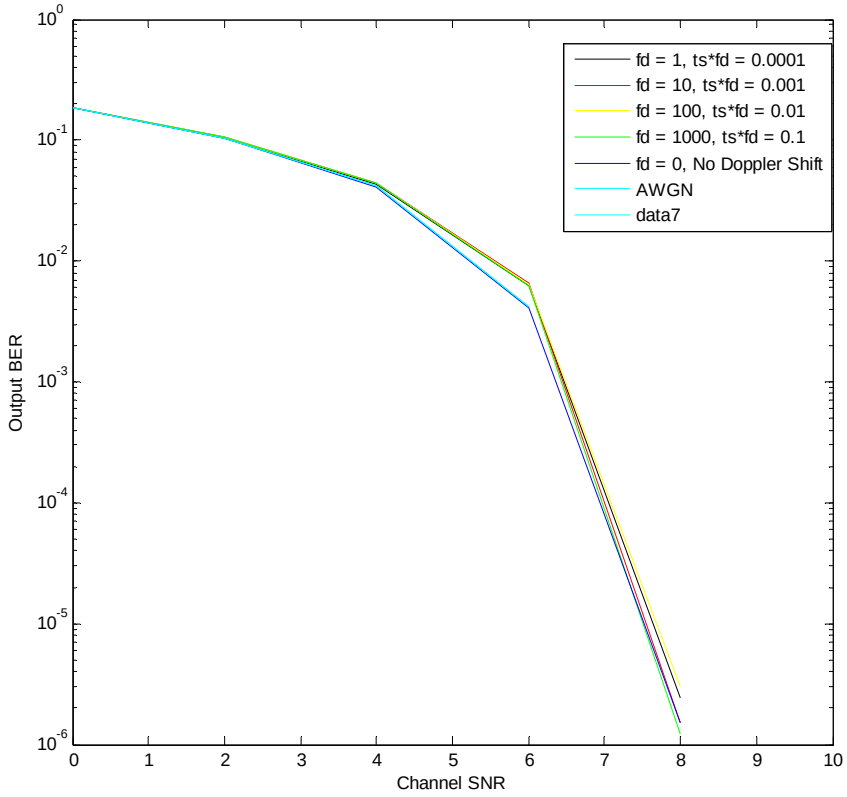
| | Output BER | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Rician - Fd =1 | 0.1961 | 0.1214 | 0.0615 | 0.0250 | 0.0027 | 0.0000 | 0 | 0 | 0 | 0 |
| Rician - Fd =10 | 0.1988 | 0.1241 | 0.0637 | 0.0262 | 0.0030 | 0.0000 | 0 | 0 | 0 | 0 |
| Rician - Fd =100 | 0.1992 | 0.1249 | 0.0638 | 0.0263 | 0.0031 | 0.0000 | 0 | 0 | 0 | 0 |
| Rician - Fd =1000 | | | | | | | | | | |
| Rician – NO DOPPLER | 0.1913 | 0.1094 | 0.0452 | 0.0061 | 0 | 0 | 0 | 0 | 0 | 0 |
| AWGN Channel | 0.1849 | 0.1038 | 0.0417 | 0.0042 | 0 | 0 | 0 | 0 | 0 | 0 |




**Σχήμα 5.10** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=10)

**Πίνακας 5.11** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=100)

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 6<br>k = 55<br>ts = 1/10.000<br>K = 100 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| Rician - Fd =1 | 0.1851 | 0.1048 | 0.0432 | 0.0062 | 0.0000 | 0 | 0 | 0 | 0 | 0 |
| Rician - Fd =10 | 0.1864 | 0.1058 | 0.0441 | 0.0065 | 0.0000 | 0 | 0 | 0 | 0 | 0 |
| Rician - Fd =100 | 0.1864 | 0.1053 | 0.0441 | 0.0064 | 0.0000 | 0 | 0 | 0 | 0 | 0 |
| Rician - Fd =1000 | | | | | | | | | | |
| Rician – NO DOPPLER | 0.1833 | 0.1023 | 0.0408 | 0.0041 | 0.0000 | 0 | 0 | 0 | 0 | 0 |
| AWGN Channel | 0.1849 | 0.1038 | 0.0417 | 0.0042 | 0 | 0 | 0 | 0 | 0 | 0 |



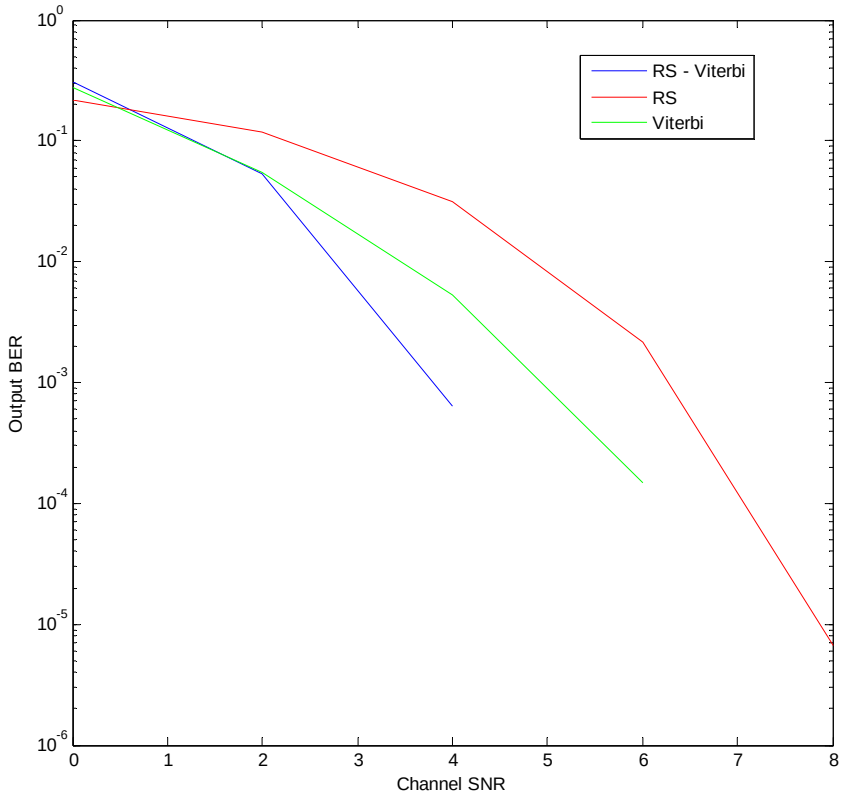

**Σχήμα 5.11** Απόδοση Rician Διάλου σαν συνάρτηση της μέγιστης συχνότητας μετατόπησης Doppler (Rician παράμετρος K=100)

## *5.3. Reed-Solomon - Viterbi σε AWGN Δίαυλο*

Στην ενότητα αυτή μελετάμε την επίδοση του Concatenated RS-Viterbi κώδικα υπό διαφορετικές συνθήκες. Όσον αφορά την προγραμματιστική υλοποίηση του Viterbi, χρησιμοποιώ χαρακτηριστικά που βελτιώνουν την επίδοση του και συνεπώς βελτιώνουν και την επίδοση οποιουδήποτε συνδυασμού του, όπως είναι ο Concatenated RS-Viterbi κώδικας που ασχολούμαστε. Στο Σχήμα 5.12 μοντελοποιούμε την επίδοση του συστήματος για απλό RS και Viterbi κώδικα σχετικά με τον συνδυασμό τους Concatenated RS-Viterbi κώδικα. Όπως περιμένουμε και από τα θεωρητικά αποτελέσματα των προηγούμενων παραγράφων, στο παρακάτω σχήμα απεικονίζεται πως για μια οποιαδήποτε τιμή του SNR καναλιού, η πιθανότητα λάθους στην έξοδο του συστήματος (BER) για τον Concatenated κώδικα είναι μικρότερη σχετικά με την χρησιμοποίηση απλού RS ή Viterbi κώδικα. Μια παρέκκλιση από αυτήν την απόδοση έχουμε για πολύ χαμηλές τιμές του σηματοθορυβικού θορύβου (0-1 dB), όπου ο Concatenated κώδικας επηρεάζεται από τον Viterbi, υπερχειλίζεται από λάθη όπου δεν είναι ικανός να τα διορθώσει εξαιτίας της μικρής τιμής ισχύς του σήματος.

