



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

**Τεχνικές Κωδικοποίησης Καναλιού με Έμφαση στους
Συγκεραστικούς και στους Τούρμπο Κώδικες**

Αλέξανδρος Μ. Κασιώτης

ΑΘΗΝΑ
ΦΕΒΡΟΥΑΡΙΟΣ 2012

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Τεχνικές Κωδικοποίησης Καναλιού με Έμφαση στους Συγκεραστικούς και στους Τούρμπο Κώδικες

Αλέξανδρος Μ. Κασιώτης

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Νικόλαος Καλουπτσίδης, Καθηγητής ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Νικόλαος Καλουπτσίδης, Καθηγητής ΕΚΠΑ

Σέργιος Θεοδωρίδης, Καθηγητής ΕΚΠΑ

Θωμάς Σφηκόπουλος, Καθηγητής ΕΚΠΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

**Νικόλαος Καλουπτσίδης
Καθηγητής ΕΚΠΑ**

**Σέργιος Θεοδωρίδης
Καθηγητής ΕΚΠΑ**

**Θωμάς Σφηκόπουλος
Καθηγητής ΕΚΠΑ**

**Λάζαρος Μεράκος
Καθηγητής ΕΚΠΑ**

**Ιωάννης Σταυρακάκης
Καθηγητής ΕΚΠΑ**

**Εμμανουήλ Σαγκριώτης
Αναπληρωτής Καθηγητής ΕΚΠΑ**

**Νικόλαος Κολοκοτρώνης
Λέκτορας Παν/μίου Πελοποννήσου**

Ημερομηνία Εξέτασης: 20/02/2012

ΠΕΡΙΛΗΨΗ

Η κωδικοποίηση καναλιού αποτελεί απαραίτητη προϋπόθεση για την αξιόπιστη μετάδοση πληροφορίας σε οποιοδήποτε ψηφιακό σύστημα επικοινωνιών. Στα σύγχρονα συστήματα, οι απαιτήσεις από τον κώδικα καναλιού είναι ιδιαίτερα αυξημένες όσον αφορά την απόδοση, την πολυπλοκότητα, αλλά και την προσαρμοστική του ικανότητα. Ένα από τα ευρέως χρησιμοποιούμενα σχήματα κωδικοποίησης είναι οι συγκεραστικοί κώδικες, εξαιτίας του διαγράμματος trellis και των αποδοτικών αλγορίθμων αποκωδικοποίησης, αλλά και γιατί αποτελούν δομικό στοιχείο ισχυρότερων σχημάτων κωδικοποίησης, όπως οι κώδικες τούρμπο. Η υπολογιστική πολυπλοκότητα της αποκωδικοποίησης μεγίστης πιθανοφάνειας ενός συγκεραστικού κώδικα είναι ανάλογη της πολυπλοκότητας του διαγράμματος trellis που χρησιμοποιείται.

Στην παρούσα διατριβή προτείνεται η σχεδίαση συγκεκριμένης κατηγορίας κωδίκων χαμηλής πολυπλοκότητας με κατάλληλη προσαρμογή του διαγράμματος trellis των διάτρητων συγκεραστικών κωδίκων (punctured convolutional codes). Στόχος είναι η βελτίωση της απόδοσης, με λογική αύξηση της πολυπλοκότητας του διαγράμματος trellis. Στα πλαίσια της έρευνας παρέχεται ένας ικανοποιητικός αριθμός νέων κωδίκων διαφόρων ρυθμών και τιμών πολυπλοκότητας. Σε πολλές περιπτώσεις διαπιστώνεται πως ελάχιστη αύξηση της πολυπλοκότητας μπορεί να οδηγήσει σε μεγάλη βελτίωση της απόδοσης, συγκριτικά με τους διάτρητους συγκεραστικούς κώδικες.

Παρουσιάζεται επίσης μια μέθοδος σχεδίασης νέων ευέλικτων συγκεραστικών κωδίκων, συνδυάζοντας τις τεχνικές της απαλοιφής μονοπατιών του διαγράμματος trellis (path pruning technique) και της απαλοιφής κωδικών bit (puncturing technique). Οι νέοι κώδικες μπορούν να μεταβάλλουν το ρυθμό τους και την πολυπλοκότητα του διαγράμματος trellis, και κατ'επέκταση την υπολογιστική πολυπλοκότητα της διαδικασίας αποκωδικοποίησης, οδηγώντας σε σχήματα κωδικοποίησης που κάνουν αποδοτικότερη διαχείριση των πόρων του συστήματος, εν συγκρίσει με τους κλασικούς συγκεραστικούς κώδικες μεταβλητού ρυθμού.

Η πολυπλοκότητα του διαγράμματος trellis επηρεάζει την υπολογιστική πολυπλοκότητα οποιουδήποτε αλγόριθμου αποκωδικοποίησης βασίζεται σε αυτό, όπως για παράδειγμα ο αλγόριθμος BCJR και οι παραλλαγές του. Γι' αυτό το λόγο, εξετάζεται η δυνατότητα εφαρμογής των προαναφερθέντων αποτελεσμάτων χρησιμοποιώντας αναδρομικούς συγκεραστικούς κωδικοποιητές, οι οποίοι είναι κατάλληλοι ως περιεχόμενοι κωδικοποιητές των τούρμπο κωδίκων. Στόχος είναι η σχεδίαση αποδοτικών ευέλικτων τούρμπο σχημάτων κωδικοποίησης. Προσομοιώσεις δείχνουν ότι σε συγκεκριμένες περιοχές τιμών του σηματοθορυβικού λόγου, μια σημαντική μείωση της υπολογιστικής πολυπλοκότητας της αποκωδικοποίησης μπορεί ακόμα και να μειώσει το ρυθμό εσφαλμένων bit (bit error rate).

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Κωδικοποίηση Καναλιού

ΛΕΞΕΙΣ-ΚΛΕΙΔΙΑ: αποκωδικοποίηση μέγιστης πιθανοφάνειας, διάγραμμα trellis, ελάχιστη trellis πολυπλοκότητα, συγκεραστικοί κώδικες, τούρμπο κώδικες.

ABSTRACT

Channel coding is a prerequisite for the reliable transmission of information in any digital communications system. Modern systems have increased demands from the channel code, concerning performance, complexity and adaptability. Convolutional codes constitute a widely used channel coding scheme, due to their trellis diagram and their efficient decoding algorithms, as well as because they constitute building blocks of more complicated and robust channel codes, like turbo codes. The computational complexity of the maximum likelihood decoding procedure of a convolutional code is proportional to the complexity of the corresponding trellis diagram.

In this thesis, a family of low complexity convolutional codes is constructed, by modifying appropriately the trellis diagram of punctured convolutional codes. The goal is to improve performance at the expense of a reasonable low increase of the trellis complexity. Many new convolutional codes of various code rates and values of complexity are provided. In many cases, a small increase in complexity can lead to a great improvement of performance, compared to punctured convolutional codes.

Furthermore, a method is presented for designing new flexible convolutional codes, by combining the techniques of path pruning and puncturing. The new codes can vary their rate, as well as the complexity of their trellis diagram, and hence the computational complexity of the decoding algorithm, leading to coding schemes that manage more efficiently the system resources, compared to classical variable rate convolutional codes.

The complexity of the trellis diagram affects the computational complexity of every trellis based decoding algorithm, like BCJR and its variations. Thus, the possibility of applying the aforementioned results using recursive convolutional encoders, which are used as constituent encoders in turbo codes, is investigated. The goal is to construct efficient flexible turbo coding schemes. Simulation results indicate that in specific ranges of the signal to noise ratio, a great decrease in the computational complexity of the decoding procedure can even result to a decrease in the bit error rate.

SUBJECT AREA: Channel Coding

KEYWORDS: convolutional codes, maximum likelihood decoding, minimum trellis complexity, trellis diagram, turbo codes.

*στους γονείς μου Μάρκο και Θάλεια
στην αδερφή μου Μαρίζα
στη γιαγιά μου Ελευθερία
στη Φωτεινή*

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	10
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	11
ΚΑΤΑΛΟΓΟΣ ΣΥΜΒΟΛΙΣΜΩΝ.....	13
ΑΚΡΩΝΥΜΙΑ	15
ΕΥΧΑΡΙΣΤΙΕΣ	21
1 ΕΙΣΑΓΩΓΗ.....	23
1.1 Αντικείμενο και Δομή της Διατριβής	26
2 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΣΥΓΚΕΡΑΣΤΙΚΟΥΣ ΚΩΔΙΚΕΣ	31
2.1 Βασικά Στοιχεία της Θεωρίας Κωδικοποίησης	31
2.1.1 Μπλοκ Κώδικες	33
2.2 Συγκεραστικοί Κώδικες	36
2.2.1 Γραφικές Αναπαραστάσεις	41
2.2.2 Ελεύθερη Απόσταση-Φάσματα Βάρους των Συγκεραστικών Κωδίκων	43
2.2.3 Αποκωδικοποίηση	45
2.2.4 Κατηγορίες Συγκεραστικών Κωδίκων	49
2.3 Το Ελάχιστο Διάγραμμα Trellis για Γραμμικούς Μπλοκ και Συγκεραστικούς Κώδικες	52
3 ΚΑΤΑΣΚΕΥΕΣ ΣΥΓΚΕΡΑΣΤΙΚΩΝ ΚΩΔΙΚΩΝ ΧΑΜΗΛΗΣ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ .	63
3.1 Νέες Κατηγορίες Χρονικά Μεταβαλλόμενων Κωδίκων	63
3.2 Ο Ισοδύναμος Χρονικά Αμετάβλητος Κωδικοποιητής	65
3.3 Η Ελάχιστη Trellis Πολυπλοκότητα και Το Αντίστοιχο Ελάχιστο Trellis Module . .	69
3.4 Αποτελέσματα Αναζήτησης Κωδίκων και Συγκρίσεις	74
3.5 Ζητήματα Αποκωδικοποίησης	80
3.6 Βέλτιστα Trellis Modules για $(n, n - 1)$ Συγκεραστικούς Κώδικες	85
3.7 Άνω Φράγματα για την Ελεύθερη Απόσταση των Νέων Κωδίκων	91
3.A Απόδειξη Λήμματος 3.5	93

4	ΕΥΕΛΙΚΤΟΙ ΣΥΓΚΕΡΑΣΤΙΚΟΙ ΚΩΔΙΚΕΣ: ΜΕΤΑΒΛΗΤΟΣ ΡΥΘΜΟΣ ΚΑΙ ΠΟΛΥ-ΠΛΟΚΟΤΗΤΑ.....	97
4.1	Η Τεχνική της Απαλοιφής Μονοπατιών του Διαγράμματος Trellis	98
4.2	Συνδυάζοντας τις Τεχνικές της Απαλοιφής κωδικών Bit και της Απαλοιφής Μονοπατιών του Διαγράμματος Trellis: Ανάλυση Πολυπλοκότητας	101
4.2.1	Απαλοιφή Μονοπατιών του Διαγράμματος Trellis	101
4.2.2	Απαλοιφή Κωδικών Bit	106
4.2.3	Περιορισμοί	107
4.3	Διαδικασία Κατασκευής – Αναζήτησης	109
4.4	Οικογένειες Ευέλικτων Συγκεραστικών Κωδίκων	116
4.5	Προσομοιώσεις	121
4.A	Χαρακτηριστικά των υπό Εξέταση Κωδίκων	128
5	ΑΝΑΔΡΟΜΙΚΟΙ ΕΥΕΛΙΚΤΟΙ ΣΥΓΚΕΡΑΣΤΙΚΟΙ ΚΩΔΙΚΟΠΟΙΗΤΕΣ ΓΙΑ ΤΟΥΡΜΠΟ ΚΩΔΙΚΕΣ.....	131
5.1	Εισαγωγή στους Κώδικες Τούρμπο	131
5.1.1	Ο Τούρμπο Κωδικοποιητής	132
5.1.2	Ο Τούρμπο Αποκωδικοποιητής	132
5.1.3	Σχεδίαση Τούρμπο Κωδίκων	136
5.2	Αναδρομικοί Ευέλικτοι Συγκεραστικοί Κωδικοποιητές	141
5.3	Αποτελέσματα Αναζήτησης και Προσομοιώσεις	144
6	ΣΥΝΟΨΗ-ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ.....	161
6.1	Μελλοντικές Κατευθύνσεις	162
	ΑΝΑΦΟΡΕΣ.....	165
	ΕΥΡΕΤΗΡΙΟ.....	173

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

2.1	Μπλοκ διάγραμμα ενός τυπικού τηλεπικοινωνιακού συστήματος.	31
2.2	Συγκεραστικός κωδικοποιητής ρυθμού $R = 1/2$	37
2.3	Αναδρομικός συστηματικός συγκεραστικός κωδικοποιητής ρυθμού $R = 1/2$	39
2.4	Διάγραμμα καταστάσεων του συγκεραστικού κωδικοποιητή στο Σχ. 2.2.	42
2.5	Διάγραμμα trellis του συγκεραστικού κωδικοποιητή στο Σχ. 2.2.	42
2.6	Trellis module του συγκεραστικού κωδικοποιητή στο Σχ. 2.2.	43
2.7	Το trellis module του $(3, 2, 2)$ PCC που προκύπτει από τον $\mathbf{G}(D)$ στην (2.18) σε συνδυασμό με τον πίνακα απαλοιφής στην (2.51).	51
2.8	Το συμβατικό trellis module του $(3, 2, 2)$ PCC στο Παράδειγμα 2.17.	52
2.9	Ελάχιστο διάγραμμα trellis του $(7, 4)$ γραμμικού μπλοκ κώδικα Hamming στο Παράδειγμα 2.20.	56
2.10	Συμβατικό trellis module του συγκεραστικού κώδικα στο Παράδειγμα 2.22.	59
2.11	Συμβατικό trellis module του συγκεραστικού κώδικα στο Παράδειγμα 2.22, που προκύπτει από μη κανονικό πίνακα γεννήτορα.	60
2.12	Το ελάχιστο trellis module του συγκεραστικού κώδικα στο Παράδειγμα 2.22.	61
3.1	Trellis module του $(8, 4)$ συγκεραστικού κώδικα στο παράδειγμα 3.1.	65
3.2	Ελάχιστο trellis module του $(8, 4)$ συγκεραστικού κώδικα στο Παράδειγμα 3.1.	73
3.3	Bit-oriented ελάχιστο trellis module του $(8, 4)$ συγκεραστικού κώδικα στο Παράδειγμα 3.1.	73
3.4	Προσομοιώσεις (AWGN κανάλι και διαμόρφωση BPSK) κωδίκων στους Πίνακες 3.1, 3.2 και 3.3 (σημειωμένοι με αστερίσκο) και PCCs.	80
3.5	Προσομοιώσεις (AWGN κανάλι και διαμόρφωση BPSK) κωδίκων στους Πίνακες 3.1, 3.2 και 3.3 (σημειωμένοι με αστερίσκο) και PCCs.	81
4.1	Τα trellis modules των κωδίκων (α') \mathcal{C} και (β') \mathcal{C}_{pr} , του Παραδείγματος 4.1.	99
4.2	Το trellis module του κώδικα \mathcal{C}_1 στο Παράδειγμα 4.5.	113
4.3	Το trellis module του κώδικα \mathcal{C}_2 στο Παράδειγμα 4.5.	114
4.4	Το trellis module του κώδικα \mathcal{C}_3 στο Παράδειγμα 4.5.	114
4.5	Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμούς (α') $1/2$ και (β') $2/3$	122
4.6	Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμούς (α') $3/4$ και (β') $4/5$	123
4.7	Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμούς (α') $5/6$ και (β') $6/7$	124
4.8	Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμό $7/8$	125

4.9	Information bit oriented trellis module που αναπαριστά τον κώδικα \mathcal{C}_2 .	129
4.10	Trellis module που αναπαριστά τον κώδικα \mathcal{C}_4 .	129
4.11	Information bit-oriented trellis module που αναπαριστά τον κώδικα \mathcal{C}_4 .	130
5.1	Ο τούρμπο κωδικοποιητής: (α') μπλοκ διάγραμμα και (β') ένα συγκεκριμένο παράδειγμα.	133
5.2	Μπλοκ διάγραμμα του τούρμπο αποκωδικοποιητή.	135
5.3	Η μορφή της καμπύλης απόδοσης ενός τούρμπο κώδικα.	136
5.4	Διαγράμματα EXIT που αντιστοιχούν στον συγκεραστικό κωδικοποιητή της (5.1) για (α') $E_b/N_0 = 0.5$ και (β') $E_b/N_0 = 0$.	140
5.5	Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Χρησιμοποιείται τυχαίος συμπλέκτης μήκους 1000.	149
5.6	Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Επιλέγεται καινούργιος τυχαίος συμπλέκτης μήκους 1000 για κάθε λέξη που μεταδίδεται.	151
5.7	Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Χρησιμοποιείται τυχαίος συμπλέκτης μήκους 10000.	153
5.8	Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Επιλέγεται καινούργιος τυχαίος συμπλέκτης μήκους 10000 για κάθε λέξη που μεταδίδεται.	155