**Πίνακας 5.12** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi κωδικοπίησης, RS κωδικοπίησης και Viterbi κωδικοπίησης

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 3<br>k = 5<br>noPackets = 10000 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| RS - Viterbi | 0.3040 | 0.0527 | 0.0006 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RS | 0.2167 | 0.1179 | 0.0318 | 0.0022 | 0.0000 | 0 | 0 | 0 | 0 | 0 |
| Viterbi | 0.2788 | 0.0553 | 0.0053 | 0.0001 | 0 | 0 | 0 | 0 | 0 | 0 |



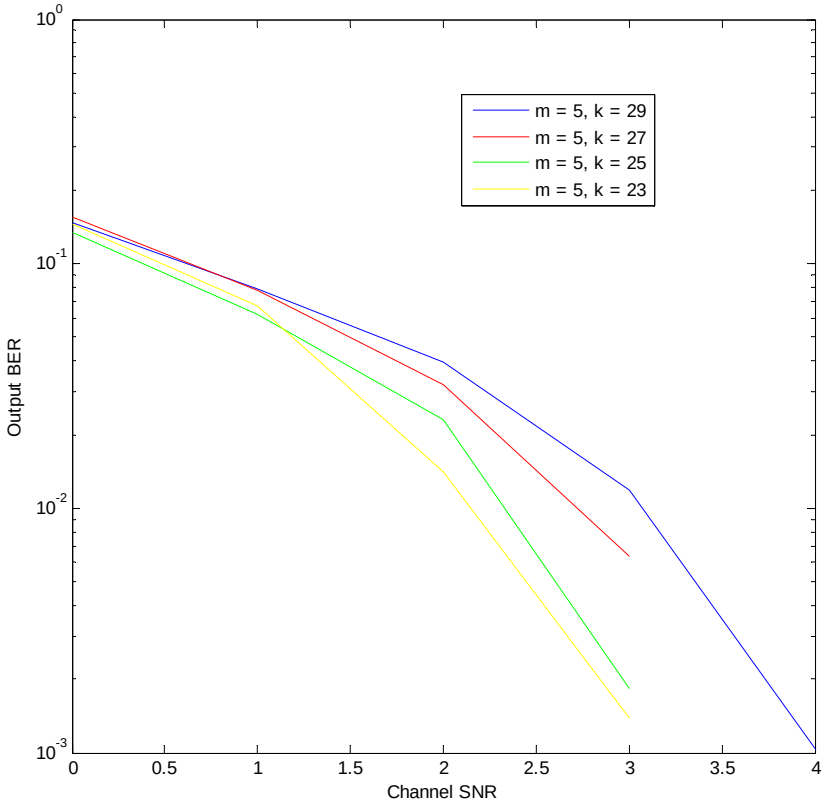

**Σχήμα 5.12** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi κωδικοπίησης, RS κωδικοπίησης και Viterbi κωδικοπίησης

Η απόδοση του ποσοστού λάθους του κώδικα σε AWGN κανάλι βελτιώνεται με την αύξηση της τιμής της ικανότητας διόρθωσης του κώδικα t. Συνεπώς με την αύξηση του t, ή αντίστοιχα με την μείωση του μήκους αποκωδικοποιημένου συμβόλου k παρατηρούμε εμφανή βελτίωση της απόδοσης του συστήματος όπως απεικονίζεται στο **Σχήμα 5.13**.

**Πίνακας 5.13** Απόδοση (BER) συναλυσώμενης Reed-Solomon -Viterbi κωδικοπίησης σαν συνάρτηση της ικανότητας διόρθωσης του RS κώδικα

| **Coding Data** | **Channel SNR (dB)** | | | | | | |
|---|---|---|---|---|---|---|---|
| m = 5 noPackets = 100 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| | **Output BER** | | | | | | |
| m = 5 k = 29 | 0.1481 | 0.0786 | 0.0396 | 0.0119 | 0.0010 | 0 | 0 |
| m = 5 k = 27 | 0.1553 | 0.0780 | 0.0319 | 0.0064 | 0 | 0 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| m = 5<br>k = 25 | 0.1347 | 0.0621 | 0.0230 | 0.0018 | 0 | 0 | 0 |
| m = 5<br>k = 23 | 0.1458 | 0.0672 | 0.0140 | 0.0014 | 0 | 0 | 0 |




**Σχήμα 5.13** Απόδοση (BER) συναλυσώμενης Reed-Solomon -Viterbi κωδικοπίησης σαν συνάρτηση της ικανότητας διόρθωσης του RS κώδικα

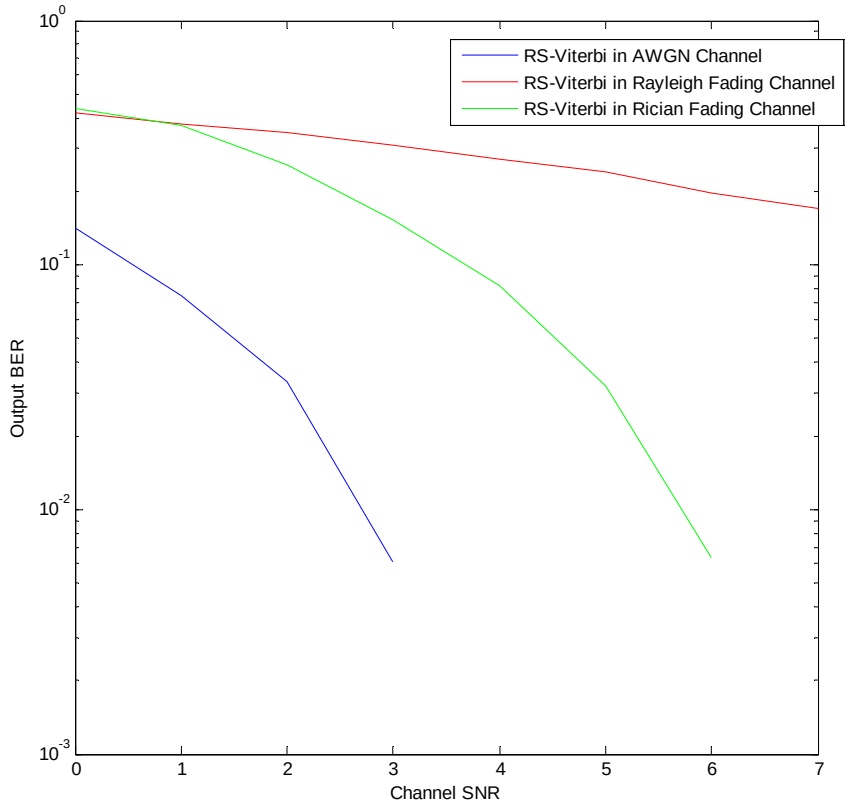
## *5.4. Reed-Solomon - Viterbi σε Δίαυλο Διαλείψεων: Rayleigh/ Ricean*

Τα αποτελέσματα της προσομοίωσης των σχημάτων της παρούσας παραγράφου αφορούν τον Concatenated RS-Viterbi κώδικα υπό την αλλαγή κάποιων παραγόντων. Η απόδοση ποσοστού λάθους του κώδικα σε AWGN κανάλι είναι, όπως περιμέναμε και από τα θεωρητικά αποτελέσματα, βελτιωμένη σε σύγκριση με παρουσία Rayleigh ή Rician εξασθένησης. Επιπλέον, όπως ισχύει και για τον απλό RS κώδικα που εξετάσαμε

στην παράγραφο 5.2, προκύπτουν χαμηλότερες τιμές πιθανότητας λάθους σε εξασθένηση Rician σχετικά με Rayleigh εξασθένηση.