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

3.1	Αποτελέσματα αναζήτησης	76
3.2	Αποτελέσματα αναζήτησης (συνέχεια)	77
3.3	Αποτελέσματα αναζήτησης (συνέχεια)	78
3.4	Συγκρίσεις με κώδικες από τη βιβλιογραφία	79
4.1	Οικογένεια κωδίκων που παράγεται από το μητρικό συγκεκριαστικό κωδικοποιητή $\mathbf{G}(D) = [1 + D^4 + D^5 + D^6 + D^8 \quad 1 + D + D^3 + D^5 + D^6 + D^7 + D^8]$	117
4.2	Οικογένεια κωδίκων που παράγεται από το μητρικό συγκεκριαστικό κωδικοποιητή $\mathbf{G}(D) = [1 + D^3 + D^4 + D^6 + D^8 \quad 1 + D + D^2 + D^3 + D^4 + D^6 + D^7 + D^8 \quad 1 + D^2 + D^3 + D^4 + D^8]$	118
4.3	Οικογένεια κωδίκων που παράγεται από το μητρικό συγκεκριαστικό κωδικοποιητή $\mathbf{G}(D) = [1 + D^3 + D^4 + D^6 + D^8 \quad 1 + D + D^2 + D^3 + D^4 + D^6 + D^7 + D^8 \quad 1 + D^2 + D^3 + D^4 + D^8 \quad 1 + D + D^2 + D^3 + D^5 + D^7 + D^8]$	119
4.4	Λεπτομερής επεξεργασία των προσομοιώσεων της κωδικής οικογένειας στον Πίνακα 4.3	127
5.1	Οικογένεια κωδικοποιητών που παράγεται από τον αναδρομικό συστηματικό μητρικό κωδικοποιητή $\mathbf{G}(D) = \left[1 \quad \frac{1+D+D^4+D^5}{1+D^2+D^3+D^4+D^5} \right]$	145
5.2	Αριθμός επαναλήψεων αποκωδικοποίησης των κωδίκων στο Σχ. 5.5	150
5.3	Αριθμός επαναλήψεων αποκωδικοποίησης των κωδίκων στο Σχ. 5.6	152
5.4	Αριθμός επαναλήψεων αποκωδικοποίησης των κωδίκων στο Σχ. 5.7	154
5.5	Αριθμός επαναλήψεων αποκωδικοποίησης των κωδίκων στο Σχ. 5.8	156
5.6	Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1, όταν χρησιμοποιείται ο τυχαίος συμπλέκτης μήκους 1000 των προσομοιώσεων στο Σχ. 5.5.	157
5.7	Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1, όταν χρησιμοποιείται ο τυχαίος συμπλέκτης μήκους 1000 των προσομοιώσεων στο Σχ. 5.5 (συνέχεια).	158
5.8	Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1. Χρησιμοποιείται διαφορετικός τυχαίος συμπλέκτης μήκους 1000 για κάθε μεταδιδόμενη λέξη.	159
5.9	Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1. Χρησιμοποιείται διαφορετικός τυχαίος συμπλέκτης μήκους 1000 για κάθε μεταδιδόμενη λέξη (συνέχεια).	160

ΚΑΤΑΛΟΓΟΣ ΣΥΜΒΟΛΙΣΜΩΝ

- $\{\mathbf{G}_j(t)\}_{j=0}^{mt}$: Ακολουθία $1 \times n_t$ δυαδικών πινάκων τη χρονική στιγμή t
 v_t : Ακολουθία εξόδου συγκεραστικού κωδικοποιητή τη χρονική στιγμή t
 $\mathbf{G}_{sys}(D)$: Αναδρομικός συστηματικός κωδικοποιητής
 $\lceil x \rceil$: Άνω ακέραιο μέρος του αριθμού x
 \mathbf{G}_{scalar} : Βαθμιδωτός πίνακας γεννήτορας
 $\text{rank}(G)$: Βαθμός πίνακα G
 \mathbf{g}_l : Γραμμή l του πίνακα γεννήτορα \mathbf{G}
 \mathcal{T} : Διάγραμμα Trellis
 b_j : Διάσταση χώρου ακμών του τμήματος trellis j
 s_j : Διάσταση χώρου καταστάσεων του διαγράμματος trellis σε βάθος j
 \mathcal{C}^\perp : Δυικός κώδικας του \mathcal{C}
 d_{min} : Ελάχιστη απόσταση κώδικα
 \mathcal{M}_{min} : Ελάχιστο trellis module
 E : Αριθμός συνενώσεων σε ένα trellis module
 d_f : Ελεύθερη απόσταση
 $[L(\mathbf{g}_l), R(\mathbf{g}_l)]$: Ενεργό διάστημα της \mathbf{g}_l
 $L(\mathbf{g}_l)$: Θέση πρώτου μη μηδενικού στοιχείου της \mathbf{g}_l
 $R(\mathbf{g}_l)$: Θέση τελευταίου μη μηδενικού στοιχείου της \mathbf{g}_l
 \mathbf{S}_t : Κατάσταση συγκεραστικού κωδικοποιητή τη χρονική στιγμή t
 $v_{j,i}$: Κωδικό bit i που μεταδίδεται κατά το trellis τμήμα j
 $L(u_t^{(i)}|\mathbf{r})$: Λογάριθμος λόγου πιθανοφάνειας
 $L_a(u_t^{(i)})$: Λογάριθμος λόγου a priori πιθανοτήτων
 E_b/N_0 : Λόγος ισχύος σήματος προς θόρυβο ανά bit

- $\gamma_t(S, S')$: Μετρική ακμής αλγορίθμου BCJR για αρχική και τελική κατάσταση S και S'
- m_t : Μνήμη κωδικοποιητή τη χρονική στιγμή t
- $P_d(e)$: Πιθανότητα επιλογής λανθασμένου μονοπατιού βάρους d
- $P(E)$: Πιθανότητα σφάλματος αποκωδικοποίησης
- $P_b(E)$: Πιθανότητα σφάλματος αποκωδικοποίησης ενός bit πληροφορίας
- \mathbf{G} : Πίνακας γεννήτορας
- \mathbf{G}_{ms} : Πίνακας γεννήτορας ελαχίστου ενεργού διαστήματος
- $\mathbf{G}_{pu}(D)$: Πίνακας γεννήτορας μετά από απαλοιφή κωδικών bit
- $\mathbf{G}(D)$: Πίνακας γεννήτορας στο πεδίο μετασχηματισμού D
- $\Theta(D)$: Πίνακας εντοπισμού μονοπατιών
- Σ : Σύνολο καταστάσεων διαγράμματος trellis (μπλοκ κώδικες) ή trellis module (συγκεραστικοί κώδικες)
- $|\Sigma|$: Πληθυκότητα συνόλου Σ
- $\alpha_t(S)$: Προς τα εμπρός μετρική αλγορίθμου BCJR για κατάσταση S
- $\beta_t(S)$: Προς τα πίσω μετρική αλγορίθμου BCJR για κατάσταση S
- R : Ρυθμός (n, k) κώδικα
- (n, k, m) : Συγκεραστικός κώδικας μήκους n , διάστασης k και μνήμης m
- $A(X)$: Συνάρτηση απαρίθμησης βάρους
- $T(X, W, L)$: Συνάρτηση εισόδου-εξόδου απαρίθμησης βάρους
- \mathbf{G}_{sys} : Συστηματικός πίνακας γεννήτορας
- m : Συνολική μνήμη συγκεραστικού κωδικοποιητή
- $I(x; y)$: Αμοιβαία πληροφορία μεταξύ των τυχαίων μεταβλητών x και y
- T_{ζ_1, ζ_2} : Σύνολο τμημάτων trellis module με αλλαγή διάστασης χώρου καταστάσεων από ζ_1 σε ζ_2
- TC : Trellis πολυπλοκότητα
- $TC(\mathcal{M})$: Trellis πολυπλοκότητα του trellis module \mathcal{M}
- $A^{CT}(w, Z)$: Υπό συνθήκη συνάρτηση απαρίθμησης βάρους τούρμπο κώδικα

AKPONYMIA

ACS	Add Compare Select
AWGN	Additive White Gaussian Noise
BCJR	Bahl Cocke Jelinec Raviv
BER	Bit Error Rate
BSC	Binary Symmetric Channel
CRC	Cyclic Redundancy Check
DMC	Discrete Memoryless Channel
ECT	Exhaustive Computation Technique
EXIT	EXtrinsic Information Transfer
GPCC	Generalized Punctured Convolutional Codes
IC	Integrated Circuit
IOWEF	Input Output Weight Enumerator Function
LLR	Log Likelihood Ratio
LTE	Long Term Evolution
LR	Left Right
MAP	Maximum a posteriori Probability
MLD	Maximum Likelihood Decoding
MPSK	M-Phase Shift Keying
MSGM	Minimum Span Generator Matrix
PCC	Punctured Convolutional Codes
RCPC	Rate Compatible Punctured Convolutional Codes
PDF	Probability Density Function
SISO	Soft Input Soft Output
VRPCC	Variable Rate Punctured Convolutional Codes
WRAN	Wireless Regional Area Networks

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις πιο θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου κ. Ν. Καλουπτσίδη, Καθηγητή του Πανεπιστημίου Αθηνών, για την υποστήριξη, τη συμπαράσταση και την εμπιστοσύνη που μου έδειξε όλα αυτά τα χρόνια των σπουδών μου. Η διαρκής επιστημονική του καθοδήγηση, οι εποικοδομητικές μας συζητήσεις και οι παροτρύνσεις του έπαιξαν καθοριστικό ρόλο στην προσπάθειά μου.

Ευχαριστώ επίσης ολόθερμα τον κ. Σ. Θεοδωρίδη, Καθηγητή του Πανεπιστημίου Αθηνών και τον κ. Θ. Σφηκόπουλο, Καθηγητή του Πανεπιστημίου Αθηνών που δέχτηκαν να αποτελέσουν μέλη της τριμελούς συμβουλευτικής επιτροπής και να αξιολογήσουν την υποβαλλόμενη διατριβή.

Ευχαριστώ θερμά τους κ. Α. Μεράκο, Καθηγητή του Πανεπιστημίου Αθηνών, κ. Ι. Σταυρακάκη, Καθηγητή του Πανεπιστημίου Αθηνών, κ. Ε. Σαγκριώτη, Αν. Καθηγητή του Πανεπιστημίου Αθηνών και κ. Ν. Κολοκοτρώνη, Λέκτορα του Πανεπιστημίου Πελοποννήσου, που δέχτηκαν να συμμετάσχουν στην επταμελή μου εξεταστική επιτροπή.

Ευχαριστώ θερμά τον κ. Π. Ριζομυλιώτη, Επ. Καθηγητή του Πανεπιστημίου Αγαίου, για την άριστη συνεργασία που είχαμε στα πλαίσια της διδακτορικής διατριβής και για την σημαντική συμβολή του στην ολοκλήρωση της προσπάθειάς μου. Ευχαριστώ επίσης τους φίλους και συνεργάτες Κ. Ξενούλη, Γ. Μπλεούνη, Κ. Λιμνιώτη και Θ. Γιαννακόπουλο, για τη στήριξη που μου παρείχαν όλα αυτά τα χρόνια καθώς και για τις εποικοδομητικές μας συζητήσεις.

Θέλω να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου για την πολυδιάστατη συμπαράσταση που μου προσέφερε όλα τα χρόνια των σπουδών μου. Ευχαριστώ ολόθερμα τη Φωτεινή για τη διαρκή στήριξη και συμπαράστασή της.

1. ΕΙΣΑΓΩΓΗ

Η θεωρία της κωδικοποίησης καναλιού θεμελιώθηκε από τον Claude Shannon το 1948 [1], ο οποίος απέδειξε ότι υπάρχει ένα σχήμα κωδικοποίησης με το οποίο μπορεί να επιτευχθεί οποιοδήποτε μικρή πιθανότητα σφάλματος κατά τη μετάδοση πληροφορίας μέσω ενός καναλιού, αν ο ρυθμός μετάδοσης της πληροφορίας είναι μικρότερος από τη χωρητικότητα του καναλιού και το μήκος του κώδικα είναι αρκετά μεγάλο. Η απόδειξη του Shannon δεν ήταν κατασκευαστική, καθώς βασίστηκε στην τεχνική του συνόλου των τυχαίων κωδίκων, χωρίς να αναδεικνύει τα βέλτιστα χαρακτηριστικά που πρέπει να έχει ένα σχήμα κωδικοποίησης. Επιπλέον, η απαίτηση για μεγάλα μήκη κωδίκων είναι απαγορευτική όσον αφορά σε θέματα πολυπλοκότητας και υλοποίησης. Η θεωρία κωδικοποίησης λοιπόν αποτελεί την προσπάθεια σχεδίασης κωδίκων, οι οποίοι “αγγίζουν” το θεωρητικά όρια που προέβλεψε ο Shannon, ενώ διαθέτουν αρκετή δομή, προκειμένου η κωδικοποίηση κι αποκωδικοποίηση να επιτυγχάνονται με μικρή (ή τουλάχιστον ανεκτή) πολυπλοκότητα.

Οι προσπάθειες των πρώτων χρόνων περιορίστηκαν σε αλγεβρικές τεχνικές κωδικοποίησης, δηλαδή στους γραμμικούς μπλοκ κώδικες. Έτσι, οι πρώτοι κώδικες που εμφανίστηκαν στη βιβλιογραφία ήταν οι κώδικες Hamming [2], και ακολούθησαν ο Golay [3], οι Reed Muller [4], καθώς και κυκλικοί κώδικες με έντονη αλγεβρική δομή και ισχυρή διορθωτική ικανότητα όπως οι Bose-Chaudhuri-Hocquenghem (BCH) [5, 6] και οι Reed-Solomon (RS) [7].

Οι συγκεραστικοί κώδικες (convolutional codes) ανακαλύφθηκαν το 1955 από τον Peter Elias [8], στην προσπάθεια του να βελτιστοποιήσει το ισοζύγιο απόδοσης και πολυπλοκότητας. Η “προφανής” περιγραφή τους από το διάγραμμα trellis, η ανάπτυξη αποδοτικών αλγορίθμων αποκωδικοποίησης (Viterbi) [9] οι οποίοι μπορούν να επεξεργαστούν soft μετρικές χωρίς ιδιαίτερη αύξηση της πολυπλοκότητας, σε αντίθεση με τους περισσότερους αλγεβρικούς αλγόριθμους των μπλοκ κωδίκων, καθώς και το γεγονός ότι η πολυπλοκότητα της αποκωδικοποίησης αυξάνεται γραμμικά (κι όχι εκθετικά) με το μήκος της κωδικής ακολουθίας, κατέστησαν τους συγκεραστικούς κώδικες εξαιρετικά δημοφιλείς στις εφαρμογές.

Η πολυπλοκότητα του διαγράμματος trellis ενός συγκεραστικού κώδικα, η οποία αυξάνεται εκθετικά με τη μνήμη του κωδικοποιητή και τον κωδικό ρυθμό, επηρεάζει άμεσα την πολυπλοκότητα του αλγόριθμου Viterbi. Έτσι, η αποκωδικοποίηση των κωδίκων υψηλού ρυθμού γίνεται ιδιαίτερα δύσκολο εγχείρημα. Για το λόγο αυτό προτείνεται η κατηγορία των *διάτρητων συγκεραστικών κωδίκων* (punctured convolutional codes) [10], οι οποίοι παράγονται από συγκεραστικούς κώδικες χαμηλού ρυθμού, απαλοίφοντας ένα ποσοστό των κωδικών τους bit βάσει ενός πίνακα απαλοίφης (puncturing matrix) προκειμένου να αυξηθεί ο ρυθμός. Οι διάτρητοι κώδικες μπορούν να αποκωδικοποιηθούν χρησιμοποιώντας ένα χρονικά μεταβαλλόμενο διάγραμμα trellis μειωμένης πολυπλοκότητας, το οποίο έχει προκύψει από διάγραμμα trellis του χαμηλόρυθμου *μητρικού κώδικα*. Στη βιβλιογραφία παρουσιάζεται μεγάλο πλήθος “καλών”

διάτρητων κωδίκων [11, 12, 13, 14, 15].

Γρήγορα έγινε αντιληπτό πως πέραν από την εξαιρετική απόδοση υπάρχουν κι άλλα χαρακτηριστικά που πρέπει να έχει ένας κώδικας για να είναι χρήσιμος σε πρακτικές εφαρμογές. Για παράδειγμα, στην περίπτωση που τα χαρακτηριστικά του καναλιού μεταβάλλονται με το χρόνο, ή στην περίπτωση που διαφορετικά είδη πληροφορίας απαιτούν διαφορετικά επίπεδα προστασίας, η χρήση ενός και μοναδικού ισχυρού κώδικα (χαμηλού ρυθμού και υψηλής πολυπλοκότητας αποκωδικοποίησης) σε όλες τις συνθήκες οδηγεί σε σπατάλη των πόρων του συστήματος. Από την άλλη, η υλοποίηση πολλών κωδικοποιητών και αποκωδικοποιητών, προκειμένου να επιλέγεται κάθε φορά ο κατάλληλος, θα ήταν ιδιαίτερα πολύπλοκη, ειδικά στις περιπτώσεις μικρών τερματικών (όπως π.χ. τα κινητά τηλέφωνα). Η ιδανική λύση είναι η κατασκευή ενός και μοναδικού κωδικοποιητή (και αποκωδικοποιητή αντίστοιχα), του οποίου κάποια παράμετρος (π.χ. ο ρυθμός κωδικοποίησης) μπορεί να μεταβάλλεται, έτσι ώστε να προσαρμόζεται η διορθωτική του ικανότητα στις αντίστοιχες απαιτήσεις. Τη δυνατότητα αυτή μπορούν να την προσφέρουν οι διάτρητοι συγκεραστικοί κώδικες *μεταβλητού ρυθμού* (variable rate punctured convolutional codes-VRPCC) [11, 12, 16, 17]. Το εν λόγω σχήμα κωδικοποίησης αποτελείται από ένα “μπτρικό” συγκεραστικό κώδικα χαμηλού ρυθμού, κι ένα σύνολο πινάκων απαλοιφής κωδικών bit που οδηγούν σε διάφορους κωδικούς ρυθμούς. Ανάλογα με την περίπτωση επιλέγεται ο κατάλληλος πίνακας απαλοιφής, κι έτσι γίνεται αποδοτικότερη χρήση του εύρους ζώνης (bandwidth). Για παράδειγμα, το πρότυπο IEEE 802.22 για γνωστικά ασύρματα περιφερειακά δίκτυα (cognitive wireless regional area networks (WRAN)) χρησιμοποιεί (σαν υποχρεωτικό σχήμα κωδικοποίησης) διάτρητους συγκεραστικούς κώδικες ρυθμού 2/3, 3/4 και 5/6 που προκύπτουν από έναν μπτρικό συγκεραστικό κώδικα ρυθμού 1/2 [18]. Παρόμοιες κατασκευές αποτελούν και οι διάτρητοι συγκεραστικοί κώδικες *συμβατού ρυθμού* (rate compatible punctured convolutional codes-RCPC)[19], με τη διαφορά ότι οι κωδικές λέξεις των κωδίκων υψηλού ρυθμού περιέχονται στις λέξεις των κωδίκων χαμηλού ρυθμού, γεγονός που τους καθιστά κατάλληλους για σχήματα *αυτόματης αίτησης επανάληψης* (Automatic repeat request-ARQ). Στα άρθρα [20, 21, 22] η κατασκευή συγκεραστικών κωδίκων μεταβλητού ρυθμού βασίζεται στην *τεχνική απαλοιφής μονοπατιών* (path pruning) του διαγράμματος trellis. Σ’ αυτήν την περίπτωση, η απαλοιφή μονοπατιών και συνεπώς κωδικών λέξεων οδηγεί στη μείωση του ρυθμού του μπτρικού κώδικα.

Η επιτυχία του αλγόριθμου του Viterbi, ο οποίος χρησιμοποιεί το διάγραμμα trellis των συγκεραστικών κωδίκων, ενέπνευσε την ιδέα της αναπαράστασης ενός γραμμικού μπλοκ κώδικα με ένα αντίστοιχο (χρονικά μεταβαλλόμενο) διάγραμμα trellis, προκειμένου να αποκωδικοποιηθεί κι αυτός με τον αλγόριθμο του Viterbi. Πιο συγκεκριμένα, μελετήθηκε το διάγραμμα trellis με τη μικρότερη πολυπλοκότητα (ελάχιστο διάγραμμα trellis) το οποίο μπορεί να περιγράψει έναν γραμμικό μπλοκ κώδικα [23, 24]. Η ιδέα του ελάχιστου διαγράμματος trellis των μπλοκ κωδίκων ώθησε στην αναζήτηση διαγραμμάτων trellis μειωμένης πολυπλοκότητας και για τους

συγκεραστικούς κώδικες. Έτσι, στα [25] και [26] αποδεικνύεται ότι κάθε συγκεραστικός κώδικας μπορεί να αναπαρασταθεί από ένα ελάχιστο διάγραμμα trellis, λιγότερο πολύπλοκο από το συμβατικό. Όπως αναφέρεται χαρακτηριστικά στο [25], οι διάτρητοι συγκεραστικοί κώδικες δεν είναι τίποτα παραπάνω από κώδικες των οποίων οι πίνακες γεννήτορες έχουν συγκεκριμένη μορφή, γι' αυτό και μπορούν να περιγραφούν από διαγράμματα trellis συγκεκριμένης πολυπλοκότητας. Επίσης, τίθεται ως ανοιχτό πρόβλημα η κατασκευή κι άλλων τέτοιων κατηγοριών καλών συγκεραστικών κωδίκων χαμηλής πολυπλοκότητας [25]. Ενώ λοιπόν μέχρι τότε ως πολυπλοκότητα ενός συγκεραστικού κώδικα θεωρούνταν η μνήμη του κωδικοποιητή του (η οποία κατά κάποιον τρόπο σχετίζεται με την πολυπλοκότητα του συμβατικού διαγράμματος trellis) και κατασκευάζονταν κώδικες συγκεκριμένου ρυθμού και μνήμης, πολλοί ερευνητές άρχισαν να σχεδιάζουν κώδικες με κριτήριο την πολυπλοκότητα του ελάχιστου διαγράμματος trellis [27, 28, 29, 30, 31, 32, 33]. Συγκεκριμένα, στα [27, 28] κατασκευάζονται συγκεραστικοί κώδικες χρησιμοποιώντας τον πίνακα ελέγχου ισοτιμίας (parity check matrix), των οποίων το ελάχιστο διάγραμμα trellis έχει την ίδια πολυπλοκότητα με το αντίστοιχο διάγραμμα trellis των διάτρητων κωδίκων, ενώ στα [29, 30] σχεδιάζονται μέσω του πίνακα ελέγχου ισοτιμίας κώδικες ρυθμού $(n - 1)/n$ για διάφορες τιμές της ελάχιστης πολυπλοκότητας. Επίσης, στα [31, 32, 33] παρουσιάζεται η οικογένεια των γενικευμένων διάτρητων συγκεραστικών κωδίκων (generalized punctured convolutional codes GPCC), που περιλαμβάνει κώδικες διαφόρων ρυθμών και τιμών πολυπλοκότητας, οι οποίοι σχεδιάζονται θέτοντας συγκεκριμένους περιορισμούς στον πίνακα γεννήτορα (generator matrix).

Το 1993, παρουσιάζονται στο συνέδριο International Conference on Communications (ICC) οι τούρμπο κώδικες (turbo codes) [34], οι οποίοι δείχνουν να πραγματοποιούν την “υπόσχεση” που είχε δώσει ο Shannon πενήντα χρόνια νωρίτερα. Πράγματι, οι εν λόγω κώδικες επιτυγχάνουν χαμηλή πιθανότητα σφάλματος για αρκετά μικρή τιμή του σηματοθορυβικού λόγου. Ένας τούρμπο κώδικας αποτελείται από την παράλληλη σύνδεση δύο συστηματικών αναδρομικών (recursive systematic) συγκεραστικών κωδικοποιητών διαμέσου ενός συμπλέκτη (interleaver). Η αποκωδικοποίηση του γίνεται επαναληπτικά στους αντίστοιχους συγκεραστικούς αποκωδικοποιητές, χρησιμοποιώντας τον αλγόριθμο BCJR [23], έναν ξεχασμένο μέχρι τότε αλγόριθμο αποκωδικοποίησης συγκεραστικών κωδίκων, καθώς είχε την ίδια απόδοση με τον αλγόριθμο του Viterbi και αρκετά μεγαλύτερη πολυπλοκότητα. Η ύπαρξη του συμπλέκτη δυσκολεύει αρκετά την ανάλυση του εν λόγω κώδικα, ενώ η από κοινού βελτιστοποίηση του συγκεραστικού κωδικοποιητή και του συμπλέκτη είναι αδύνατη, καθώς υπάρχουν εξαιρετικά πολλοί συνδυασμοί (οι τιμές που παίρνει συνήθως το μήκος του συμπλέκτη κυμαίνονται από 10^2 έως 10^5). Η ανάλυση των Benedetto και Montorsi [35, 36] απλοποιεί το πρόβλημα, καθώς εντοπίζει τα χαρακτηριστικά που πρέπει να έχει ο συγκεραστικός κωδικοποιητής προκειμένου να μεγιστοποιείται η μέση απόδοση (για μέτριο και υψηλό σηματοθορυβικό λόγο) των τούρμπο κωδίκων που τον χρησιμοποιούν. Εφόσον επιλεγεί ο κατάλληλος συγκεραστικός κωδικοποιητής, υπάρχουν τεχνικές οι

οποίες μπορούν να οδηγήσουν στην εύρεση ενός συμπλέκτη ο οποίος θα αυξήσει την απόδοση του κώδικα παραπάνω από τη μέση απόδοση. Στη βιβλιογραφία υπάρχουν αρκετές εργασίες που σχεδιάζουν συγκεραστικούς κωδικοποιητές για τούρμπο κώδικες [37, 38].

Όπως ακριβώς και στην περίπτωση των συγκεραστικών κωδίκων, η τεχνική της απαλοιφής κωδικών bit μπορεί να μεταβάλει το ρυθμό ενός τούρμπο κώδικα, ενώ η χρήση ενός συνόλου πινάκων απαλοιφής οδηγεί σ' έναν διάτρητο τούρμπο κώδικα μεταβλητού ρυθμού [39, 40, 41]. Δυστυχώς όμως, όπως αναφέρθηκε και παραπάνω, η από κοινού βελτιστοποίηση του πίνακα απαλοιφής και του "μπτρικού" τούρμπο κώδικα είναι εξαιρετικά δύσκολη, καθώς υπάρχουν πολλοί συνδυασμοί, ενώ για κάθε πιθανό υποψήφιο θα πρέπει να υπολογιστούν μετρικές (όπως π.χ. η ελάχιστη απόσταση του κώδικα που προκύπτει), χρησιμοποιώντας ιδιαίτερα πολύπλοκους αλγόριθμους [42]. Μία διαφορετική προσέγγιση είναι η κατασκευή διάτρητων συγκεραστικών κωδικοποιητών με τη μέθοδο που παρουσιάστηκε στα [35, 36, 37, 38], κατάλληλων να χρησιμοποιηθούν σε τούρμπο κώδικες. Με λίγα λόγια, προκειμένου να σχεδιάσει κανείς τούρμπο κώδικες μεταβλητού ρυθμού, είναι προτιμότερο να σχεδιάσει συγκεραστικούς κώδικες μεταβλητού ρυθμού βελτιστοποιημένους για τούρμπο κωδικοποίηση [43, 44]. Αυτή η μέθοδος χρησιμοποιείται στο [44], όπου παράγεται μια οικογένεια αναδρομικών διάτρητων συγκεραστικών κωδικοποιητών μεταβλητού ρυθμού, από έναν μπτρικό αναδρομικό κωδικοποιητή χαμηλού ρυθμού.

1.1 Αντικείμενο και Δομή της Διατριβής

Αντικείμενο της διατριβής αποτελεί η μελέτη κι η σχεδίαση συγκεραστικών κωδικοποιητών, με σκοπό να χρησιμοποιηθούν είτε ως αυτόνομα σχήματα κωδικοποίησης, είτε ως δομικά στοιχεία ισχυρότερων και πολυπλοκότερων συστημάτων όπως οι τούρμπο κώδικες. Σχεδιαστικό κριτήριο αποτελεί η πολυπλοκότητα του διαγράμματος trellis, και κατ' επέκτασιν η υπολογιστική πολυπλοκότητα των αλγορίθμων αποκωδικοποίησης.

Ειδικότερα, παρουσιάζεται μια νέα οικογένεια συγκεραστικών κωδίκων, η οποία προκύπτει τροποποιώντας το διάγραμμα trellis των διάτρητων συγκεραστικών κωδίκων. Στόχος είναι η λογική αύξηση της πολυπλοκότητας, με σκοπό τη βελτίωση της απόδοσης. Η αύξηση της πολυπλοκότητας επιτυγχάνεται μεταβάλλοντας ελεγχόμενα την διάσταση του χώρου καταστάσεων του αρχικού trellis. Η προκύπτουσα οικογένεια κωδίκων αποτελεί υποκατηγορία των γενικευμένων διάτρητων συγκεραστικών κωδίκων [31, 32, 33]. Αλγεβρικές δομές οι οποίες εγγυώνται συγκεκριμένη ελάχιστη απόσταση (όπως π.χ. στην περίπτωση των γραμμικών μπλοκ BCH κωδίκων) είναι εξαιρετικά δύσκολο να βρεθούν για τους συγκεραστικούς κώδικες. Έτσι, προκειμένου να βρεθεί ο βέλτιστος κώδικας πρέπει να εξεταστούν όλες οι δυνατές επιλογές. Σε αντίθεση με την προτεινόμενη οικογένεια, οι GPCC αποτελούν μία ευρεία κατηγορία κωδίκων, και όσο αυξάνεται ο κωδικός ρυθμός και η ελάχιστη πολυπλοκότητα, ο χώρος αναζήτησης (search space) γιγαντώνεται, καθιστώντας δύσκολη την εύρεση καλών κωδίκων. Στην παρούσα

διατριβή παρουσιάζονται νέοι κώδικες διαφόρων ρυθμών και τιμών πολυπλοκότητας. Σε πολλές περιπτώσεις οι εν λόγω κώδικες παρουσιάζουν σημαντική βελτίωση της απόδοσης, για μικρή αύξηση της πολυπλοκότητας, συγκριτικά με τους διάτρητους συγκεραστικούς κώδικες. Επίσης, παρουσιάζεται μία αποδοτική τεχνική για την κατασκευή του ελάχιστου διαγράμματος trellis των συγκεκριμένων κωδίκων, καθώς και ένα άνω φράγμα της ελεύθερης απόστασης τους.

Στην παρούσα διατριβή, προτείνεται μία μέθοδος για την κατασκευή *ευέλικτων* συγκεραστικών κωδίκων, οι οποίοι μπορούν να μεταβάλλουν τον ρυθμό τους, καθώς και την πολυπλοκότητα του διαγράμματος trellis, και κατ' επέκτασιν την υπολογιστική πολυπλοκότητα του αλγόριθμου του Viterbi. Η μέθοδος συνδυάζει τις τεχνικές της απαλοιφής μονοπατιών του διαγράμματος trellis (path pruning technique) και της απαλοιφής κωδικών bit (puncturing technique). Χρησιμοποιώντας το διάγραμμα trellis ενός χαμηλόρυθμου μπτρικού συγκεραστικού κώδικα, παράγονται διαγράμματα trellis διαφόρων τιμών πολυπλοκότητας, που αντιστοιχούν σε κώδικες διαφόρων ρυθμών. Σε αντίθεση με τους κλασσικούς συγκεραστικούς κώδικες μεταβλητού ρυθμού [13, 16, 17], με τους οποίους “ανταλλάσσεται” κέρδος κωδικοποίησης για εύρος ζώνης (bandwidth) και αντιστρόφως, οι προτεινόμενες κατασκευές προσθέτουν μία ακόμη “διάσταση”, την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης, οδηγώντας έτσι σε σχήματα κωδικοποίησης που διαχειρίζονται αποδοτικότερα τους πόρους του συστήματος. Επίσης, αναλύεται η επίδραση του συνδυασμού των δύο χρησιμοποιούμενων τεχνικών στο “προφίλ πολυπλοκότητας” του μπτρικού κώδικα, ενώ παρουσιάζονται και τρία σχήματα ευέλικτων κωδίκων που παράγονται από μπτρικούς κωδικοποιητές ρυθμού $1/2$, $1/3$ και $1/4$. Κάθε ένα περιέχει σημαντικό αριθμό κωδίκων. Όπως παρατηρείται, η απόδοση των ευέλικτων κωδίκων είναι συγκρίσιμη με την απόδοση των κωδίκων μεταβλητού ρυθμού στα [11, 12]. Με λίγα λόγια, η επιπρόσθετη δυνατότητα μεταβολής της υπολογιστικής πολυπλοκότητας της αποκωδικοποίησης δεν επιφέρει μείωση της απόδοσης εν συγκρίσει με τους διάτρητους κώδικες μεταβλητού ρυθμού.

Οι ευέλικτοι συγκεραστικοί κώδικες χρησιμοποιούνται στη συνέχεια, προκειμένου να κατασκευαστούν ευέλικτα τούρμπο σχήματα κωδικοποίησης. Συγκεκριμένα, εξετάζεται η δυνατότητα εφαρμογής των προαναφερθέντων αποτελεσμάτων χρησιμοποιώντας αναδρομικούς μπτρικούς κωδικοποιητές, οι οποίοι είναι κατάλληλοι ως περιεχόμενοι κωδικοποιητές των τούρμπο κωδίκων. Ακολουθώντας την προσέγγιση των εργασιών [35, 36, 37, 38, 44], οι εν λόγω ευέλικτοι αναδρομικοί συγκεραστικοί κωδικοποιητές σχεδιάζονται με κριτήριο μετρικές που βελτιστοποιούν τη μέση απόδοση των κωδίκων που τους χρησιμοποιούν. Έτσι, παρουσιάζεται μία κατάλληλη οικογένεια ευέλικτων κωδικοποιητών που παράγεται από έναν μπτρικό συγκεραστικό κώδικα ρυθμού $1/2$. Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν την εν λόγω κωδική οικογένεια, καθώς και τυχαίους συμπλέκτες διαφόρων μηκών δείχνουν ότι σε συγκεκριμένες περιοχές τιμών του σηματοθορυβικού λόγου, μια σημαντική μείωση της υπολογιστικής πολυπλοκότητας της αποκωδικοποίησης μπορεί ακόμα και να μειώσει το ρυθμό εσφαλμένων bit (bit error

rate). Δηλαδή, η εφαρμογή της συγκεκριμένης μεθόδου για την κατασκευή ευέλικτων τούρμπο κωδίκων οδηγεί σε ιδιαίτερα αποδοτικά σχήματα κωδικοποίησης.