**Πίνακας 5.14** Σύγκριση Απόδοσης (BER) Συναλυσώμενης Reed-Solomon -Viterbi Κωδικοπίησης σε AWGN, Rayleigh και Rician Διαύλους

| **Coding Data** | **Channel SNR (dB)** | | | | | |
|---|---|---|---|---|---|---|
| m = 5<br>k = 27<br>noPackets = 100<br>ts = 0.0001<br>fd = 100 | 0 | 1 | 2 | 3 | 4 | 5 |
| | **Output BER** | | | | | |
| RS-Viterbi in AWGN Channel | 0.1416 | 0.0749 | 0.0335 | 0.0061 | 0 | 0 |
| RS-Viterbi in Rayleigh Fading Channel | 0.4213 | 0.3786 | 0.3486 | 0.3090 | 0.2698 | 0.2394 |
| RS-Viterbi in Rician Fading Channel (K=1) | 0.4366 | 0.3716 | 0.2556 | 0.1533 | 0.0819 | 0.0320 |

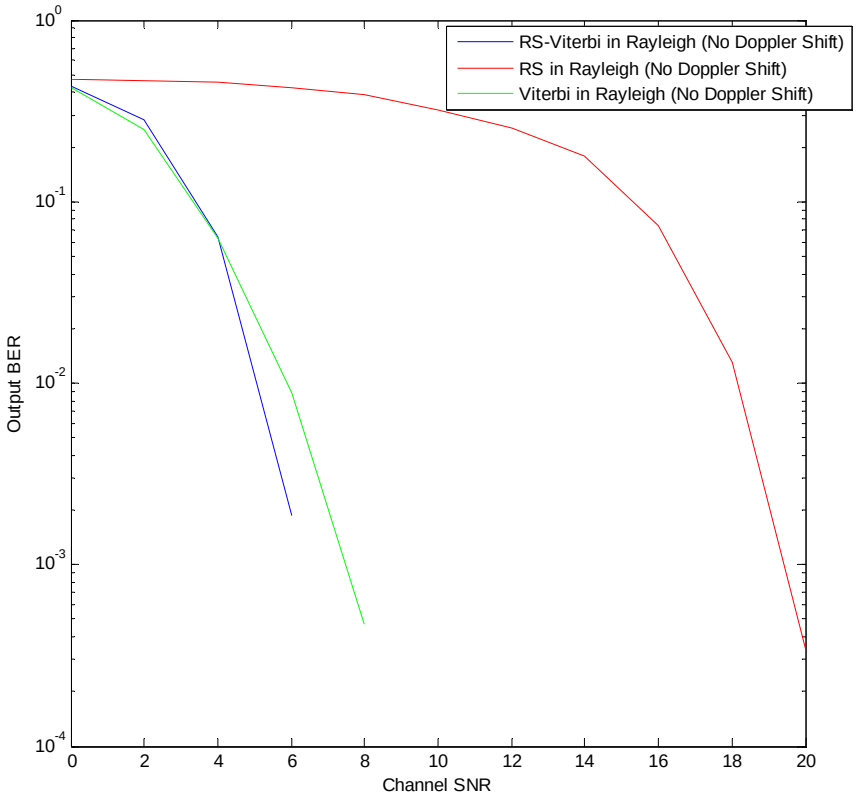


**Σχήμα 5.14** Σύγκριση Απόδοσης (BER) Συναλυσώμενης Reed-Solomon -Viterbi Κωδικοπίησης σε AWGN, Rayleigh και Rician Διαύλους

Στο Σχήμα 5.15 μοντελοποιούμε την επίδοση RS, Viterbi και του συνδυασμού τους RS-Viterbi Concatenated κώδικα παρουσία Rayleigh εξασθένησης. Για οποιαδήποτε τιμή SNR του καναλιού παρατηρούμε πως η πιθανότητα λάθους στην έξοδο του συστήματος είναι μικρότερη στην περίπτωση του Concatenated RS-Viterbi κώδικα, ακολουθεί η επίδοση του Viterbi κώδικα και έπειτα του απλού RS κώδικα.

**Πίνακας 5.15** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rayleigh Δίαυλο χωρίς μετατόπιση Doppler

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 5<br>k = 23<br>noPackets =1000 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| RS- Viterbi in Rayleigh Channel | 0.4326 | 0.2847 | 0.0644 | 0.0019 | 0 | 0 | 0 | 0 | 0 | 0 |
| RS in Rayleigh Channel | 0.4753 | 0.4619 | 0.4555 | 0.4265 | 0.3879 | 0.3213 | 0.2530 | 0.1786 | 0.0735 | 0.0129 |
| Viterbi in Rayleigh Channel | 0.4231 | 0.2500 | 0.0633 | 0.0089 | 0.0005 | 0 | 0 | 0 | 0 | 0 |



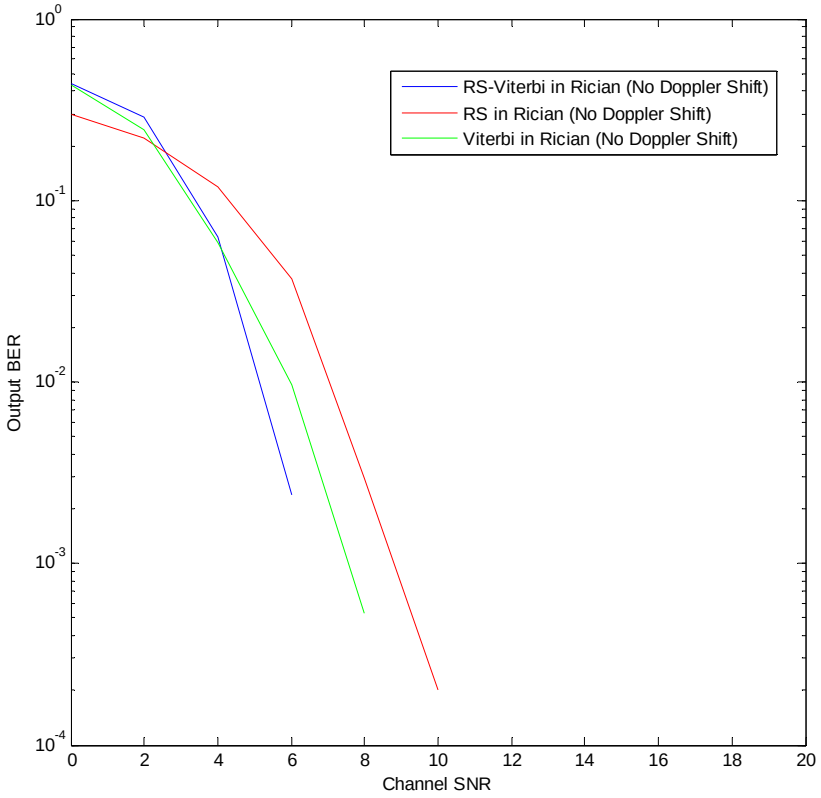

**Σχήμα 5.15** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rayleigh Δίαυλο χωρίς μετατόπιση Doppler

Στο Σχήμα 5.16 μοντελοποιούμε την επίδοση RS, Viterbi και του συνδυασμού τους RS-Viterbi Concatenated κώδικα παρουσία Rician εξασθένησης. Για οποιαδήποτε τιμή SNR του καναλιού παρατηρούμε πως η πιθανότητα λάθους στην έξοδο του συστήματος είναι μικρότερη στην περίπτωση του Concatenated RS-Viterbi κώδικα, ακολουθεί η επίδοση του Viterbi κώδικα και έπειτα του απλού RS κώδικα.

**Πίνακας 5.16** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rician Δίαυλο (K=10) χωρίς μετατόπιση Doppler

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 5<br>k = 23<br>noPackets =1000<br>K=10 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |

| | Output BER | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| RS- Viterbi in Rician Channel | 0.4404 | 0.2866 | 0.0627 | 0.0024 | 0 | 0 | 0 | 0 | 0 | 0 |
| RS in Rician | 0.3001 | 0.2187 | 0.1197 | 0.0373 | 0.0029 | 0.0002 | 0 | 0 | 0 | 0 |
| Viterbi in Rician | 0.4326 | 0.2442 | 0.0586 | 0.0097 | 0.0005 | 0 | 0 | 0 | 0 | 0 |