Η δομή της διατριβής όπως αυτή διαρθρώνεται ανά κεφάλαιο είναι η ακόλουθη:

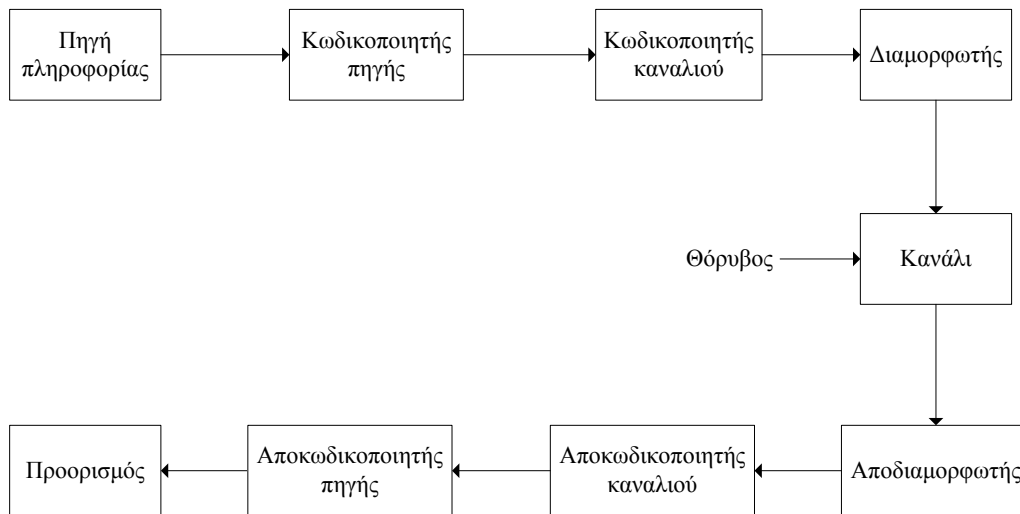
- Στο **Κεφάλαιο 2** εισάγονται βασικές έννοιες της θεωρίας των συγκεραστικών κωδίκων, χαρακτηριστικά των κωδικοποιητών τους, αλγόριθμοι αποκωδικοποίησης, το ελάχιστο διάγραμμα trellis για μπλοκ και συγκεραστικούς κώδικες, καθώς και κάποιες από τις σημαντικότερες κατηγορίες συγκεραστικών κωδίκων.
- Στο **Κεφάλαιο 3** Παρουσιάζεται η προτεινόμενη κατηγορία συγκεραστικών κωδίκων χαμηλής πολυπλοκότητας, ενώ αποδεικνύονται βασικά χαρακτηριστικά για τους χρονικά αμετάβλητους κωδικοποιητές των συγκεκριμένων κωδίκων. Υπολογίζεται η ελάχιστη πολυπλοκότητα, και περιγράφεται μία αποδοτική τεχνική για την κατασκευή του ελάχιστου διαγράμματος trellis. Επίσης, παρουσιάζονται τρεις πίνακες με κώδικες της προτεινόμενης κατηγορίας, οι οποίοι συγκρίνονται με τους διάτρητους κώδικες, τους GPCC, καθώς και άλλους κώδικες της βιβλιογραφίας. Τέλος αποδεικνύεται ένα άνω φράγμα για την ελεύθερη απόσταση των νέων κωδίκων, το οποίο ενδεχομένως μπορεί να απλοποιήσει τη διαδικασία αναζήτησης.
- Στο **Κεφάλαιο 4** περιγράφεται η γενικότερη μορφή της διαδικασίας της απαλοιφής μονοπατιών του διαγράμματος trellis, καθώς και ο τρόπος με τον οποίο θα χρησιμοποιηθεί στην παρούσα διατριβή, ενώ προτείνεται μια μέθοδος η οποία συνδυάζει την τελευταία μαζί με την τεχνική απαλοιφής κωδικών bit για την κατασκευή ευέλικτων συγκεραστικών κωδίκων. Διερευνάται η επίδραση των δύο τεχνικών στην πολυπλοκότητα του διαγράμματος trellis, και παρέχονται κλειστοί τύποι διαφορών ποσοτήτων που αφορούν την πολυπλοκότητα. Επίσης, περιγράφεται η διαδικασία κατασκευής μίας οικογένειας ευέλικτων κωδίκων από ένα μητρικό κώδικα. Εφαρμόζοντας τη συγκεκριμένη διαδικασία σχεδιάζονται τρεις οικογένειες ευέλικτων κωδίκων, οι οποίοι συγκρίνονται με αντίστοιχα σχήματα κωδικοποίησης όπως οι διάτρητοι συγκεραστικοί κώδικες μεταβλητού ρυθμού, ενώ παρέχονται και αποτελέσματα προσομοιώσεων.
- Στο **Κεφάλαιο 5** γίνεται μια σύντομη παρουσίαση βασικών εννοιών των τούρμπο κωδίκων, ενώ αναλύεται η εφαρμογή των αποτελεσμάτων του προηγούμενου κεφαλαίου σε αναδρομικούς μητρικούς κωδικοποιητές, προκειμένου να αναπτυχθούν ευέλικτοι τούρμπο κώδικες. Κατασκευάζεται μία οικογένεια ευέλικτων αναδρομικών συγκεραστικών κωδικοποιητών, κατάλληλη για να χρησιμοποιηθεί ως περιεχόμενος κωδικοποιητής σε τούρμπο κώδικες. Τέλος, παρουσιάζονται προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν την εν λόγω οικογένεια, με διάφορους τυχαίους συμπλέκτες.

- Στο **Κεφάλαιο 6** συνοψίζονται τα αποτελέσματα και παρουσιάζονται μελλοντικές ερευνητικές κατευθύνσεις.

2. ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΣΥΓΚΕΡΑΣΤΙΚΟΥΣ ΚΩΔΙΚΕΣ

2.1 Βασικά Στοιχεία της Θεωρίας Κωδικοποίησης

Το μπλοκ διάγραμμα ενός τηλεπικοινωνιακού συστήματος φαίνεται στο Σχ. 2.1. Η πηγή πληροφορίας παράγει την πληροφορία που πρέπει να μεταδοθεί στον προορισμό, ενώ ο κωδικοποιητής πηγής τη μετασχηματίζει σε μια ακολουθία δυαδικών ψηφίων (binary digits-bits), η οποία ονομάζεται ακολουθία πληροφορίας \mathbf{u} . Ο κωδικοποιητής καναλιού μετασχηματίζει την ακολουθία πληροφορίας \mathbf{u} σε μία κωδικοποιημένη ακολουθία, η οποία ονομάζεται κωδική λέξη \mathbf{v} . Στις περισσότερες των περιπτώσεων (όπως και στην παρούσα διατριβή), η \mathbf{v} είναι δυαδική ακολουθία. Ο διαμορφωτής μετατρέπει τα σύμβολα στην έξοδο του κωδικοποιητή καναλιού σε κυματομορφές κατάλληλες για μετάδοση, διάρκειας T δευτερολέπτων. Οι κυματομορφές εκπέμπονται στο κανάλι και αλλοιώνονται από το θόρυβο. Στο δέκτη τώρα, ακολουθείται η αντίστροφη διαδικασία. Δηλαδή, ο αποκωδικοποιητής επεξεργάζεται κάθε ληφθείσα κυματομορφή διάρκειας T και παράγει στην έξοδό του τη ληφθείσα ακολουθία \mathbf{r} , η οποία αντιστοιχεί στην κωδική λέξη \mathbf{v} . Ο αποκωδικοποιητής καναλιού παράγει από την \mathbf{r} την εκτιμώμενη ακολουθία πληροφορίας $\hat{\mathbf{u}}$, και ο αποκωδικοποιητής πηγής με τη σειρά του εκτιμάει την πληροφορία που παράχθηκε στην πηγή, και την μεταφέρει στον προορισμό.



Σχήμα 2.1. Μπλοκ διάγραμμα ενός τυπικού τηλεπικοινωνιακού συστήματος.

Προκειμένου να μεταδοθεί η πληροφορία χρησιμοποιώντας $M = 2^l$ κυματομορφές, η έξοδος του κωδικοποιητή καναλιού πρέπει να χωριστεί σε τμήματα των l bit, τα οποία ονομάζονται σύμβολα. Κάθε σύμβολο αντιστοιχίζεται σε ένα από τα M σήματα ενός συνόλου σημάτων S . Κάθε σήμα είναι μία κυματομορφή διάρκειας T . Ένα παράδειγμα M -αδικής διαμόρφωσης είναι η M -αδική διαμόρφωση μετατόπισης φάσης (M-phase shift keying-MPSK) [45], το σύνολο

σημάτων της οποίας δίνεται από τη σχέση

$$s_i(t) = \sqrt{\frac{2E_s}{T}} \cos(2\pi f_0 t + \phi_i), \quad 0 \leq t \leq T \quad (2.1)$$

όπου $\phi_i = 2\pi(i - 1)/M$, για $1 \leq i \leq M$.

Μία συνήθης μορφή θορύβου σε ένα τηλεπικοινωνιακό σύστημα είναι ο *προσθετικός λευκός γκαουσιανός θόρυβος* (additive white gaussian noise–AWGN) [46, 45]. Εάν το μεταδιδόμενο σήμα είναι το

$$r(t) = s(t) + n(t) \quad (2.2)$$

όπου $n(t)$ είναι μία Γκαουσιανή τυχαία διεργασία με μονόπλευρη φασματική πυκνότητα ισχύος N_0 .

Ο αποδιαμορφωτής παράγει μία έξοδο που αντιστοιχεί στο λαμβανόμενο σήμα, σε κάθε διάστημα των T δευτερολέπτων. Η έξοδος μπορεί να είναι ένας οποιοσδήποτε πραγματικός αριθμός, ή μία τιμή από ένα προαποφασισμένο σύνολο Q τιμών. Στη δεύτερη περίπτωση, εφόσον η έξοδος του αποδιαμορφωτή σε ένα συγκεκριμένο διάστημα δεν εξαρτάται από τις προηγούμενες μεταδόσεις, ο συνδυασμός ενός M -αδικού διαμορφωτή, του φυσικού καναλιού και του αποδιαμορφωτή μπορούν να μοντελοποιηθούν σαν ένα *διακριτό κανάλι χωρίς μνήμη* (discrete memoryless channel–DMC). Το DMC περιγράφεται από ένα σύνολο *πιθανοτήτων μετάβασης* (transition probabilities) $P(i|j)$, $0 \leq i \leq M - 1$, $0 \leq j \leq Q - 1$, όπου i και j τα σύμβολα στην είσοδο του διαμορφωτή και στην έξοδο του αποδιαμορφωτή αντίστοιχα, και $P(i|j)$ η πιθανότητα να λάβουμε το j , δεδομένου ότι μεταδόθηκε το i . Στην περίπτωση που $M = Q = 2$ και η κατανομή πλάτους του θορύβου είναι συμμετρική, προκύπτει το γνωστό *δυναμικό συμμετρικό κανάλι* (binary symmetric channel–BSC).

Εάν η έξοδος του αποδιαμορφωτή δεν είναι κβαντισμένη, τότε ο συνδυασμός του διαμορφωτή, του φυσικού καναλιού και του αποδιαμορφωτή αποτελεί ένα κανάλι διακριτής εισόδου–συνεχούς εξόδου. Η έξοδος του καναλιού είναι μια τυχαία μεταβλητή, η οποία μπορεί να πάρει οποιαδήποτε πραγματική τιμή y . Εάν το φυσικό κανάλι υπόκειται μόνο σε AWGN με μέση τιμή μηδέν και μονόπλευρη φασματική πυκνότητα ισχύος N_0 , τότε η έξοδος του καναλιού είναι μία Γκαουσιανή τυχαία μεταβλητή με διακύμανση $\sigma^2 = N_0/2$. Σ' αυτήν την περίπτωση, το κανάλι περιγράφεται από ένα σύνολο M (Γκαουσιανών) υπό συνθήκη *συναρτήσεων πυκνότητας πιθανότητας* (probability density function–pdf) $p(y|x)$, $x \in \{0, 1, \dots, M - 1\}$. Εάν για παράδειγμα χρησιμοποιείται BPSK διαμόρφωση, όπου $M = 2$ και $x \in \{0, 1\}$, το κανάλι περιγράφεται από τις παρακάτω υπό συνθήκη pdf

$$\begin{aligned} p(y|x = 0) &= \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y + \sqrt{E_s})^2}{N_0}} \\ p(y|x = 1) &= \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y - \sqrt{E_s})^2}{N_0}} \end{aligned} \quad (2.3)$$

όπου E_s είναι η ενέργεια του σήματος.

Ο αποκωδικοποιητής καναλιού (τον οποίο θα αποκαλούμε απλά αποκωδικοποιητή) πρέπει να παράγει την εκτιμώμενη ακολουθία πληροφορίας $\hat{\mathbf{u}}$, ακολουθώντας κάποια στρατηγική. Ισοδύναμα, εφόσον υπάρχει αντιστοιχία ένα προς ένα μεταξύ κωδικών λέξεων και ακολουθιών πληροφορίας, μπορεί να παράγει μια εκτίμηση \hat{v} της κωδικής λέξης v . Προφανώς, η καλύτερη στρατηγική είναι αυτή που θα ελαχιστοποιήσει την πιθανότητα να συμβεί σφάλμα κατά την αποκωδικοποίηση. Σφάλμα κατά την αποκωδικοποίηση συμβαίνει όταν $\hat{v} \neq v$. Δοθέντος ότι η ληφθείσα ακολουθία είναι η \mathbf{r} , η υπό συνθήκη πιθανότητα σφάλματος κατά την αποκωδικοποίηση ορίζεται ως

$$P(E|\mathbf{r}) = P(\hat{v} \neq v|\mathbf{r}).$$

Η πιθανότητα σφάλματος κατά την αποκωδικοποίηση δίνεται από τη σχέση

$$P(E) = \sum_{\mathbf{r}} P(E|\mathbf{r})P(\mathbf{r})$$

όπου $P(\mathbf{r})$ είναι η πιθανότητα της ληφθείσας ακολουθίας \mathbf{r} . Η $P(\mathbf{r})$ δεν εξαρτάται από την αποκωδικοποίηση, οπότε ένας βέλτιστος κανόνας αποκωδικοποίησης (ένας κανόνας δηλαδή που ελαχιστοποιεί την $P(E)$), θα πρέπει να ελαχιστοποιεί την $P(E|\mathbf{r}) = P(\hat{v} \neq v|\mathbf{r})$ για όλα τα \mathbf{r} , ή ισοδύναμα να μεγιστοποιεί την $P(\hat{v} = v|\mathbf{r})$. Έτσι, για συγκεκριμένο \mathbf{r} , η \hat{v} θα πρέπει να είναι η κωδική λέξη v η οποία μεγιστοποιεί την ποσότητα

$$P(v|\mathbf{r}) = \frac{P(\mathbf{r}|v)P(v)}{P(\mathbf{r})}. \quad (2.4)$$

Με λίγα λόγια, ως εκτιμώμενη κωδική λέξη επιλέγεται αυτή που είναι πιθανότερο να εστάλη, δεδομένου ότι ελήφθη η ακολουθία \mathbf{r} . Εάν όλες οι λέξεις πληροφορίας, και συνεπώς όλες οι κωδικές λέξεις, έχουν την ίδια πιθανότητα εμφάνισης, τότε ο κανόνας αποκωδικοποίησης στην (2.4) είναι ισοδύναμος με τη μεγιστοποίηση της $P(\mathbf{r}|v)$. Ο αποκωδικοποιητής ο οποίος χρησιμοποιεί τον προαναφερθέντα κανόνα, ονομάζεται *αποκωδικοποιητής μεγίστης πιθανοφάνειας* (maximum likelihood decoder–MLD).

Ο κωδικοποιητής καναλιού μπορεί να παράγει έναν *μπλοκ κώδικα* (block code) ή έναν *συγκεραστικό κώδικα* (convolutional code). Με τη δεύτερη κατηγορία θα ασχοληθούμε εκτενώς στην επόμενη ενότητα, καθώς και στα παρακάτω Κεφάλαια. Βασικές έννοιες που σχετίζονται με τους μπλοκ κώδικες θα παρουσιαστούν στην επόμενη υποενότητα.

2.1.1 Μπλοκ Κώδικες

Θεωρούμε ότι η έξοδος μιας πηγής πληροφορίας είναι μία ακολουθία από bit. Στη μπλοκ κωδικοποίηση, η παραπάνω ακολουθία χωρίζεται σε μπλοκ των k bit, τα οποία συμβολίζονται

με \mathbf{u} . Υπάρχουν 2^k διαφορετικά μπλοκ από bit, τα οποία αποτελούν τις λέξεις ή ακολουθίες πληροφορίας. Ο κωδικοποιητής αντιστοιχίζει κάθε λέξη πληροφορίας σε μία συγκεκριμένη ακολουθία \mathbf{v} , η οποία αποτελείται από n bit, $n > k$, και ονομάζεται κωδική λέξη. Το σύνολο των 2^k κωδικών λέξεων λέγεται *μπλοκ κώδικας*. Προφανώς, οι κωδικές λέξεις θα πρέπει να είναι διακριτές μεταξύ τους, άρα η αντιστοιχία μεταξύ μιας λέξης πληροφορίας \mathbf{u} και της κωδικής της λέξης \mathbf{v} είναι ένα προς ένα.

Η διαδικασία αποκωδικοποίησης ενός τυχαίου μπλοκ κώδικα έχει απαγορευτική πολυπλοκότητα, ειδικά όσο αυξάνουν τα k και n , γι' αυτό και είναι απαραίτητη η ύπαρξη δομής στον κώδικα. Στη συντριπτική πλειοψηφία των κωδίκων που εμφανίζονται στη βιβλιογραφία το δημοφιλέστερο είδος δομής είναι η γραμμικότητα.