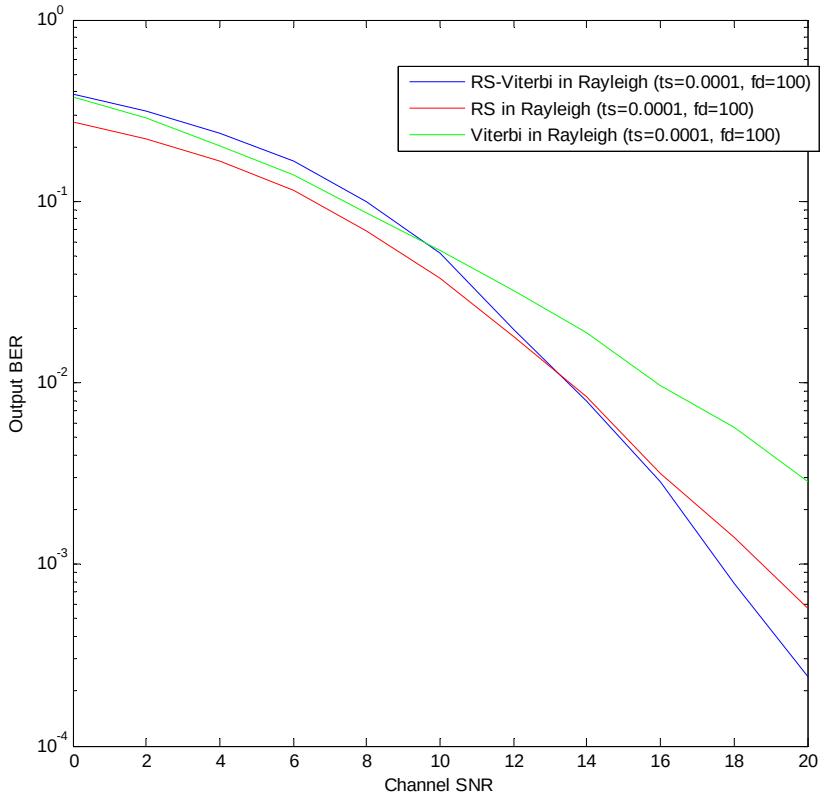



**Σχήμα 5.16** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rician Δίαυλο (K=10) χωρίς μετατόπιση Doppler

**Πίνακας 5.17** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rayleigh Δίαυλο (ts = 0.0001, fd = 100)

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 5 k = 23 noPackets =1000 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |

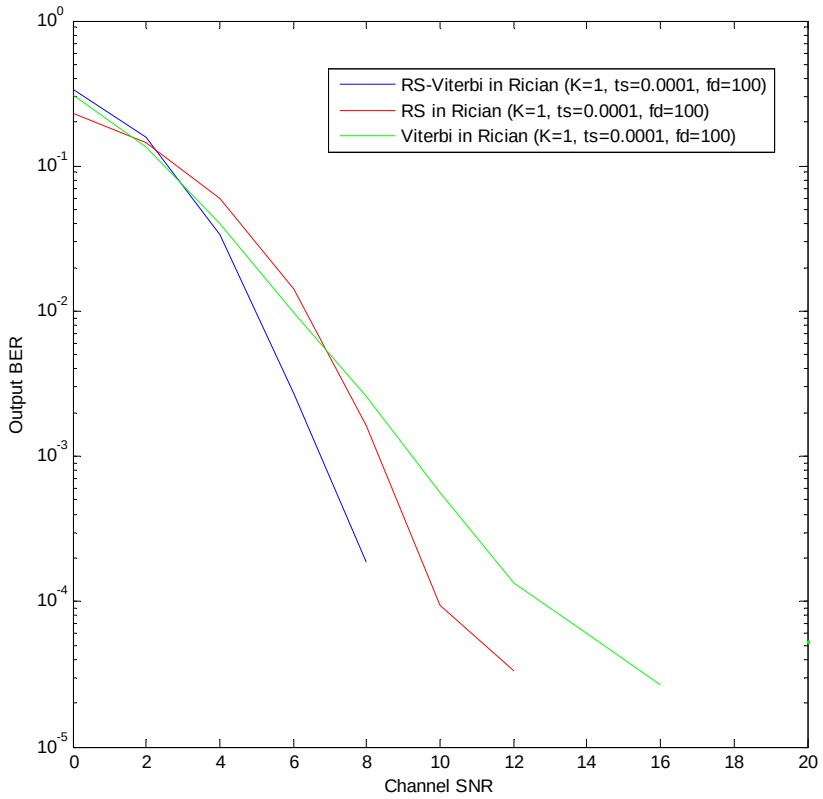
| | Output BER | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| RS- Viterbi in Rayleigh Channel | 0.3890 | 0.3147 | 0.2364 | 0.1659 | 0.0988 | 0.0520 | 0.0197 | 0.0080 | 0.0029 | 0.0008 |
| RS in Rayleigh Channel | 0.2745 | 0.2202 | 0.1656 | 0.1137 | 0.0689 | 0.0374 | 0.0178 | 0.0084 | 0.0032 | 0.0014 |
| Viterbi in Rayleigh Channel | 0.3761 | 0.2856 | 0.2034 | 0.1391 | 0.0867 | 0.0539 | 0.0322 | 0.0190 | 0.0096 | 0.0057 |




**Σχήμα 5.17** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rician Δίαυλο (ts = 0.0001, fd = 100)

**Πίνακας 5.18** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rayleigh Δίαυλο (K=1, ts = 0.0001, fd = 100)

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 5<br>k = 23<br>noPackets =1000<br>K=10 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| RS- Viterbi in Rician Channel | 0.3326 | 0.1557 | 0.0329 | 0.0027 | 0.0001 | 0 | 0 | 0 | 0 | 0 |
| RS in Rician | 0.2286 | 0.1435 | 0.0601 | 0.0143 | 0.0016 | 0.0001 | 0 | 0 | 0 | 0 |
| Viterbi in Rician | 0.3079 | 0.1314 | 0.0414 | 0.0106 | 0.0026 | 0.0007 | 0.0001 | 0.0001 | 0.0000 | 0.0000 |



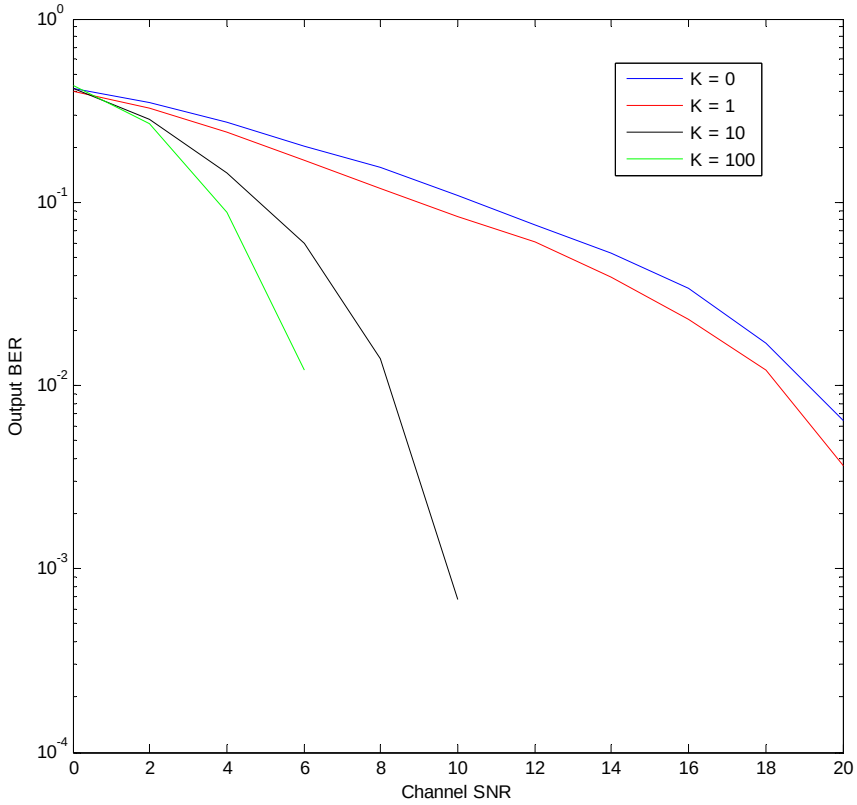

**Σχήμα 5.18** Σύγκριση απόδοσης (BER) συναλυσώμενης Reed-Solomon -Viterbi, RS και Viterbi κωδικοπίησης σε Rician Δίαυλο (K=1, ts = 0.0001, fd = 100)

Με την παρακάτω προσομοίωση εξετάζουμε την επίδοση του Concatenated RS-Viterbi κώδικα υπό Rician κανάλι εξασθένησης για διαφορετικές τιμές της μεταβλητής

K. Για μεγαλύτερες τιμές της μεταβλητής K, έχουμε λιγότερη επίδραση της εξασθένησης στην πιθανότητα λάθους στην έξοδο του σήματος και το κανάλι πλησιάζει τη συμπεριφορά AWGN καναλιού. Αντίθετα, εάν K=0 τότε το Rician κανάλι αντιστοιχεί σε κανάλι Rayleigh με μηδενική άμεση οπτική διαδρομή.

**Πίνακας 5.19** Απόδοση (BER) συναλυσώμενης Reed-Solomon -Viterbi κωδικοπίησης σε Rician Δίαυλο σαν συνάρτηση του παράγοντα K.

| **Coding Data** | **Channel SNR (dB)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| m = 5<br>k = 27<br>Rician<br>Channel | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| | **Output BER** | | | | | | | | | |
| K = 0 | 0. 4178 | 0. 3483 | 0. 2743 | 0. 2030 | 0. 1537 | 0. 1085 | 0.0748 | 0.0529 | 0.0338 | 0.0171 |
| K = 1 | 0. 4044 | 0. 3261 | 0. 2418 | 0. 1685 | 0. 1186 | 0. 0833 | 0. 0605 | 0. 0390 | 0. 0230 | 0. 0121 |
| K = 10 | 0.4175 | 0.2839 | 0.1450 | 0.0592 | 0.0139 | 0.0007 | 0 | 0 | 0 | 0 |
| K = 100 | 0.4324 | 0.2677 | 0.0873 | 0.0121 | 0 | 0 | 0 | 0 | 0 | 0 |




**Σχήμα 5.19** Απόδοση (BER) συναλυσώμενης Reed-Solomon -Viterbi κωδικοπίησης σε Rician Δίαυλο σαν συνάρτηση του παράγοντα K.

## 5.5. Συμπεράσματα των Προσομοιώσεων

Η προγραμματιστική υλοποίηση καναλιών μέσω της βοήθειας του matlab είχε ως σκοπό την ανάλυση της επίδοσης των κωδίκων υπό την παρουσία διαφορετικού περιβάλλοντος. Η εκτίμηση της επίδοσης (BER) των τεχνικών κωδικοποίησης Reed-Solomon και του συνδυασμού αυτού με τον συνελικτικό Viterbi ως συναλυσώμενος κώδικας σύμφωνα με τα παραπάνω γραφήματα μας οδηγούν στα εξής συμπεράσματα:

- Η επίδραση του Reed-Solomon κώδικα βελτιώνει την συμπεριφορά του συστήματος, συγκρίσει με ένα μη κωδικοποιημένο μήνυμα, για όλες σχεδόν τις τιμές του σηματοθορυβικού λόγου του διαύλου. Εξαίρεση στο συμπέρασμα αυτό παρατηρούμε γα πολύ χαμηλές τιμές της ισχύς του σήματος (SNR = 0 – 4 dB), όπου το σύστημα υπερχειλίζεται από λάθη.
- Η απόδοση ποσοστού λάθους του Reed-Solomon κώδικα παρουσία όλων των τύπων διαύλου που μελετάμε (AWGN, Fading) βελτιώνεται με την αύξηση της τιμής της ικανότητας διόρθωσης λαθών του κώδικα. Ισοδύναμα, η επίδοση του κώδικα σε ένα μήνυμα με σταθερό ρυθμό κωδικοποίησης, βελτιώνεται με την αύξηση του αριθμού των bits ανά σύμβολο.
- Όταν ένας χρήστης (ή οι ανακλάσεις στο περιβάλλον του), είναι σε κίνηση, η ταχύτητα της κίνησης προκαλεί μια αλλαγή στην συχνότητα του σήματος που μεταδίδεται κατά μήκος κάθε διαδρομής, φαινόμενο γνωστό ως μετατόπιση Doppler. Στην περίπτωση της "γρήγορης" εξασθένησης (fd*ts ≈ 1) η επίδοση του Reed-Solomon κώδικα υποβαθμίζεται αρκετά, ενώ στην περίπτωση "αργής" εξασθένησης (fd*ts <<1) η επίδοση του κώδικα πλησιάζει την συμπεριφορά συστήματος χωρίς μετατόπιση Doppler.
- Η επίδοση τόσο του Reed-Solomon κώδικα, του συνελικτικού Viterbi, όσο και του συνδυασμού τους, παρουσία απλού προσθετικού θορύβου, βελτιώνεται συγκρίσει παρουσίας εξασθένησης τύπου Rayleigh ή Rician. Επιπλέον, στην περίπτωση της Rician εξασθένησης όπου υπάρχει και άμεση οπτική διαδρομή μετάδοσης του σήματος, παρατηρούμε βελτιωμένη επίδοση των κωδίκων σχετικά με την Rayleigh εξασθένηση. Όσο αυξάνουμε την Rician παράμετρο K, έχουμε λιγότερη επίδραση της εξασθένησης στην έξοδο του σήματος και η επίδοση τους πλησιάζει τη συμπεριφορά απλού προσθετικού θορύβου.
- Ο συνδυασμός του Reed-Solomon με τον Viterbi παρουσία λευκού προσθετικού θορύβου, βελτιώνει την επίδοση του συστήματος σχετικά με την εφαρμογή καθενός από τους κώδικες ξεχωριστά. Εξετάζοντας την περίπτωση διαύλου με εξασθένηση, τόσο στην περίπτωση της Rician εξασθένησης αλλά και εντονότερα παρουσία Rayleigh εξασθένησης ο συναλυσώμενης κώδικας βελτιώνει την επίδοση σε υψηλά SNR αλλά την υποβαθμίζει σε χαμηλά. Το φαινόμενο αυτό είναι ακόμη εντονότερο παρουσία μετατόπισης Doppler. Το αίτιο που επηρεάζει την μείωση της επίδοσης του συναλυσώμενου κώδικα σε χαμηλά SNR έγκειται στην πολύ χαμηλή επίδοση του Viterbi κώδικα σε χαμηλά SNR και περισσότερο στον συνδυασμό χαμηλά SNR με μετατόπιση Doppler.

# 6. Συμπεράσματα – Προτάσεις για Μελλοντική Έρευνα

## 6.1. Συμβολή της εργασίας

Βασικός στόχος της παρούσας εργασίας ήταν η μελέτη και προσομοίωση του Reed-Solomon καθώς και των συνδυασμών του, παρουσίας διάφορων καναλιών, με σκοπό να εξαχθούν ποιοτικά αλλά και ποσοτικά συμπεράσματα για την συμπεριφορά τους. Αναλυτικότερα, σύμφωνα με την δομή της εργασίας, παραθέτω τους τομείς που μελέτησα εκτενώς για την εξαγωγή χρήσιμων συμπερασμάτων.

- Επισκόπηση και κατηγοριοποίηση των τεχνικών κωδικοποίησης καναλιού με έμφαση στον Reed-Solomon κώδικα ως τον πλέον κατάλληλο κώδικα για την αντιμετώπιση πολλαπλών εκρήξεων λαθών και με πολυάριθμες πρακτικές εφαρμογές.
- Κέρδος κωδικοποίησης, διόρθωση λαθών και βελτιωμένη πιθανότητα μετάδοσης λανθασμένης πληροφορίας κατά την διέλευση της από το δίαυλο με παράλληλη αναφορά στο βάρος της εισαγωγής κώδικα με έμφαση στην απαίτηση μεγαλύτερου εύρους ζώνης.
- Βελτιωμένη απόδοση συστήματος με την εφαρμογή σειριακής συναλυσώμενης κωδικοποίησης. Συγκεκριμένα το αποτέλεσμα του συνδυασμού ενός εξωτερικού Reed-Solomon κώδικα, μέσω ενδιάμεσων interleaving, και ενός εσωτερικού δυαδικού συνελικτικού κώδικα, είναι πολύ ισχυρό και εξηγεί την χρήση του συστήματος σε πολλές εφαρμογές, που κυμαίνονται από το χώρο των επικοινωνιών ως την ψηφιακή μετάδοση υψηλής ευκρίνειας τηλεόραση.
- Οι Reed-Solomon κώδικες, έχουν ένα ευρύ φάσμα εφαρμογών στους τομείς των ψηφιακών επικοινωνιών και αποθήκευσης. Ανάλυση της προσφοράς του Reed-Solomon σε κάθε εφαρμογή καθώς και γνωστοποίηση της χρησιμότητας και προσφοράς των εφαρμογών στα σύγχρονα συστήματα επικοινωνίας.
- Συγκριτική μελέτη τύπων διαύλου, με έμφαση στους διαύλους που απαντώνται στις κινητές και δορυφορικές επικοινωνίες (AWGN, Rayleigh, Rice).
- Παρουσίαση της σχεδιαστικής πορείας για την ανάπτυξη προσομοιωτή. Τεκμηρίωση του Matlab ως σημαντικού προσομοιωτικού εργαλείου και εξαγωγή συμπερασμάτων προσομοίωσης που επαληθεύουν τα θεωρητικά δεδομένα περί της απόδοσης του κώδικα υπό διαφορετικό εξωτερικό περιβάλλον και διαφορετικές εσωτερικές παραμέτρους του κώδικα.