Ορισμός 2.1. Ένας μπλοκ κώδικας με 2^k κωδικές λέξεις μήκους n καλείται *γραμμικός* (n, k) κώδικας αν και μόνο αν οι κωδικές του λέξεις αποτελούν έναν k -διάστατο υπόχωρο του F_2^n .

Προφανώς, οι συνιστώσες των κωδικών λέξεων ενός γραμμικού μπλοκ κώδικα μπορούν να είναι στοιχεία οποιουδήποτε πεδίου F_q . Στην παρούσα διατριβή θα ασχοληθούμε μόνο με δυαδικούς κώδικες.

Εφόσον ένας (n, k) γραμμικός κώδικας \mathcal{C} είναι ένας k -διάστατος υπόχωρος του F_2^n , υπάρχουν k γραμμικά ανεξάρτητες κωδικές λέξεις $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1}$, των οποίων οι γραμμικοί συνδυασμοί παράγουν όλες τις λέξεις \mathbf{v} του \mathcal{C} . Δηλαδή,

$$\mathbf{v} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \dots + u_{k-1}\mathbf{g}_{k-1} \quad (2.5)$$

όπου $u_i \in \{0, 1\}$, $0 \leq i \leq k - 1$. Εάν οι παραπάνω γραμμικά ανεξάρτητες κωδικές λέξεις διαταχθούν ως γραμμές ενός $k \times n$ πίνακα

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} \quad (2.6)$$

και θεωρήσουμε ότι η $\mathbf{u} = (u_0, u_1, \dots, u_{k-1})$ είναι μια λέξη πληροφορίας, τότε η αντίστοιχη κωδική λέξη προκύπτει ως

$$\mathbf{v} = \mathbf{uG}. \quad (2.7)$$

Η σχέση (2.7) περιγράφει τη διαδικασία κωδικοποίησης στην περίπτωση ενός γραμμικού μπλοκ κώδικα, και ο \mathbf{G} ονομάζεται *πίνακας γεννήτορας* (generator matrix) του κώδικα \mathcal{C} . Οποιοδήποτε σύνολο k γραμμικώς ανεξάρτητων κωδικών λέξεων (δηλαδή οποιαδήποτε βάση) του \mathcal{C} μπορεί να αποτελέσει έναν πίνακα γεννήτορα. Όπως γίνεται όμως εύκολα αντιληπτό από την (2.7), ένας διαφορετικός πίνακας γεννήτορας οδηγεί σε διαφορετική αντιστοίχιση μεταξύ κωδικών λέξεων και λέξεων πληροφορίας.

Ένας σημαντικός πίνακας γεννήτορας του \mathcal{C} είναι ο *συστηματικός*, ο οποίος είναι της μορφής

$$\mathbf{G}_{sys} = [\mathbf{P} | \mathbf{I}_{k \times k}] \quad (2.8)$$

όπου $\mathbf{I}_{k \times k}$ είναι ο $k \times k$ μοναδιαίος πίνακας, ενώ ο \mathbf{P} είναι ένας $(n - k) \times k$ δυαδικός πίνακας. Σε κάθε κωδική λέξη που παράγεται από τον \mathbf{G}_{sys} τα k τελευταία bit είναι αυτούσια τα bit της λέξης πληροφορίας, ενώ τα υπόλοιπα $n - k$ bit είναι τα *bit ελέγχου ισοτιμίας* (parity check bits). Έτσι, μετά την αποκωδικοποίηση, η εκτιμώμενη λέξη πληροφορίας προκύπτει άμεσα από την εκτιμώμενη κωδική λέξη.

Ο *δυϊκός χώρος* (dual space) του \mathcal{C} , αποτελεί έναν $(n - k, n)$ γραμμικό μπλοκ κώδικα \mathcal{C}^\perp . Κάθε κωδική λέξη του \mathcal{C} είναι ορθογώνια με κάθε κωδική λέξη του \mathcal{C}^\perp και αντιστρόφως. Οποιοσδήποτε πίνακας γεννήτορας του \mathcal{C}^\perp αποτελεί έναν *πίνακα ελέγχου ισοτιμίας* (parity check matrix) \mathbf{H} για τον \mathcal{C} . Χρησιμοποιώντας τον \mathbf{H} μπορούμε εύκολα να διαπιστώσουμε εάν μία ληφθείσα ακολουθία \mathbf{r} είναι κωδική λέξη, υπολογίζοντας το *σύνδρομο* (syndrome) της

$$\mathbf{s} = \mathbf{rH}^T. \quad (2.9)$$

Η \mathbf{r} είναι κωδική λέξη αν και μόνο αν το σύνδρομο της είναι ίσο με το μηδενικό διάνυσμα $n - k$ συνιστωσών.

Ορισμός 2.2. Το *βάρος Hamming* (Hamming weight) μιας ακολουθίας \mathbf{v} είναι το πλήθος των μη μηδενικών συνιστωσών της, και συμβολίζεται με $wt(\mathbf{v})$.

Ορισμός 2.3. Ως *απόσταση Hamming* (Hamming distance) μεταξύ δύο ακολουθιών \mathbf{v} και \mathbf{w} ορίζεται το πλήθος των συνιστωσών στις οποίες διαφέρουν, και συμβολίζεται με $d(\mathbf{v}, \mathbf{w})$.

Εφόσον τα στοιχεία μιας ακολουθίας παίρνουν τιμές στο F_2 , ισχύει ότι $d(\mathbf{v}, \mathbf{w}) = wt(\mathbf{v} + \mathbf{w})$.

Η σημαντικότερη παράμετρος, η οποία καθορίζει την απόδοση ενός (n, k) γραμμικού μπλοκ κώδικα \mathcal{C} είναι η *ελάχιστη απόσταση* (minimum distance) d_{min} μεταξύ δύο κωδικών του λέξεων. Δηλαδή,

$$d_{min} = \min\{d(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in \mathcal{C}, \mathbf{v} \neq \mathbf{w}\}. \quad (2.10)$$

Το άθροισμα δύο κωδικών λέξεων ενός (n, k) γραμμικού μπλοκ κώδικα \mathcal{C} είναι επίσης κωδική λέξη. Έτσι λοιπόν, η ελάχιστη απόσταση του \mathcal{C} ισούται με το ελάχιστο βάρος Hamming wt_{min} που μπορεί να έχει μία μη μηδενική κωδική του λέξη.

Το πλήθος των σφαλμάτων που μπορεί να ανιχνεύσει ο \mathcal{C} (error detection capability) είναι ίσο με $d_{min} - 1$. Επίσης, το πλήθος των σφαλμάτων που μπορεί να διορθώσει (random error correction capability) ο \mathcal{C} , θεωρώντας ότι η μετάδοση γίνεται μέσω του BSC, είναι ίσο με $\lfloor \frac{d_{min}-1}{2} \rfloor$.

Γενικότερα, η αποκωδικοποίηση ενός (n, k) γραμμικού μπλοκ κώδικα γίνεται με τη χρήση της *τυπικής διάταξης* (standard array) [17], και μπορεί να απλουστευτεί, εάν λάβουμε υπόψιν μας τα σύνδρομα των ληφθέντων ακολουθιών (syndrome decoding). Ειδικότερα, πιο αποδοτι-

κοί αλγόριθμοι μπορούν να χρησιμοποιηθούν εάν ληφθούν υπόψιν τα επιπρόσθετα αλγεβρικά χαρακτηριστικά της εκάστοτε κατηγορίας κωδίκων, όπως π.χ. ο αλγόριθμος Berlekamp-Massey στην περίπτωση των κωδίκων BCH και Reed-Solomon [6].

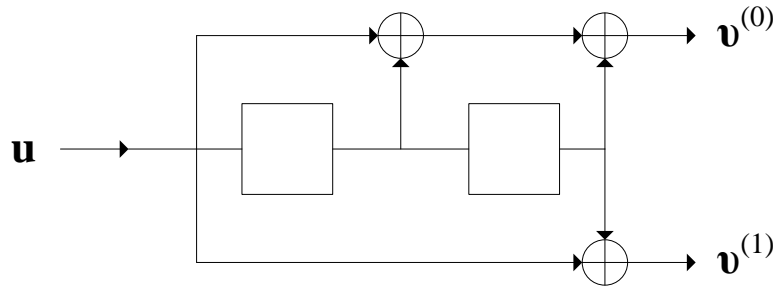
2.2 Συγκεραστικοί Κώδικες

Οι συγκεραστικοί κώδικες παράγονται από γραμμικά ακολουθιακά κυκλώματα. Η ακολουθία πληροφορίας εισάγεται σειριακά στον κωδικοποιητή (έτσι θα αποκαλούμε πλέον τα εν λόγω ακολουθιακά κυκλώματα) και η κωδική ακολουθία προκύπτει από τους γραμμικούς συνδυασμούς των bit πληροφορίας. Στο Σχ. 2.2 απεικονίζεται ένας συγκεραστικός κωδικοποιητής. Τη χρονική στιγμή t εισάγεται στον κωδικοποιητή το bit u_t της ακολουθίας πληροφορίας $\mathbf{u} = (u_0, u_1, \dots)$, ενώ βρίσκονται ήδη αποθηκευμένα τα bit u_{t-1} και u_{t-2} . Ο κωδικοποιητής έχει δύο εξόδους οι οποίες παράγουν τις κωδικές ακολουθίες $\mathbf{v}^{(0)} = (v_0^{(0)}, v_1^{(0)}, \dots)$ και $\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, \dots)$. Τη χρονική στιγμή t τα $v_t^{(0)}$ και $v_t^{(1)}$ προκύπτουν από τις σχέσεις

$$\begin{aligned} v_t^{(0)} &= \sum_{i=0}^2 u_{t-i} g_i^{(0)} \\ v_t^{(1)} &= \sum_{i=0}^2 u_{t-i} g_i^{(1)} \end{aligned} \quad (2.11)$$

όπου το $g_i^{(j)}$ είναι ίσο με 1 εάν το i στοιχείο μνήμης συνδέεται με τον αθροιστή της εξόδου j και ίσο με 0 σε διαφορετική περίπτωση. Γενικότερα, τα u_t , $v_t^{(j)}$ και $g_i^{(j)}$ παίρνουν τιμές (στην παρούσα διατριβή και στην συντριπτική πλειοψηφία της βιβλιογραφίας) στο F_2 (δηλαδή οι προσθέσεις γίνονται σε mod2). Οι ακολουθίες $\mathbf{g}^{(0)} = (g_0^{(0)} \ g_1^{(0)} \ g_2^{(0)})$ και $\mathbf{g}^{(1)} = (g_0^{(1)} \ g_1^{(1)} \ g_2^{(1)})$ καλούνται *ακολουθίες γεννήτορες* (generator sequences), ή απλά γεννήτορες, και επί της ουσίας είναι η *κρουστική απόκριση* (impulse response) μεταξύ της εισόδου και της πρώτης εξόδου και μεταξύ της εισόδου και της δεύτερης εξόδου αντίστοιχα. Για τον κωδικοποιητή του Σχ. 2.2, $\mathbf{g}^{(0)} = (1 \ 1 \ 1)$ και $\mathbf{g}^{(1)} = (1 \ 0 \ 1)$. Οι ακολουθίες εξόδου είναι αποτέλεσμα του συγκερασμού της ακολουθίας εισόδου και των γεννητόρων, γι' αυτό και οι εν λόγω κώδικες λέγονται συγκεραστικοί. Οι δύο ακολουθίες εξόδου πολυπλέκονται, προκειμένου να αποσταλούν σειριακά στο κανάλι. Για κάθε bit πληροφορίας παράγονται δύο κωδικά bit, οπότε ο ρυθμός R του κωδικοποιητή είναι ίσος με $1/2$.

Στη γενικότερη του μορφή, ένας συγκεραστικός κωδικοποιητής μπορεί να έχει k εισόδους και n εξόδους. Σ' αυτήν την περίπτωση, η ακολουθία πληροφορίας χωρίζεται σε k επιμέρους ακολουθίες $\mathbf{u}^{(0)}, \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k-1)}$, κάθε μία εκ των οποίων τροφοδοτεί την αντίστοιχη είσοδο. Τη χρονική στιγμή t εισάγεται στον κωδικοποιητή το διάνυσμα $\mathbf{u}_t = (u_t^{(0)}, u_t^{(1)}, \dots, u_t^{(k-1)})$, και παράγεται η $\mathbf{v}_t = (v_t^{(0)}, v_t^{(1)}, \dots, v_t^{(n-1)})$, όπου $\mathbf{v}^{(j)} = (v_0^{(j)}, v_1^{(j)}, \dots)$, $0 \leq j \leq n-1$, η ακολουθία της j -οστής εξόδου. Στη γενική της περίπτωση, η διαδικασία της συγκεραστικής



Σχήμα 2.2. Συγκεραστικός κωδικοποιητής ρυθμού $R = 1/2$.

κωδικοποίησης περιγράφεται από τη σχέση

$$\mathbf{v}_t = \sum_{i=0}^{\nu} \mathbf{u}_{t-i} \mathbf{G}_i, \quad t \geq 0 \quad (2.12)$$

όπου

$$\mathbf{G}_i = \begin{bmatrix} g_{0,i}^{(0)} & g_{0,i}^{(1)} & \cdots & g_{0,i}^{(n-1)} \\ g_{1,i}^{(0)} & g_{1,i}^{(1)} & \cdots & g_{1,i}^{(n-1)} \\ \vdots & \vdots & & \vdots \\ g_{k-1,i}^{(0)} & g_{k-1,i}^{(1)} & \cdots & g_{k-1,i}^{(n-1)} \end{bmatrix} \quad (2.13)$$

και $\mathbf{g}_j^{(l)} = (g_{j,0}^{(l)}, g_{j,1}^{(l)}, \dots, g_{j,\nu_j}^{(l)})$ είναι η κρουστική απόκριση της l -οστής εξόδου στην j -οστή είσοδο. Με ν_j συμβολίζουμε το μήκος του καταχωρητή της j -οστής εισόδου (δηλαδή το πλήθος των στοιχείων μνήμης που τον απαρτίζουν), ενώ ν είναι το μεγαλύτερο εκ των ν_j . Το σύνολο όλων των ακολουθιών εξόδου αποτελεί τον συγκεραστικό κώδικα.

Εάν θεωρήσουμε ότι $\mathbf{u} = (\mathbf{u}_0, \mathbf{u}_1, \dots)$ κι ότι $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \dots)$, η διαδικασία της κωδικοποίησης περιγράφεται εναλλακτικά και από τη σχέση

$$\mathbf{v} = \mathbf{u} \mathbf{G}_{scalar} \quad (2.14)$$

όπου ο ημι-άπειρος (με την έννοια ότι το μήκος της ακολουθία πληροφορίας μπορεί να είναι οσοδήποτε μεγάλο) βαθμιδωτός πίνακας

$$\mathbf{G}_{scalar} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_\nu & & \\ & \mathbf{G}_0 & \cdots & \mathbf{G}_{\nu-1} & \ddots & \\ & & \ddots & \vdots & & \\ & & & \mathbf{G}_0 & & \\ & & & & & \ddots \end{bmatrix} \quad (2.15)$$

καλείται και πίνακας γεννήτορας στο πεδίο του χρόνου.

Η πράξη του συγκερασμού μεταξύ διακριτών ακολουθιών στο πεδίο του χρόνου μετατρέπεται σε πολλαπλασιασμό πολυωνύμων στο πεδίο κάποιου μετασχηματισμού. Οι ακολουθίες γεννήτορες, οι ακολουθίες πληροφορίας και οι ακολουθίες εξόδου μπορούν να περιγραφούν ως πολυώνυμα, χρησιμοποιώντας τον *τελεστή καθυστέρησης* (delay operator) D (η παραπάνω διαδικασία καλείται και μετασχηματισμός- D) [17, 47, 48]. Έτσι, η διαδικασία της κωδικοποίησης περιγράφεται και από τη σχέση

$$\mathbf{V}(D) = \mathbf{U}(D)\mathbf{G}(D) \quad (2.16)$$

όπου $\mathbf{U}(D) = (\mathbf{U}^{(0)}(D), \mathbf{U}^{(1)}(D), \dots, \mathbf{U}^{(k-1)}(D))$ και $\mathbf{U}^l(D) = \sum u_t^{(j)} D^t$. Αντίστοιχα, $\mathbf{V}(D) = (\mathbf{V}^{(0)}(D), \mathbf{V}^{(1)}(D), \dots, \mathbf{V}^{(n-1)}(D))$ και $\mathbf{V}^l(D) = \sum v_t^{(l)} D^t$. Τέλος,

$$\mathbf{G}(D) = \sum_{i=0}^{\nu} \mathbf{G}_i D^i = \begin{bmatrix} g_0^{(0)}(D) & g_0^{(1)}(D) & \dots & g_0^{(n-1)}(D) \\ g_1^{(0)}(D) & g_1^{(1)}(D) & \dots & g_1^{(n-1)}(D) \\ \vdots & \vdots & & \vdots \\ g_{k-1}^{(0)}(D) & g_{k-1}^{(1)}(D) & \dots & g_{k-1}^{(n-1)}(D) \end{bmatrix} \quad (2.17)$$

όπου $g_j^{(l)}(D) = \sum_{i=0}^{\nu_j} g_{j,i}^{(l)} D^i$. Ο $\mathbf{G}(D)$ καλείται πίνακας γεννήτορας στο πεδίο του μετασχηματισμού- D . Στο υπόλοιπο της διατριβής θα τον αποκαλούμε απλά πίνακα γεννήτορα. Ο πίνακας γεννήτορας που αντιστοιχεί στον κωδικοποιητή του Σχ. 2.2 είναι ο

$$\mathbf{G}(D) = (1 + D + D^2 \quad 1 + D^2). \quad (2.18)$$

Ορισμός 2.4. Η μνήμη της εισόδου j του κωδικοποιητή ορίζεται ως

$$\nu_j = \max_{0 \leq l \leq n-1} \{\deg(g_j^{(l)}(D))\}. \quad (2.19)$$

Ορισμός 2.5. Η τάξη της μνήμης (memory order) του κωδικοποιητή ορίζεται ως

$$\nu = \max_{0 \leq j \leq k-1} \{\nu_j\}. \quad (2.20)$$

Ορισμός 2.6. Η συνολική μνήμη (overall memory) του κωδικοποιητή ορίζεται ως

$$m = \sum_{j=0}^{k-1} \nu_j. \quad (2.21)$$

Ο πίνακας γεννήτορας $\mathbf{G}(D)$ είναι η συνάρτηση μεταφοράς του κωδικοποιητή, γι' αυτό στο υπόλοιπο της διατριβής δεν θα διαχωρίζουμε τις δύο αυτές έννοιες (τον κωδικοποιητή και τον πίνακα γεννήτορα).