## *6.2. Προτάσεις για Μελλοντική Έρευνα*

Ολοκληρώνοντας την παρούσα διπλωματική εργασία, παρουσιάζουμε συνοπτικά ορισμένα θέματα προς μελλοντική έρευνα στα πλαίσια νέων εργασιών. Η λίστα που παραθέτουμε είναι ενδεικτική και όχι εξαντλητική.

- Εκτίμηση της επίδοσης των διαφόρων τεχνικών κωδικοποίησης που μας απασχόλησαν στα πλαίσια της εργασίας λαμβάνοντας υπόψη ρεαλιστικότερα μοντέλο διάδοσης (π.χ. με χρήση του Winner Channel Model)
- Εκτίμηση της επίδοσης των διαφόρων τεχνικών κωδικοποίησης χρησιμοποιώντας ρεαλιστικότερες παραμέτρους λειτουργίας, όπως ρυθμό κωδικοποίησης, μήκος λέξης κ.ο.κ. Ανάλογα με το εκάστοτε σύστημα στο οποίο θα εφαρμοστούν οι τεχνικές κωδικοποίησης (DVB, WiMAX, LTE) θα ληφθούν οι κατάλληλες τιμές για τις παραπάνω παραμέτρους.
- Μελέτη και αξιολόγηση τεχνικών Interleaving σε συνδυασμό με διάφορα σχήματα κωδικοποίησης και διαμόρφωσης.
- Επέκταση των τεχνικών κωδικοποίησης για συστήματα πολλαπλών φερόντων OFDM ή/και πολλαπλών κεραιοσυστημάτων MIMO
- Μελέτη και αξιολόγηση τεχνικών concatenating κωδικοποίησης λαμβάνοντας υπόψη διάφορους συνδυασμούς block και συνελικτικών κωδίκων (RS, Viterbi, LDPC, Turbo,….)
- Εκτίμηση επίδοσης διαφόρων τεχνικών με χρήση αναλυτικών εκφράσεων και σύγκριση των αποτελεσμάτων με αντίστοιχες προσομοιώσεις (Monte Carlo) .

# ΒΙΒΛΙΟΓΡΑΦΙΑ & ΑΝΑΦΟΡΕΣ


[1] Sklar, B., “Digital Communications, Fundamentals and Applications,” Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] Moreira, J. C. and Farrell, P. G., “Essentials of Error-Control Coding,” John Wiley & Sons, Ltd, 2006.

[3] Todd K. Moon, “Error Correction Coding, Mathematical Methods and Algorithms,” A John Wiley and Sons, 2005.

[4] Christian B Schlegel and Lance C. Perez, “Trellis and Turbo Coding,” IEEE Press, a John Wiley and Sons, 2004.

[5] Tranter, W., Shanmugan, K., Rappaport, T. and Kosbar, K., “Principles of Communication Systems Simulation with Wireless Applications, ” Prentice Hall, 2004

[6] Proakis, J. G. and Manolakis, D. G., “Digital Signal Processing-Principles, Algorithms & applications,” Prentice Hall, third edition, 1996.

[7] Vinary K. Ingle and John J. Proakis, "Digital Signal Processing Using Matlab V4," bookware Companion Series, PWS, 1997.

[8] Finite Field Arithmetic, channel analysis, http://en.wikipedia.org/wiki/

[9] Proakis, J. G., "Digital Communications, " 4th edition, 2005

[10] Robert H. Morelos_Zaragoza, "The Art of Error Correcting Coding," John Wiley And Sons,Ltd., 2002

[11] Digital Video Broadcasting, , http://en.wikipedia.org/wiki/Digital_Video_Broadcasting

[12] WiMAX e-book, http://www.scribd.com/doc/3191854/wimax-ch3

# ΠΑΡΑΡΤΗΜΑΤΑ: Κώδικας Matlab

### 1. *RS in AWGN Channel (Input: SNR) – Αρχείο RSInAWGN.m*

```
clear all;
m=input('number of bits per symbol "m" :')
    % Number of bits per symbol
k=input('number of symbols per word "k" :')
    %Word length (number of symbols) before coding
n=2^m-1
    %Word length (number of symbols) after coding
noPackets=input('noPackets :')
    % Number of words to process
errorcapability=fix((n-k)/2)
messageColumn = randint(noPackets*k*m,1)
    % random Sequence of 0,1
message=reshape(messageColumn,noPackets*k,m)
symbolMessage=bi2de(message,'left-msb')
     % converts it to decimal system
words=reshape(symbolMessage,noPackets,k)
    %gf is matlab function (all operations in galoi field)
words=gf(words,m)
code = rsenc(words,n,k)
    %rsenc is matlab function
    %words-->uncoded words
    %code-->coded words
dcode = double(code.x)
    %converts to double system in order operations be available
dcodebi = de2bi(dcode,'left-msb')
    % Convert integers to bits.
dcodebi = reshape(dcodebi,noPackets*n*m,1)
words=double(words.x)


                %channel
M = 2;    % DPSK signal
dpskSig = dpskmod(dcodebi,M)
```

```
metritis = 1
for SNR = 0:2:20
    % Range of SNR values, in dB.
    rxSig = awgn(dpskSig,SNR,'measured',[], 'dB');
        % Add Gaussian noise.
    rx =dpskdemod(rxSig,M);
        % Demodulate.
        % Ignore first sample because of DPSK initial condition.

    rxMessage=reshape(rx,noPackets*n,m)
        % the received message to the decoder (after demodulation)
    rxSymbolMessage=bi2de(rxMessage,'left-msb')
        % converts it to decimal system
    rxWords=reshape(rxSymbolMessage,noPackets,n)
    rxWords=gf(rxWords,m)
        % decoding operations in galoi field

    [dec,cnumerr] = rsdec(rxWords,n,k)
        %rsdec is matlab function
    dec=double(dec.x)
        %dec-->decoded words
    z = de2bi(words,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    z = reshape(z.',prod(size(z)),1);

    x = de2bi(dec,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    x = reshape(x.',prod(size(x)),1);

    [number_of_errors,bit_error_rate] = biterr(z,x)
        % biterr is matlab function, counts different bits between
        % sequences(z,x))

     output_bit_error_prob(metritis)=bit_error_rate
    metritis = metritis + 1;
end
SNR = 0:2:20
semilogy(SNR,output_bit_error_prob)
```

## 2. *RS in AWGN Channel (Input: BER) - Αρχείο RSInAWGN_BER.m*

```
clear all;
m=input('number of bits per symbol "m" :')
    % Number of bits per symbol
k=input('number of symbols per word "k" :')
    %Word length (number of symbols) before coding
n=2^m-1
    %Word length (number of symbols) after coding
noPackets=input('noPackets :')
    % Number of words to process
errorcapability=fix((n-k)/2)
```

```
messageColumn = randint(noPackets*k*m,1)
    % random Sequence of 0,1
message=reshape(messageColumn,noPackets*k,m)
symbolMessage=bi2de(message,'left-msb')
     % converts it to decimal system
words=reshape(symbolMessage,noPackets,k)
    %gf is matlab function (all operations in galoi field)
words=gf(words,m)
code = rsenc(words,n,k)
    %rsenc is matlab function
    %words-->uncoded words
    %code-->coded words
words=double(words.x)
    %converts to double system in order operations be available




metritis=1
                %channel
for channel_bit_error_prob=[0.00001  0.00003  0.0001  0.0003  0.001  0.003
0.01 0.03]

    noChannelBitErrors=fix(channel_bit_error_prob*n*m*noPackets)
    errorWords=randerr(1,n*m*noPackets,noChannelBitErrors)
    biErrorWords=reshape(errorWords,n*noPackets,m)
    errorWords=bi2de(biErrorWords,'left-msb')
    errorWords=reshape(errorWords,noPackets,n)
    errorWords=gf(errorWords,m)
        %addition in galoi field (coded words + errorWords)
    receivedWords=errorWords+code

    t=errorcapability
        %rdec is matlab function
    [dec,cnumerr] = rsdec(receivedWords,n,k)
        %dec-->decoded words
    dec=double(dec.x)

    z = de2bi(words,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    z = reshape(z.',prod(size(z)),1);

    x = de2bi(dec,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    x = reshape(x.',prod(size(x)),1);

    [number_of_errors,bit_error_rate] = biterr(z,x)
        % biterr is matlab function, counts different bits between
        % sequences(z,x))

    output_bit_error_prob(metritis)=bit_error_rate
    metritis=metritis+1

end
```

```
channel_bit_error_prob=[0.00001  0.00003  0.0001  0.0003  0.001  0.003  0.01
0.03]
semilogy(channel_bit_error_prob,output_bit_error_prob,'g')
```