Ένας συγκεραστικός κώδικας \mathcal{C} μπορεί να παραχθεί από διάφορους ισοδύναμους πίνακες γεννήτορες.

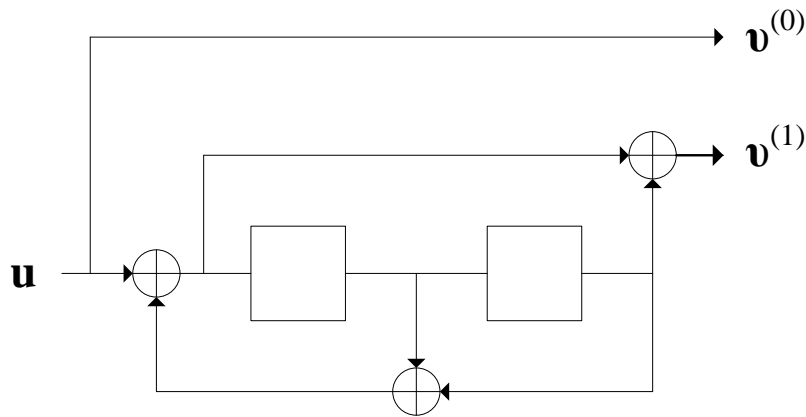
Θεώρημα 2.7. [47, 48, 49] Δύο πίνακες γεννήτορες $\mathbf{G}(D)$ και $\mathbf{G}'(D)$ ρυθμού $R = k/n$ είναι ισοδύναμοι αν και μόνο αν υπάρχει ένας $k \times k$ πίνακας $\mathbf{T}(D)$ μη μηδενικής ορίζουσας, του οποίου τα στοιχεία βρίσκονται στο $F_2(D)$, τέτοιος ώστε $\mathbf{G}(D) = \mathbf{T}(D)\mathbf{G}'(D)$.

Το $F_2(D)$ είναι το πεδίο που απαρτίζεται από όλα τα στοιχεία $P(D)/Q(D)$, όπου τα $P(D)$, $Q(D)$ είναι πολυώνυμα με συντελεστές στο F_2 , και $Q(D) \neq 0$.

Ανάμεσα στους πολλούς πίνακες γεννήτορες που μπορούν να παράγουν τον \mathcal{C} , υπάρχουν κάποιοι που επιδιώκουμε να κατασκευάσουμε, και κάποιοι που πρέπει να αποφεύγονται. Καταρχάς, τα στοιχεία ενός πίνακα γεννήτορα, εκτός από πολυώνυμα, μπορούν να είναι και ρητές συναρτήσεις, δηλαδή λόγοι πολυωνύμων $P(D)/Q(D)$. Ο παρονομαστής υποδηλώνει την ύπαρξη ανάδρασης στο ακολουθιακό κύκλωμα. Για παράδειγμα, ο πολυωνυμικός πίνακας γεννήτορας $\mathbf{G}(D)$ στην (2.18) γράφεται

$$\mathbf{G}(D) = (1 + D + D^2) \left(1 \frac{1 + D^2}{1 + D + D^2} \right) = \mathbf{T}(D)\mathbf{G}_{sys}(D). \quad (2.22)$$

Ο κωδικοποιητής που αντιστοιχεί στον $\mathbf{G}_{sys}(D)$ απεικονίζεται στο Σχ. (2.3), και ονομάζεται αναδρομικός συστηματικός κωδικοποιητής (recursive systematic encoder). Ονομάζεται αναδρομικός εξαιτίας του βρόγχου ανάδρασης, και συστηματικός διότι κάθε χρονική στιγμή t , το κωδικό bit $v_t^{(0)}$ είναι το bit πληροφορίας u_t . Όπως θα δούμε και παρακάτω, η κατηγορία των αναδρομικών συγκεραστικών κωδικοποιητών είναι ιδιαίτερος σημαντική για την κατασκευή τούρμπο κωδίκων (turbo codes) [34].



Σχήμα 2.3. Αναδρομικός συστηματικός συγκεραστικός κωδικοποιητής ρυθμού $R = 1/2$.

Ορισμός 2.8. Ένας πίνακας γεννήτορας ονομάζεται βασικός (basic) εάν είναι πολυωνυμικός και έχει πολυωνυμικό δεξιό αντίστροφο.

Θεώρημα 2.9. [47, 48, 49] Ένας πολυωνυμικός πίνακας $\mathbf{G}(D)$ είναι βασικός αν και μόνο αν ο μέγιστος κοινός διαιρέτης (gcd) των οριζουσών όλων των $k \times k$ υποπινάκων του είναι ίσος με 1.

Ορισμός 2.10. Ο πολυωνυμικός πίνακας γεννήτορας του οποίου η “προφανής” υλοποίηση απαιτεί τα λιγότερα στοιχεία μνήμης από όλους τους ισοδύναμους πίνακες γεννήτορες ονομάζεται *κανονικός* (canononical) ή *ελάχιστος-βασικός* (minimal-basic).

Ως προφανής υλοποίηση ορίστηκε από τον Forney η *κανονική μορφή ελεγχτή* (controller canonical form). Κάθε συγκεραστικός κώδικας \mathcal{C} μπορεί να παραχθεί από έναν κανονικό πίνακα γεννήτορα [47, 48, 49], και κάθε ρητός πίνακας γεννήτορας είναι ισοδύναμος με έναν κανονικό. Όπως είναι λογικό, επιθυμούμε να παράγουμε έναν συγκεραστικό κώδικα χρησιμοποιώντας έναν κανονικό πίνακα γεννήτορα, αφού απαιτεί τα λιγότερα στοιχεία μνήμης (δηλαδή έχει τη μικρότερη δυνατή συνολική μνήμη απ’ όλους τους ισοδύναμους γεννήτορες).

Πριν προχωρήσουμε στις ικανές και αναγκαίες συνθήκες που πρέπει να ικανοποιούνται προκειμένου ένας γεννήτορας πίνακας να είναι κανονικός, θα ορίσουμε τον δυαδικό πίνακα \mathbf{G}_h ενός πίνακα γεννήτορα $\mathbf{G}(D)$. Το στοιχείο του \mathbf{G}_h που βρίσκεται στην j γραμμή και την l στήλη είναι ο συντελεστής του όρου ν_j του πολυωνύμου $g_j^{(l)}(D)$.

Θεώρημα 2.11. [47, 48, 49] Ένας πολυωνυμικός πίνακας $\mathbf{G}(D)$ είναι κανονικός, αν και μόνο αν είναι βασικός, και ο αντίστοιχος πίνακας \mathbf{G}_h είναι βαθμού (rank) k .

Ορισμός 2.12. Ένας πίνακας γεννήτορας λέγεται *καταστροφικός* εάν υπάρχει ακολουθία πληροφορίας απείρου βάρους Hamming, η οποία αντιστοιχίζεται σε κωδική ακολουθία πεπερασμένου βάρους.

Ο καταστροφικός πίνακας γεννήτορας θα πρέπει να αποφεύγεται για την παραγωγή του \mathcal{C} , καθώς πεπερασμένος αριθμός σφαλμάτων από το κανάλι μπορεί να οδηγήσει σε άπειρο αριθμό σφαλμάτων κατά την αποκωδικοποίηση.

Θεώρημα 2.13. [47, 48, 49] Ένας πίνακας γεννήτορας $\mathbf{G}(D)$ δεν είναι καταστροφικός αν και μόνο αν έχει πολυωνυμικό δεξιό ψευδοαντίστροφο¹ πίνακα.

Θεώρημα 2.14. [47, 48, 49] Ένας πολυωνυμικός πίνακας γεννήτορας $\mathbf{G}(D)$ δεν είναι καταστροφικός αν και μόνο αν ο μέγιστος κοινός διαιρέτης (gcd) των οριζουσών όλων των $k \times k$ υποπινάκων του είναι ίσος με D^l , για $l \geq 0$.

Σε πλήρη αντιστοιχία με τους μπλοκ κώδικες, ορίζεται ο πίνακας ελέγχου ισοτιμίας και για τους συγκεραστικούς κώδικες.

Ορισμός 2.15. Ο πίνακας ελέγχου ισοτιμίας $\mathbf{H}(D)$ ενός συγκεραστικού κώδικα \mathcal{C} που παράγεται από έναν πίνακα γεννήτορα $\mathbf{G}(D)$ είναι ένας $(n - k) \times n$ πίνακας για τον οποίο ισχύει $\mathbf{G}(D)\mathbf{H}(D) = \mathbf{0}$.

Ο $\mathbf{H}(D)$ παράγει τον $(n, n - k)$ δυϊκό συγκεραστικό κώδικα \mathcal{C}^\perp του \mathcal{C} .

Τα παραπάνω αποτελούν μια ανασκόπηση των σημαντικότερων συγκεραστικών κωδικοποιητών. Ο κώδικας που παράγεται από έναν $k \times n$ πίνακα γεννήτορα θα λέγεται (n, k) συγκεραστικός κώδικας \mathcal{C} . Η συνολική μνήμη m ενός κωδικοποιητή είναι χαρακτηριστικό του ίδιου του κωδικοποιητή. Η ελάχιστη όμως μνήμη που μπορεί να έχει ένας κωδικοποιητής ο

¹Δεξιός ψευδοαντίστροφος του $\mathbf{G}(D)$ είναι ο πίνακας $\mathbf{G}^{-1}(D)$ για τον οποίο ισχύει $\mathbf{G}(D)\mathbf{G}^{-1}(D) = D^l$, για $l \geq 0$.

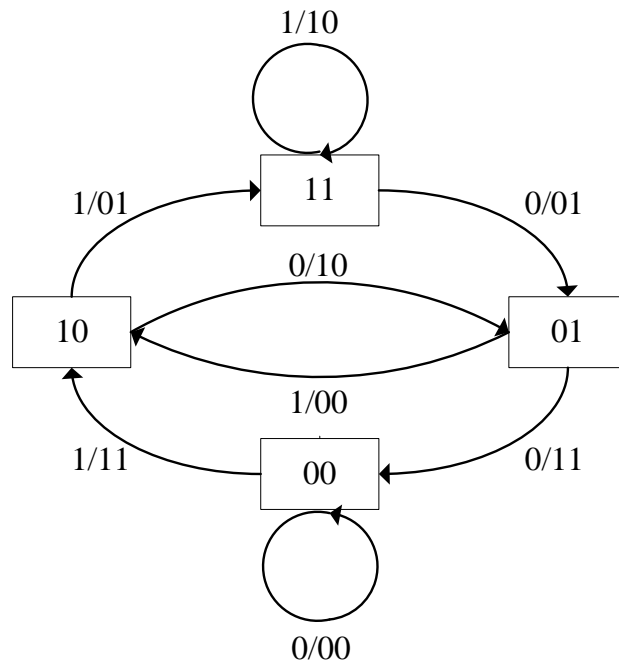
οποίος παράγει τον C χαρακτηρίζει πλέον τον ίδιο τον κώδικα. Έτσι, εάν ο C παράγεται από έναν κανονικό πίνακα γεννήτορα με μήμη m , θα τον περιγράψουμε ως (n, k, m) συγκεραστικό κώδικα. Πολλές φορές βέβαια στη βιβλιογραφία, ένας κώδικας χαρακτηρίζεται ως (n, k, m) όταν παράγεται από έναν πίνακα γεννήτορα με μήμη m , χωρίς να είναι γνωστό εάν ο τελευταίος είναι κανονικός. Ενδεχομένως να χρησιμοποιήσουμε κι εδώ, σε κάποια σημεία που μας βολεύει, αυτόν τον “χαλαρό” συμβολισμό.

2.2.1 Γραφικές Αναπαραστάσεις

Ο συγκεραστικός κωδικοποιητής είναι ένα γραμμικό ακολουθιακό κύκλωμα το οποίο υλοποιείται χρησιμοποιώντας καταχωρητές ολίσθησης, και η λειτουργία του μπορεί να περιγραφεί από ένα *διάγραμμα καταστάσεων* (state diagram). Το περιεχόμενο των καταχωρητών ονομάζεται *κατάσταση* (state) του κωδικοποιητή. Δηλαδή, η κατάσταση ενός (n, k, m) συγκεραστικού κωδικοποιητή τη χρονική στιγμή t είναι η πλειάδα

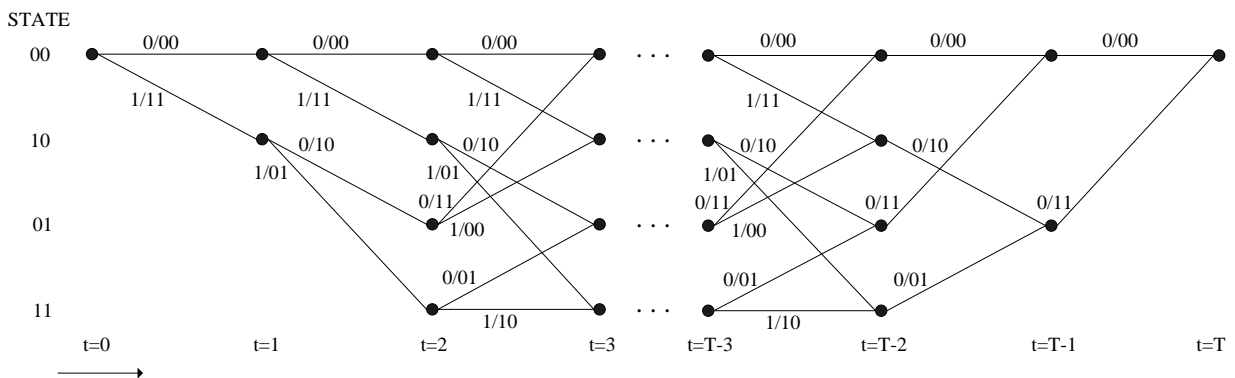
$$\mathbf{S}_t = \left[S_{t,1}^{(0)} \quad S_{t,2}^{(0)} \quad \dots \quad S_{t,\nu_0}^{(0)} \quad S_{t,1}^{(1)} \quad S_{t,2}^{(1)} \quad \dots \quad S_{t,\nu_1}^{(1)} \quad \dots \quad S_{t,1}^{(k-1)} \quad S_{t,2}^{(k-1)} \quad \dots \quad S_{t,\nu_{k-1}}^{(k-1)} \right] \quad (2.23)$$

όπου $S_{t,i}^{(l)}$ είναι το περιεχόμενο της i -οστής “θέσης” του καταχωρητή της l -οστής εισόδου τη χρονική στιγμή t . Το τμήμα \mathbf{u}_t της ακολουθίας πληροφορίας θα οδηγήσει τον κωδικοποιητή από την κατάσταση \mathbf{S}_t στην κατάσταση \mathbf{S}_{t+1} , και θα παράγει το τμήμα \mathbf{v}_t της κωδικής ακολουθίας. Στο Σχ. 2.4 απεικονίζεται το διάγραμμα καταστάσεων του κωδικοποιητή στο Σχ. 2.2. Οι μεταβάσεις μεταξύ καταστάσεων συμβολίζονται με βέλη, ενώ σε κάθε μετάβαση σημειώνεται η τιμή του bit πληροφορίας που την προκάλεσε και οι τιμές των κωδικών bit που παράχθηκαν με τη μορφή u/v . Η διαδικασία της κωδικοποίησης μιας ακολουθίας πληροφορίας μπορεί να αναπαρασταθεί γραφικά, εάν συνδυάσουμε το διάγραμμα καταστάσεων του κωδικοποιητή και τη διάσταση του χρόνου, κατασκευάζοντας έτσι το *διάγραμμα trellis* (trellis diagram). Με το διάγραμμα trellis, κάθε χρονική στιγμή αναπαρίσταται με ένα ξεχωριστό διάγραμμα καταστάσεων. Καταρχάς να διευκρινίσουμε εδώ πως η διαδικασία της κωδικοποίησης ξεκινάει θεωρώντας ότι ο κωδικοποιητής βρίσκεται στην κατάσταση μηδέν. Όταν εισαχθεί όλη η ακολουθία πληροφορίας στον κωδικοποιητή, ο τελευταίος θα οδηγηθεί πάλι στην κατάσταση μηδέν, εισάγοντας μία ακολουθία από bit κατάλληλης τιμής (για ακολουθία από μηδενικά στην περίπτωση που ο κωδικοποιητής είναι μη αναδρομικός), το μήκος της οποίας εξαρτάται από τη μήμη του κωδικοποιητή. Γι’ αυτό το λόγο, στα πρώτα και στα τελευταία τμήματα του διαγράμματος trellis δεν εμφανίζονται όλες οι πιθανές καταστάσεις. Στο Σχ. 2.5 απεικονίζεται το διάγραμμα trellis του κωδικοποιητή του Σχ. 2.2, θεωρώντας ότι κωδικοποιούμε μια ακολουθία πληροφορίας μήκους T . Σε κάθε χρονική στιγμή t εμφανίζονται όλες οι δυνατές καταστάσεις που μπορεί να βρίσκεται ο κωδικοποιητής. Από κάθε κατάσταση “αναχωρούν” *ακμές* (edges) (ή *κλαδιά* (branches)) που συμβολίζουν τις δυνατές μεταβάσεις, οι οποίες θα οδηγήσουν τον κωδικοποιητή σε νέα



Σχήμα 2.4. Διάγραμμα καταστάσεων του συγκεραστικού κωδικοποιητή στο Σχ. 2.2.