### 3. *RS in Fading Channel (Input: SNR)*

#### *a. Rayleigh Channel – Αρχείο RSInRayleigh.m*

```
clear all;
m=input('number of bits per symbol "m" :')
    % Number of bits per symbol
k=input('number of symbols per word "k" :')
    %Word length (number of symbols) before coding
n=2^m-1
    %Word length (number of symbols) after coding
noPackets=input('noPackets :')
    % Number of words to process
errorcapability=fix((n-k)/2)
messageColumn = randint(noPackets*k*m,1)
    % random Sequence of 0,1
message=reshape(messageColumn,noPackets*k,m)
symbolMessage=bi2de(message,'left-msb')
     % converts it to decimal system
words=reshape(symbolMessage,noPackets,k)
    %gf is matlab function (all operations in galoi field)
words=gf(words,m)
code = rsenc(words,n,k)
    %rsenc is matlab function
    %words-->uncoded words
    %code-->coded words
dcode = double(code.x)
    %converts to double system in order operations be available
dcodebi = de2bi(dcode,'left-msb')
    % Convert integers to bits.
dcodebi = reshape(dcodebi,noPackets*n*m,1)
words=double(words.x)


                %channel

    % DBPSK modulation
chan = rayleighchan(fd,ts);
    % Generate data and apply fading channel.
M = 2; % DBPSK modulation order
dpskSig = dpskmod(dcodebi,M)  % DPSK signal
fadedSig = filter(chan,dpskSig) % Effect of channel


SNR = 0:2:20; % Range of SNR values, in dB.
for metritis = 1:length(SNR)
```

```
        % Add Gaussian noise.
    rxSig = awgn(fadedSig,SNR(metritis));
        % Demodulate.
    rx = dpskdemod(rxSig,M);
    rxMessage=reshape(rx,noPackets*n,m)
        % the received message to the decoder (after demodulation)
    rxSymbolMessage=bi2de(rxMessage,'left-msb')
        % converts it to decimal system
    rxWords=reshape(rxSymbolMessage,noPackets,n)
    rxWords=gf(rxWords,m)
        % decoding operations in galoi field

    [dec,cnumerr] = rsdec(rxWords,n,k)
        %rsdec is matlab function
        %dec-->decoded words
    dec=double(dec.x)

    z = de2bi(words,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    z = reshape(z.',prod(size(z)),1);

    x = de2bi(dec,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    x = reshape(x.',prod(size(x)),1);

    [number_of_errors,bit_error_rate] = biterr(z,x)
        % biterr is matlab function, counts different bits between
        % sequences(z,x))

     output_bit_error_prob(metritis)=bit_error_rate


end

SNR = 0:2:20
semilogy(SNR,output_bit_error_prob)
```

### b. *Rician Channel – Αρχείο RSInRician.m*

```
chan = ricianchan(fd,ts,K);
```

## 4. *Viterbi in AWGN Channel (Input: SNR)*

```
clear all;
m=input('number of bits per symbol "m" :')
    % Number of bits per symbol
k=input('number of symbols per word "k" :')
    %Word length (number of symbols) before coding
```

```
n=2^m-1
    %Word length (number of symbols) after coding
noPackets=input('noPackets :')
    % Number of words to process
errorcapability=fix((n-k)/2)
messageColumn = randint(noPackets*k*m,1)
    % random Sequence of 0,1
M = 2;

                    %  Viterbi

msg = messageColumn; % Random data
t = poly2trellis(7,[171 133 171 133]); % Define trellis.
code2 = convenc(msg,t); % Encode the data.

metritis = 1;
for SNR = 0:2:20
    dpskSig = dpskmod(code2,M)
    ncode = awgn(dpskSig,SNR,'measured',244)
    ncode =dpskdemod(ncode,M);
        % Quantize to prepare for soft-decision decoding.
    qcode = quantiz(ncode,[0.001,.1,.3,.5,.7,.9,.999]);

    tblen = 2; delay = tblen; % Traceback length
    decoded = vitdec(qcode,t,tblen,'cont','soft',3); % Decode.

        % Compute bit error rate.
    [number,ratio] = biterr(decoded(delay+1:end),msg(1:end-delay))
    output_bit_error_prob(metritis)=ratio
    metritis = metritis + 1;
end
SNR = 0:2:20
semilogy(SNR,output_bit_error_prob)
```

## 5. *RS-Viterbi in AWGN Channel (Input: SNR) - Αρχείο RS_ViterbiInAWGN.m*

```
clear all;
m=input('number of bits per symbol "m" :')
    % Number of bits per symbol
k=input('number of symbols per word "k" :')
    %Word length (number of symbols) before coding
n=2^m-1
    %Word length (number of symbols) after coding
noPackets=input('noPackets :')
    % Number of words to process
errorcapability=fix((n-k)/2)
messageColumn = randint(noPackets*k*m,1)
    % random Sequence of 0,1
message=reshape(messageColumn,noPackets*k,m)
symbolMessage=bi2de(message,'left-msb')
     % converts it to decimal system
words=reshape(symbolMessage,noPackets,k)
    %gf is matlab function (all operations in galoi field)
```

```
words=gf(words,m)
code = rsenc(words,n,k)
    %rsenc is matlab function
    %words-->uncoded words
    %code-->coded words
dcode = double(code.x)
    %converts to double system in order operations be available
dcodebi = de2bi(dcode,'left-msb')
    % Convert integers to bits.
dcodebi = reshape(dcodebi,noPackets*n*m,1)
words=double(words.x)


                    %  Viterbi

msg = dcodebi; %  data
t = poly2trellis(7,[171 133 171 133]); % Define trellis.
code2 = convenc(msg,t); % Covolutional encoding

metritis = 1;
for SNR = 0:1:7
    M = 2;
    dpskSig = pskmod(code2,M); % Modulate.
    ncode = awgn(dpskSig,SNR,'measured',244)
    ncode =pskdemod(ncode,M); % Demodulate.
    qcode = quantiz(ncode,[0.001,.1,.3,.5,.7,.9,.999]);
        % Quantize to prepare for soft-decision decoding.

    tblen = 2; delay = tblen; % Traceback length
    decoded = vitdec(qcode,t,tblen,'cont','soft',3); % Decode.

    decoded(1:end-delay)=decoded(delay+1:end)
        %              Default:              [number,ratio]              =
biterr(decoded(delay+1:end),msg(1:end-delay))

    rxMessage=reshape(decoded,noPackets*n,m)
    rxSymbolMessage=bi2de(rxMessage,'left-msb')
        % message-->converts it to decimal system
    rxWords=reshape(rxSymbolMessage,noPackets,n)
    rxWords=gf(rxWords,m)
        % RS decoding Operations done in galoi field

    [dec,cnumerr] = rsdec(rxWords,n,k)
        %rsdec is matlab function
        %dec-->decoded words
    dec=double(dec.x)


    z = de2bi(words,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    z = reshape(z.',prod(size(z)),1);

    x = de2bi(dec,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    x = reshape(x.',prod(size(x)),1);
```

```
    [number_of_errors,bit_error_rate] = biterr(z,x)
        % biterr is matlab function, counts different bits between
        % sequences(z,x))

     output_bit_error_prob(metritis)=bit_error_rate
    metritis = metritis + 1;
end
SNR = 0:1:7
semilogy(SNR,output_bit_error_prob)
```