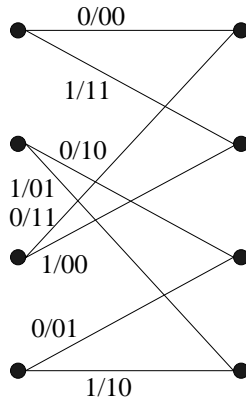
κατάσταση τη χρονική στιγμή $t + 1$. Όπως και στο διάγραμμα καταστάσεων, σε κάθε ακμή σημειώνεται η τιμή των bit πληροφορίας που προκάλεσαν την αντίστοιχη μετάβαση, καθώς και η τιμή των κωδικών bit που παράγονται (με τη μορφή u_t/v_t). Οι τιμές των u_t και v_t στις ακμές του διαγράμματος trellis λέγονται και *ετικέτες εισόδου και εξόδου* (input-output labels) αντίστοιχα. Όπως μπορούμε να παρατηρήσουμε, το διάγραμμα trellis αποτελείται (εξαιρώντας μια μικρή περίοδο στην αρχή και στο τέλος) από την επανάληψη της δομής που φαίνεται στο Σχ. 2.6, η οποία ονομάζεται *trellis module*. Μία ακολουθία ακμών που ξεκινάει από τη



Σχήμα 2.5. Διάγραμμα trellis του συγκεραστικού κωδικοποιητή στο Σχ. 2.2.

μηδενική κατάσταση τη χρονική στιγμή $t = 0$ και καταλήγει επίσης στη μηδενική κατάσταση τη χρονική στιγμή $t = T$ λέγεται *κωδικό μονοπάτι* (code path), ενώ η ακολουθία των ετικετών

εισόδου των εν λόγω ακμών αποτελεί την ακολουθία πληροφορίας (οι δύο τελευταίες ετικέτες δεν αντιστοιχούν σε bit πληροφορίας, αλλά σε μηδενικά bit που οδήγησαν τον κωδικοποιητή στην κατάσταση μηδέν) η οποία οδήγησε στην επιλογή της συγκεκριμένης “διαδρομής” στο διάγραμμα trellis. Η ακολουθία των ετικετών εξόδου αποτελούν την αντίστοιχη κωδική λέξη. Το διάγραμμα trellis περιλαμβάνει όλες τις 2^T πιθανές λέξεις πληροφορίας, και προφανώς όλες τις αντίστοιχες κωδικές λέξεις. Παρόλο που κατασκευάσαμε το διάγραμμα trellis χρησιμοποιώντας ένα συγκεκριμένο συγκεραστικό κωδικοποιητή, εάν αγνοήσουμε τις ετικέτες εισόδου, το εν λόγω διάγραμμα περιγράφει (χαρακτηρίζει) τον παραγόμενο συγκεραστικό κώδικα. Το διά-



Σχήμα 2.6. Trellis module του συγκεραστικού κωδικοποιητή στο Σχ. 2.2.

γραμμα trellis των συγκεραστικών κωδίκων θα αναλυθεί εκτενέστερα στα επόμενα κεφάλαια, καθώς αποτελεί βασικό αντικείμενο μελέτης της παρούσας διατριβής.

2.2.2 Ελεύθερη Απόσταση–Φάσματα Βάρους των Συγκεραστικών Κωδίκων

Η *ελεύθερη απόσταση* (free distance) d_f ενός συγκεραστικού κώδικα είναι το ελάχιστο βάρος Hamming που μπορεί να έχει μία (μη μηδενική) κωδική του λέξη, δηλαδή μια λέξη που αντιστοιχεί σε μονοπάτι το οποίο αναχωρεί από τη μηδενική κατάσταση και καταλήγει πάλι σ’αυτήν.

Όπως θα δούμε παρακάτω, η ελεύθερη απόσταση ενός συγκεραστικού κώδικα παίζει σημαντικό ρόλο στην απόδοση του, και επιθυμούμε να έχει όσο το δυνατόν μεγαλύτερη τιμή. Συνεπώς, είναι σημαντικό να γνωρίζουμε τη μεγαλύτερη τιμή της d_f που μπορούμε να πετύχουμε για συγκεκριμένες παραμέτρους (n, k, m) . Στη συνέχεια παρουσιάζουμε ένα άνω φράγμα για την ελεύθερη απόσταση των συγκεραστικών κωδίκων που παρουσίασε ο Heller στο [50] για ρυθμούς $1/n$, και γενικεύτηκε στο [48].

Θεώρημα 2.16. *Η ελεύθερη απόσταση ενός συγκεραστικού κώδικα με ρυθμό k/n ο οποίος παράγεται από έναν κωδικοποιητή με τάξη μνήμης ν και συνολική μνήμη m ικανοποιεί την*

ανισότητα

$$d_f \leq \min_{i \geq 1} \left\{ \left\lfloor \frac{(m+i)n}{2(1-2^{\nu-k(\nu+i)})} \right\rfloor \right\}. \quad (2.24)$$

Το παραπάνω φράγμα προκύπτει φράσσοντας την ελάχιστη απόσταση d_{min} του γραμμικού μπλοκ κώδικα που αποτελείται από τα κωδικά μονοπάτια που καταλήγουν στη μηδενική κατάσταση τη χρονική στιγμή $m+i$, για $i > 1$, με το φράγμα του Plotkin για (\tilde{n}, \tilde{k}) μπλοκ κώδικες (όχι απαραίτητα γραμμικούς)

$$d_{min} \leq \left\lfloor \frac{\tilde{n}2^{\tilde{k}}}{2(2^{\tilde{k}}-1)} \right\rfloor. \quad (2.25)$$

Σε κάποιες περιπτώσεις, μπορεί να γίνουν μικρές βελτιώσεις αν χρησιμοποιήσουμε το φράγμα του Griesmer για την ελάχιστη απόσταση των (\tilde{n}, \tilde{k}) γραμμικών μπλοκ κωδίκων [51]

$$\sum_{i=1}^{\tilde{k}-1} \left\lceil \frac{d_{min}}{2^i} \right\rceil \leq \tilde{n}. \quad (2.26)$$

Πέραν από τις κωδικές λέξεις ελαχίστου βάρους, την απόδοση ενός συγκεραστικού κώδικα επηρεάζουν και οι λέξεις μεγαλύτερων βαρών, ειδικά στις χαμηλές τιμές του σηματοθορυβικού λόγου. Συμβολίζουμε με A_d το πλήθος των κωδικών λέξεων βάρους d (προφανώς $d \geq d_f$), οι οποίες αντιστοιχούν σε μονοπάτια που αναχωρούν από τη μηδενική κατάσταση μια αρχική χρονική στιγμή $t = 0$, και επιστρέφουν σ' αυτήν μόνο μια φορά. Το σύνολο των A_d καλείται *φάσμα βάρους* (weight spectrum) του κώδικα. Επίσης, συμβολίζουμε με I_d το πλήθος των μη μηδενικών bit των λέξεων πληροφορίας που αντιστοιχούν στις κωδικές λέξεις βάρους d . Τις ακολουθίες των A_d και I_d θα τις αποκαλούμε και απλά ως φάσματα του κώδικα. Αξίζει να σημειωθεί εδώ πως τα A_d είναι χαρακτηριστικό του κώδικα, ενώ τα I_d εξαρτώνται και από τον κωδικοποιητή, δηλαδή από τον τρόπο με τον οποίο αντιστοιχίζονται λέξεις πληροφορίας σε κωδικές λέξεις.

Τα φάσματα ενός συγκεραστικού κώδικα μπορούν να υπολογιστούν χρησιμοποιώντας ένα τροποποιημένο διάγραμμα καταστάσεων, με τη χρήση του οποίου προκύπτει η *συνάρτηση εισόδου-εξόδου απαρίθμησης βάρους* (input-output weight enumerator function-IOWEF) [17]

$$T(X, W, L) = \sum_{d,w,l} A_{d,w,l} X^d W^w L^l \quad (2.27)$$

όπου τα X , W και L είναι ψευδομεταβλητές, ενώ $A_{d,w,l}$ είναι το πλήθος των κωδικών λέξεων οι οποίες έχουν βάρους d , προκύπτουν από ακολουθίες πληροφορίας βάρους w και αντιστοιχούν σε μονοπάτια μήκους L . Ως μήκος ενός μονοπατιού καλείται το πλήθος των ακμών που το απαρτίζουν. Προφανώς, λαμβάνονται υπόψιν μόνο λέξεις των οποίων τα κωδικά μονοπάτια αναχωρούν από τη μηδενική κατάσταση μια αρχική χρονική στιγμή $t = 0$, και επιστρέφουν

σ' αυτήν μόνο μια φορά. Η συνάρτηση $T(X, W, L)$ περιέχει λεπτομερείς πληροφορίες για τα φάσματα ενός συγκεραστικού κώδικα. Αν στην $T(X, W, L)$ αντικαταστήσουμε τις μεταβλητές W και L με 1, τότε προκύπτει η συνάρτηση απαρίθμησης βάρους (weight enumerator function)

$$A(X) = \sum_d A_d X^d \quad (2.28)$$

η οποία απαριθμεί το φάσμα βάρους του κώδικα. Εάν παραγωγίσουμε την $T(X, W, L)$ ως προς W και στη συνέχεια αντικαταστήσουμε τις μεταβλητές W και L με 1, προκύπτει η συνάρτηση

$$I(X) = \sum_d I_d X^d \quad (2.29)$$

η οποία απαριθμεί την ακολουθία των I_d .

Ο αναλυτικός υπολογισμός της συνάρτησης $T(X, W, L)$ ενός κώδικα δυσκολεύει όσο αυξάνεται η μήμη του κωδικοποιητή. Γι' αυτό, στις περισσότερες περιπτώσεις χρησιμοποιούνται τροποποιημένοι αλγόριθμοι αποκωδικοποίησης προκειμένου να υπολογιστούν οι πρώτοι όροι των φασμάτων, οι οποίοι όπως θα δούμε παρακάτω είναι και οι πιο σημαντικοί.

2.2.3 Αποκωδικοποίηση

Ο σημαντικότερος αλγόριθμος αποκωδικοποίησης των συγκεραστικών κωδίκων είναι ο αλγόριθμος του Viterbi [9]. Ο τελευταίος χρησιμοποιεί το διάγραμμα trellis του κώδικα και εντοπίζει την κωδική λέξη μέγιστης πιθανοφάνειας, δηλαδή τη λέξη που μεγιστοποιεί την πιθανότητα $P(\mathbf{r}|\mathbf{v})$. Για ένα κανάλι χωρίς μήμη η τελευταία γίνεται

$$P(\mathbf{r}|\mathbf{v}) = \prod_t \prod_{i=0}^{n-1} P(r_t^{(i)}|v_t^{(i)}). \quad (2.30)$$

Επειδή η λογαριθμική συνάρτηση είναι γνησίως αύξουσα, η κωδική λέξη που μεγιστοποιεί την πιθανότητα $P(\mathbf{r}|\mathbf{v})$ μεγιστοποιεί και τον λογάριθμο της $\log P(\mathbf{r}|\mathbf{v})$. Στις περισσότερες περιπτώσεις θεωρούμε ως μετρική της αποκωδικοποίησης την $\log P(\mathbf{r}|\mathbf{v})$, προκειμένου να μετατρέψουμε τους πολλαπλασιασμούς σε προσθέσεις. Έτσι, για ένα κανάλι χωρίς μήμη επιλέγεται η λέξη που μεγιστοποιεί τη μετρική

$$\log P(\mathbf{r}|\mathbf{v}) = \sum_t \sum_{i=0}^{n-1} \log P(r_t^{(i)}|v_t^{(i)}) \quad (2.31)$$

όπου η $\log P(r_t^{(i)}|v_t^{(i)})$ καλείται μετρική του i -οστού κωδικού bit τη χρονική στιγμή t , ενώ η $\sum_{i=0}^{n-1} \log P(r_t^{(i)}|v_t^{(i)})$ είναι η μετρική ακμής (branch metric), καθώς είναι το άθροισμα των μετρικών των bit που απαρτίζουν την ετικέτα εξόδου μιας ακμής του διαγράμματος trellis.

Κάθε χρονική στιγμή t ο αλγόριθμος του Viterbi υπολογίζει τη μετρική κάθε μίας εκ των 2^{m+k} ακμών. Στη συνέχεια, προσθέτει τις μετρικές των ακμών στις μετρικές των μονοπατιών τα οποία επιλέχθηκαν στο προηγούμενο βήμα προκειμένου να τις ανανεώσει (η μετρική ενός μονοπατιού τη στιγμή $t-1$ είναι $n \sum_{\tau=0}^{t-1} \sum_{i=0}^{n-1} \log P(r_{\tau}^{(i)} | v_{\tau}^{(i)})$). Τέλος, από τα μονοπάτια που καταλήγουν σε κάθε κατάσταση-προορισμό επιλέγεται και αποθηκεύεται το μονοπάτι με τη μεγαλύτερη μετρική.

Η πιθανότητα σφάλματος $P(E)$ κατά την αποκωδικοποίηση φράσσεται από την ποσότητα

$$P(E) < \sum_{d=d_f}^{\infty} A_d P_d \quad (2.32)$$

ενώ η πιθανότητα σφάλματος ενός bit πληροφορίας φράσσεται από την ποσότητα

$$P_b < \frac{1}{k} \sum_{d=d_f}^{\infty} I_d P_d. \quad (2.33)$$

Θεωρώντας, χωρίς βλάβη της γενικότητας, ότι έχει μεταδοθεί η μηδενική κωδική ακολουθία, η P_d είναι η πιθανότητα να επιλεγεί για πρώτη φορά κάποια χρονική στιγμή ένα λανθασμένο μονοπάτι βάρους d εις βάρος του μηδενικού μονοπατιού. Για το BSC η P_d δίνεται από τη σχέση

$$P_d = \begin{cases} \sum_{e=\frac{d+1}{2}}^{\infty} \binom{d}{e} p^e (1-p)^{d-e}, & d \text{ περιττός} \\ \frac{1}{2} \binom{d}{d/2} p^{d/2} (1-p)^{d/2} + \sum_{e=\frac{d}{2}+1}^{\infty} \binom{d}{e} p^e (1-p)^{d-e}, & d \text{ άρτιος.} \end{cases} \quad (2.34)$$

Θεωρώντας ότι το κανάλι μετάδοσης είναι το AWGN και ότι χρησιμοποιείται διαμόρφωση BPSK, η P_d ισούται με

$$P_d = Q \left(\sqrt{\frac{2dRE_b}{N_0}} \right). \quad (2.35)$$

Εάν χρησιμοποιήσουμε την ανισότητα [17]

$$Q(x) < e^{-\frac{x^2}{2}} \quad (2.36)$$

τότε προκύπτουν για την περίπτωση του AWGN καναλιού τα (ελαφρώς πιο χαλαρά) άνω φράγματα

$$P(E) < \sum_{d=d_f}^{\infty} A_d e^{-\frac{dRE_b}{N_0}} \quad (2.37)$$

και

$$P_b(E) < \frac{1}{k} \sum_{d=d_f}^{\infty} I_d e^{-\frac{dRE_b}{N_0}}. \quad (2.38)$$

Όπως είδαμε παραπάνω, ο αλγόριθμος του Viterbi χρησιμοποιεί το κριτήριο μεγίστης πιθα-

νοφάνειας προκειμένου να εκτιμήσει την κωδική λέξη που εστάλη. Στη συνέχεια θα παρουσιάσουμε συνοπτικά τον αλγόριθμο BCJR (Bahl, Cocke, Jelinec, Raviv), ο οποίος είναι αλγόριθμος *μεγίστης a posteriori πιθανότητας* (maximum a posteriori probability MAP), και χρησιμοποιεί κι αυτός το διάγραμμα trellis του κώδικα. Συγκεκριμένα, ο αλγόριθμος BCJR υπολογίζει για κάθε bit πληροφορίας την ποσότητα

$$L(u_t^{(i)}|\mathbf{r}) = \ln \frac{P(u_t^{(i)} = 1|\mathbf{r})}{P(u_t^{(i)} = 0|\mathbf{r})} \quad (2.39)$$

η οποία καλείται *λογάριθμος του λόγου πιθανοφάνειας* (Log Likelihood Ratio-LLR). Η εκτιμώμενη τιμή του bit πληροφορίας $u_t^{(i)}$ προκύπτει από το πρόσημο της LLR, δηλαδή

$$u_t^{(i)} = \begin{cases} 1, & \text{αν } L(u_t^{(i)}) > 0 \\ 0, & \text{αν } L(u_t^{(i)}) < 0. \end{cases} \quad (2.40)$$

Συνοπτικά, ο αλγόριθμος BCJR υπολογίζει κάθε χρονική στιγμή t για κάθε ακμή τη μετρική $\gamma_t(S', S) = P(\mathbf{u}_t)p(\mathbf{r}_t|\mathbf{v}_t)$, όπου S' και S είναι η “αρχική” και η “τελική” κατάσταση της ακμής αντίστοιχα. Με \mathbf{u}_t συμβολίζεται η k -άδα των bit πληροφορίας ενώ με \mathbf{v}_t η n -άδα των κωδικών bit που αντιστοιχούν στη συγκεκριμένη ακμή. $P(\mathbf{u}_t) = P(u_t^{(0)})P(u_t^{(1)}) \dots P(u_t^{(k-1)})$ είναι η *a priori* πιθανότητα της k -άδας \mathbf{u}_t . Θεωρώντας την αντιστοίχιση $0 \rightarrow -1$ και $1 \rightarrow 1$, πολλές φορές είναι προτιμότερο να εκφράσουμε την *a priori* πιθανότητα $P(u_t^{(i)} = \pm 1)$ ως εκθετικό όρο

$$P(u_t^{(i)} = \pm 1) = A_t^{(i)} e^{u_t^{(i)} L_a(u_t^{(i)})/2} \quad (2.41)$$

όπου

$$L_a(u_t^{(i)}) = \ln \frac{P(u_t^{(i)} = +1)}{P(u_t^{(i)} = -1)} \quad (2.42)$$

και

$$A_t^{(i)} = \frac{e^{-L_a(u_t^{(i)})/2}}{1 + e^{-L_a(u_t^{(i)})}}. \quad (2.43)$$

Ο συντελεστής $A_t^{(i)}$ δεν επηρεάζει το αποτέλεσμα της αποκωδικοποίησης, καθώς απλοποιείται σε μεταγενέστερο στάδιο, γι' αυτό και αγνοείται εξ' αρχής.