## 6. *RS-Viterbi in fading Channel (Input: SNR)*

### *c. Rayleigh Channel - Αρχείο RS_ViterbiInRayleigh.m*

```
clear all;
m=input('number of bits per symbol "m" :')
    % Number of bits per symbol
k=input('number of symbols per word "k" :')
    %Word length (number of symbols) before coding
n=2^m-1
    %Word length (number of symbols) after coding
noPackets=input('noPackets :')
    % Number of words to process
errorcapability=fix((n-k)/2)
messageColumn = randint(noPackets*k*m,1)
    % random Sequence of 0,1
message=reshape(messageColumn,noPackets*k,m)
symbolMessage=bi2de(message,'left-msb')
     % converts it to decimal system
words=reshape(symbolMessage,noPackets,k)
    %gf is matlab function (all operations in galoi field)
words=gf(words,m)
code = rsenc(words,n,k)
    %rsenc is matlab function
    %words-->uncoded words
    %code-->coded words
dcode = double(code.x)
    %converts to double system in order operations be available
dcodebi = de2bi(dcode,'left-msb')
    % Convert integers to bits.
dcodebi = reshape(dcodebi,noPackets*n*m,1)
words=double(words.x)

chan = rayleighchan(fd,ts);
    % Rayleigh Channel, Matlab Function


                    %  Viterbi
```

```
msg = dcodebi; %  data
t = poly2trellis(7,[171 133 171 133]); % Define trellis.
code2 = convenc(msg,t); % Covolutional encoding

metritis = 1;
for SNR = 0:2:20
    M = 2;
    dpskSig = dpskmod(code2,M); % Modulate.

    fadedSig = filter(chan,dpskSig) % Effect of channel

    ncode = awgn(fadedSig,SNR,'measured',244)
    ncode =dpskdemod(ncode,M); % Demodulate.
    qcode = quantiz(ncode,[0.001,.1,.3,.5,.7,.9,.999]);
        % Quantize to prepare for soft-decision decoding.

    tblen = 2; delay = tblen; % Traceback length
    decoded  =  vitdec(qcode,t,tblen,'cont','soft',3);  %  Convolutional
Decoding.

    decoded(1:end-delay)=decoded(delay+1:end)
          %             Default:             [number,ratio]             =
biterr(decoded(delay+1:end),msg(1:end-delay))


    rxMessage=reshape(decoded,noPackets*n,m)
    rxSymbolMessage=bi2de(rxMessage,'left-msb')
        % converts it to decimal system
    rxWords=reshape(rxSymbolMessage,noPackets,n)
    rxWords=gf(rxWords,m)
         % RS decoding Operations done in galoi field

    [dec,cnumerr] = rsdec(rxWords,n,k)
        % rsdec is matlab function
        %dec-->decoded words
    dec=double(dec.x)


    z = de2bi(words,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    z = reshape(z.',prod(size(z)),1);

    x = de2bi(dec,'left-msb'); % Convert integers to bits.
        % Convert z from a matrix to a vector.
    x = reshape(x.',prod(size(x)),1);

    [number_of_errors,bit_error_rate] = biterr(z,x)
        % biterr is matlab function, counts different bits between
        % sequences(z,x))

     output_bit_error_prob(metritis)=bit_error_rate
    metritis = metritis + 1;
end
SNR = 0:2:20
```

```
semilogy(SNR,output_bit_error_prob,'r')
```

### d. *Rician Channel – Αρχείο RS_ViterbiInRician.m*

```
chan = ricianchan(fd,ts,K);
```

## 7. *RS code in AWGN (Code is developed without use of Default Matlab functions in Coding/decoding procedures)*

```
clear all;
m=input('number of bits per symbol "m" ')
k=input('number of symbols per word "k" ')
n=2^m-1
    % αριθμός συμβόλων στην τυχαία ακολουθία bits
noPackets=input('noPackets ')
errorcapability=fix((n-k)/2)

message=randint(noPackets*k,m)
    %τυχαία ακολουθία από 0,1 με κάθε σειρά να αποτελεί ένα σύμβολο

symbolMessage=bi2de(message,'left-msb')
words=reshape(symbolMessage,noPackets,k)
preCodeWords=zeros(noPackets,n)
preCodeWords(:,1:k)=words
    % den einai akoma gf



                          %kwdikopoiisi


g=rsgenpoly(n,k)
    %Generator polynomial για Reed-Solomon κώδικα
preCodeWords=gf(preCodeWords,m)
p=gf(zeros(noPackets,n),m)
for i=1:noPackets
    [x,p(i,:)]=deconv(preCodeWords(i,:),g)
end
codeWords=p+preCodeWords

codeWords=double(codeWords.x)
biCodeWords=de2bi(codeWords,'left-msb')
    %  ο πίνακας μετατρέπεται σε στήλη
codeWords=gf(codeWords,m)


 %επαλήθευση πως u=m*g, οι ρίζες του g μηδενίζουν το δια/σμα u

 %z=gf(zeros(n-k,noPackets),m)
```

```
 %for i=1:noPackets
 %    for j=1:n-k
 %      z(j,i)=polyval(codeWords(i,:),gf(2,m)^j)
 %    end
 %end


                            %channel

metritis=1

for  SNR=0:20

    t=errorcapability

    biCodeWords=2*biCodeWords-1
    biReceivedWords=awgn(biCodeWords,SNR)
    for i=1:n*noPackets
        for j=1:m
            if biReceivedWords(i,j)>0
                biReceivedWords(i,j)=1
                else
                biReceivedWords(i,j)=0
            end
        end
    end

    receivedWords=bi2de(biReceivedWords,'left-msb')
    receivedWords=reshape(receivedWords,noPackets,n)
    receivedWords=gf(receivedWords,m)
    errorWords=codeWords+receivedWords


                %Υπολογισμός Συνδρόμου

    S1=gf(zeros(n-k,noPackets),m)
    S2=gf(zeros(n-k,noPackets),m)

    for i=1:noPackets
        for j=1:n-k
            S1(j,i)=polyval(receivedWords(i,:),gf(2,m)^j)
            S2(j,i)=polyval(errorWords(i,:),gf(2,m)^j)
        end
    end


     % Θέσεις λαθών και Τιμές λαθών

    deCodeWords=gf(zeros(noPackets,n),m)
    lathosWord=gf(zeros(1,n),m)
    lathosWords=gf(zeros(noPackets,n),m)


    for i=1:noPackets
```

```
v=t

if S1(:,i)==0
    lathosWord(1,:)=0
    else


        % Υπολογισμός των λαθών για τα μη μηδενικά σύμβολα

    A=ypologismosA(v,m,S1,i)
    while (det(A)==0) && (v>1)
        v=v-1
        A=ypologismosA(v,m,S1,i)
    end

    if (det(A)==0) && (v==1)
        lathosWord(1,:)=0
        else


        B=gf(zeros(v,1),m)
        L=gf(zeros(v,1),m)
        for f=1:v
            B(f)=S1(t+f,i)
        end

        L=inv(A)*B
        L=L'
        Lpoly=[L,gf(2,m)^0]

        noLathous=0
        thesiLathous=zeros(1,n)
        for h=1:n
            if polyval(Lpoly,gf(2,m)^h)==0
                noLathous=noLathous+1
                thesiLathous(1,noLathous)=n-h
            end
        end


        if noLathous==0
            lathosWord(1,:)=0
            else

            C=gf(zeros(noLathous,noLathous),m)
            D=gf(zeros(noLathous,1),m)
            syntelestesLathous=gf(zeros(noLathous,1),m)

            for y=1:noLathous
                for u=1:noLathous
                    C(y,u)=gf(2,m)^(thesiLathous(1,u)*y)
                    D(y)=S1(y,i)
                end
            end
```

```
                    syntelestesLathous=inv(C)*D
                    for l=1:noLathous
                        lathosWord(1,n-
thesiLathous(1,l))=syntelestesLathous(l,1)
                    end
                end
            end
        end
        lathosWords(i,:)=lathosWord(1,:)
        deCodeWords(i,:)=lathosWords(i,:)+receivedWords(i,:)

    end
    fDeCodeWords=deCodeWords(:,1:k)
    fDeCodeWords=double(fDeCodeWords.x)
    fCodeWords=codeWords(:,1:k)
    fCodeWords=double(fCodeWords.x)

    z = de2bi(fDeCodeWords,'left-msb'); % Convert integers to bits.
    % Convert z from a matrix to a vector.
    z = reshape(z.',prod(size(z)),1);

    x = de2bi(fCodeWords,'left-msb'); % Convert integers to bits.
    % Convert z from a matrix to a vector.
    x = reshape(x.',prod(size(x)),1);

    [number_of_errors,bit_error_rate] = biterr(z,x)

    metritis=metritis+1
    output_bit_error_prob(metritis-1)=bit_error_rate
    output_bit_error_prob(metritis)=0

 end


 SNR=0:20
 output_bit_error_prob=output_bit_error_prob(1:metritis-1)
 semilogy(SNR,output_bit_error_prob)



%ypologismos tou pinaka A

function A=ypologismosA(v,m,S1,i)
A=gf(zeros(v,v),m)
for j=1:v
      for k=1:v
            A(j,k)=S1(k+j-1,i)
      end
end
```