Στη συνέχεια, υπολογίζεται η *προς τα εμπρός μετρική* (forward metric) για κάθε κατάσταση S και για κάθε χρονική στιγμή $t + 1$

$$\alpha_{t+1}(S) = \sum_{S' \in \sigma_t} \gamma_t(S', S) \alpha_t(S') \quad (2.44)$$

όπου σ_t το σύνολο των καταστάσεων τη χρονική στιγμή t που συνδέονται με την S τη χρονική στιγμή $t + 1$, και η *προς τα πίσω μετρική* (backward metric) για κάθε κατάσταση S' και για

κάθε χρονική στιγμή t

$$\beta_t(S') = \sum_{S \in \sigma_{t+1}} \gamma_t(S', S) \beta_{t+1}(S) \quad (2.45)$$

όπου σ_{t+1} το σύνολο των καταστάσεων τη χρονική στιγμή $t + 1$ που συνδέονται με την S τη χρονική στιγμή t . Η $L(u_t^{(i)} | \mathbf{r})$ δίνεται από τη σχέση

$$L(u_t^{(i)} | \mathbf{r}) = \frac{\sum_{(S', S) \in B_t^+} \beta_{t+1}(S) \gamma_t(S', S) \alpha_t(S')}{\sum_{(S', S) \in B_t^-} \beta_{t+1}(S) \gamma_t(S', S) \alpha_t(S')} \quad (2.46)$$

όπου B_t^+ και B_t^- είναι τα σύνολα των μεταβάσεων από μία κατάσταση S' τη χρονική στιγμή t σε μία κατάσταση S τη χρονική στιγμή $t + 1$, για τις οποίες η τιμή του $u_t^{(i)}$ είναι 1 και 0 αντίστοιχα.

Συνήθως, προκειμένου να αποφύγουμε τους πολύπλοκους πολλαπλασιασμούς μετασχηματίζουμε τον αλγόριθμο του BCJR στο λογαριθμικό πεδίο (log domain). Θεωρούμε ότι $\alpha_{t+1}^*(S) = \ln(\alpha_{t+1}(S))$, $\beta_t^*(S') = \ln(\beta_t(S'))$ και $\gamma_t^*(S', S) = \ln(\gamma_t(S', S))$. Τότε, τα $\alpha_{t+1}^*(S)$ και $\beta_t^*(S')$ υπολογίζονται από τις σχέσεις

$$\alpha_{t+1}^*(S) = \max_{S' \in \sigma_t}^* \{ \gamma_t^*(S', S) + \alpha_t^*(S') \} \quad (2.47)$$

και

$$\beta_t^*(S') = \max_{S \in \sigma_{t+1}}^* \{ \gamma_t(S', S) + \beta_{t+1}^*(S) \} \quad (2.48)$$

όπου

$$\max^* \{x, y\} \equiv \ln(e^x + e^y) = \max\{x, y\} + \ln(1 + e^{-|x-y|}). \quad (2.49)$$

Στη λογαριθμική έκδοση του BCJR, η LLR δίνεται από τον τύπο

$$L^*(u_t^{(i)} | \mathbf{r}) = \max_{(S', S) \in B_t^+}^* \{ \beta_{t+1}^*(S) + \gamma_t^*(S', S) + \alpha_t^*(S') \} - \max_{(S', S) \in B_t^-}^* \{ \beta_{t+1}^*(S) + \gamma_t^*(S', S) + \alpha_t^*(S') \}. \quad (2.50)$$

Η παραπάνω εκδοχή του MAP αλγόριθμου BCJR καλείται Log-MAP. Μια πιο απλοποιημένη προσέγγιση του τελευταίου προκύπτει εάν στην (2.49) παραλείψουμε τον όρο $\ln(1 + e^{-|x-y|})$, οπότε αντί για την συνάρτηση \max^* πραγματοποιείται μόνο μια απλή σύγκριση. Η προσέγγιση αυτή ονομάζεται Max-Log-MAP.

Από τη σύντομη περιγραφή των δύο αλγορίθμων (Viterbi και BCJR) εύκολα γίνεται αντιληπτό ότι ακόμα και η πιο απλουστευμένη εκδοχή του BCJR (Max-Log-MAP) είναι αρκετά πιο πολύπλοκη από τον αλγόριθμο του Viterbi. Η υπολογιστική πολυπλοκότητα που απαιτεί μόνο η προς τα εμπρός αναδρομή του Max-Log-MAP, ο υπολογισμός των μετρικών $\alpha_{t+1}^*(S)$ σε κάθε χρονική στιγμή δηλαδή, είναι περίπου όση η συνολική υπολογιστική πολυπλοκότητα του αλγόριθμου του Viterbi. Στην περίπτωση που τα bit πληροφορίας είναι ισοπίθανα, οι δύο αλγόριθμοι έχουν συγκρίσιμη απόδοση, γι' αυτό και πάντα όταν χρησιμοποιείται ένας συγκεκρι-

ραστικός κώδικας σαν ανεξάρτητο σχήμα κωδικοποίησης, αποκωδικοποιείται με τον αλγόριθμο του Viterbi. Ο αλγόριθμος BCJR έμεινε για αρκετά χρόνια ξεχασμένος εξαιτίας της πολυπλοκότητας του. Η ανακάλυψη των τούρμπο κωδίκων όμως [34] τον έφερε ξανά στο προσκήνιο, καθώς στην επαναληπτική αποκωδικοποίηση η έξοδος του κωδικοποιητή δεν αρκεί να είναι μόνο η εκτιμώμενη τιμή του εκάστοτε bit πληροφορίας (ή κωδικού bit), αλλά χρειάζεται και ένα μέτρο της αξιοπιστίας της εκτίμησης. Η έξοδος του BCJR είναι η LLR του εκάστοτε κωδικού bit, κι όπως φαίνεται και στην (2.39) η απόλυτη τιμή της είναι ένα μέτρο της αξιοπιστίας της εκτίμησης. Ένας αλγόριθμος αποκωδικοποίησης του οποίου και η είσοδος αλλά και η έξοδος αποτελείται από πραγματικούς (μη κβαντισμένους) αριθμούς λέγεται ότι είναι αλγόριθμος *soft input-soft output* (SISO).

Όπως προκύπτει από τις σχέσεις (2.32), (2.33) σε συνδυασμό με την (2.34) για το BSC ή από τις (2.37), (2.38) για το AWGN, τα άνω φράγματα της πιθανότητας σφάλματος κατά την αποκωδικοποίηση $P(E)$ και της πιθανότητας σφάλματος ενός bit πληροφορίας $P_b(E)$ μειώνονται εκθετικά με την ελεύθερη απόσταση d_f , ενώ μειώνονται γραμμικά με το A_{d_f} και το I_{d_f} . Με λίγα λόγια, σχεδιάζοντας ένα συγκεραστικό κώδικα (κυρίως για μεγάλο σηματοθορυβικό λόγο) κύριο μέλημα είναι η μεγιστοποίηση της ελεύθερης απόστασης και η ελαχιστοποίηση των A_{d_f} και I_{d_f} (προφανώς θεωρούμε συγκεκριμένο κωδικοποιητή), και εν συνεχεία η ελαχιστοποίηση των υπόλοιπων όρων των φασμάτων του. Συγκρίνοντας δύο συγκεραστικούς κώδικες (για μέτριες και μεγάλες τιμές του σηματοθορυβικού λόγου) του ίδιου ρυθμού και μνήμης επιλέγουμε αυτόν με τη μεγαλύτερη d_f . Εάν έχουν την ίδια d_f και τα ίδια A_d για $d_f \leq d \leq l - 1$, επιλέγουμε αυτόν με το μικρότερο A_l . Επιπροσθέτως, εάν συγκρίνουμε δύο συγκεκριμένους κωδικοποιητές και ενδιαφερόμαστε για την ελαχιστοποίηση της $P_b(E)$, για την ίδια τιμή της d_f και τα ίδια I_d για $d_f \leq d \leq l - 1$, επιλέγουμε αυτόν με το μικρότερο I_l . Βέβαια, το I_d εξαρτάται σε μεγάλο βαθμό από το N_d .

2.2.4 Κατηγορίες Συγκεραστικών Κωδίκων

Στη βιβλιογραφία παρουσιάζονται καλοί συγκεραστικοί (n, k, m) κώδικες για διάφορες τιμές των n , k και m [52, 53, 54, 55, 56, 57, 58]. Στην περίπτωση των συγκεραστικών κωδίκων, σε αντίθεση με τους γραμμικούς μπλοκ, έχει αποδειχθεί δύσκολη η ανάπτυξη αλγεβρικών κατασκευών που εγγυόνται υψηλή ελεύθερη απόσταση και καλά φάσματα. Έτσι, καθώς δεν υπάρχουν συγκεκριμένα χαρακτηριστικά του πίνακα γεννήτορα που να οδηγούν σε καλούς κώδικες, προκειμένου να σχεδιαστεί ένας (n, k, m) συγκεραστικός κώδικας, πρέπει να εξεταστούν όλοι οι δυνατοί πίνακες γεννήτορες (ή πίνακες ελέγχου ισοτιμίας), να υπολογισθεί η ελεύθερη απόσταση και ενδεχομένως κάποιοι όροι των φασμάτων των αντίστοιχων κωδίκων και να επιλεγεί ο καλύτερος. Αν η αναζήτηση είναι εξαντλητική, τότε ο προκύπτων κώδικας-κωδικοποιητής είναι βέλτιστος. Στην περίπτωση που η εξαντλητική αναζήτηση είναι εξαιρετικά χρονοβόρα (εώς αδύνατη), γίνεται χρήση διαφόρων ευρηστικών σχεδιαστικών κανόνων, οι οποίοι έχουν αυξημένη

πιθανότητα να οδηγήσουν σε κώδικες υψηλής ελεύθερης απόστασης και καλών φασμάτων [59].

Σε πολλές εφαρμογές είναι απαραίτητη η χρήση κωδίκων υψηλού ρυθμού. Εύκολα όμως μπορεί να γίνει αντιληπτό από τη σύντομη αναφορά που κάναμε στο διάγραμμα trellis, ότι όσο αυξάνεται ο ρυθμός, αυξάνεται και το πλήθος των ακμών που αναχωρούν από μία κατάσταση στο trellis module του κώδικα, και συνεπώς αυξάνεται ο αριθμός των ακμών που περιέχονται στο trellis module και γενικότερα στο διάγραμμα trellis. Ως αποτέλεσμα, αυξάνεται η πολυπλοκότητα του αλγόριθμου του Viterbi (η πολυπλοκότητα του αλγόριθμου του Viterbi και η άμεση σχέση της με το διάγραμμα trellis θα συζητηθεί πιο αναλυτικά παρακάτω). Για να αντιμετωπιστεί αυτό το πρόβλημα προτάθηκαν οι *διάτρητοι συγκεραστικοί κώδικες* (punctured convolutional codes-PCC) [10]. Ο PCC είναι *περιοδικός χρονικά μεταβαλλόμενος* συγκεραστικός κώδικας (periodic time varying convolutional code) ο οποίος προκύπτει απαλείφοντας περιοδικά κάποια κωδικά bit από την έξοδο, και συνεπώς από το διάγραμμα trellis, ενός χαμηλόρυθμου $(\tilde{n}, 1, m)$ συγκεραστικού κώδικα, ο οποίος λέγεται και *μητρικός κώδικας* (mother code). Η περίοδος ενός (n, k, m) PCC είναι ίση με k , ενώ η απαλοιφή των κωδικών bit από την έξοδο του μητρικού κώδικα γίνεται με τη χρήση ενός $\tilde{n} \times k$ δυαδικού πίνακα *απαλοιφής* (puncturing matrix) P . Το στοιχείο p_{ij} του P είναι ίσο με 1 εάν το κωδικό bit που παράγεται από το i -οστό πολυώνυμο του γεννήτορα του μητρικού κώδικα μεταδίδεται τη χρονική στιγμή $t = j \bmod k$, και 0 εάν απαλείφεται. Συνεπώς, το άθροισμα των στοιχείων του P ισούται με n . Το διάγραμμα trellis του PCC είναι περιοδικό (έπειτα από ένα μικρό αρχικό διάστημα) χρονικά μεταβαλλόμενο και προκύπτει από το διάγραμμα trellis του μητρικού κώδικα, διαγράφοντας τα κατάλληλα κωδικά bit από τις ακμές του. Η περίοδος του διαγράμματος trellis του PCC είναι ίση με k , ενώ το τμήμα που αντιστοιχεί σε μία περίοδο είναι το trellis module του κώδικα.

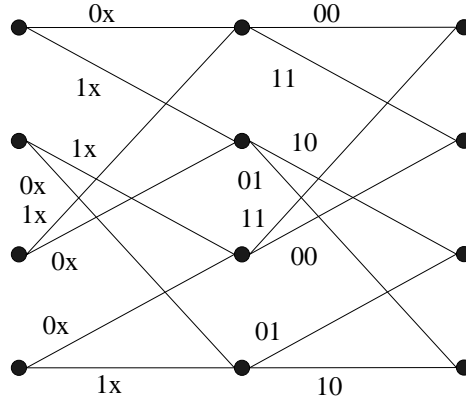
Παράδειγμα 2.17. Εάν χρησιμοποιήσουμε ως μητρικό κώδικα τον κώδικα που παράγεται από τον πίνακα γεννήτορα στην (2.18), και τον πίνακα απαλοιφής

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad (2.51)$$

προκύπτει ένας $(3, 2, 2)$ PCC, του οποίου το trellis module απεικονίζεται στο Σχ. 2.7. Τα διαγραμμένα bit συμβολίζονται με x , ενώ έχουμε παραλείψει τις ετικέτες των τιμών εισόδου των ακμών.

Στη βιβλιογραφία παρουσιάζονται διάφοροι (n, k, m) PCC. Συγκεκριμένα, στα [60, 13, 61, 11] δημοσιεύονται PCC οι οποίοι προκύπτουν από μητρικούς κώδικες των οποίων οι πίνακες γεννήτορες έχουν συγκεκριμένο αριθμό πολυωνύμων, ενώ στο [15] δημοσιεύονται οι καλύτεροι PCC θεωρώντας ότι δεν υπάρχει περιορισμός στο πλήθος των πολυωνύμων.

Οι PCC ανήκουν στην ευρύτερη οικογένεια των κωδίκων που παράγονται από περιοδικούς χρονικά μεταβαλλόμενους κωδικοποιητές με περίοδο T [62, 63, 64]. Ένας τέτοιος κωδικοποιητής



Σχήμα 2.7. Το trellis module του $(3, 2, 2)$ PCC που προκύπτει από τον $\mathbf{G}(D)$ στην (2.18) σε συνδυασμό με τον πίνακα απαλοιφής στην (2.51).

περιγράφεται από τους $\tilde{k} \times n_t$ δυαδικούς πίνακες $\mathbf{G}_j(t)$, $0 \leq j \leq \nu$, έτσι ώστε

$$\mathbf{v}_t = \sum_{j=0}^{\nu} \mathbf{u}_{t-j} \mathbf{G}_j(t), \quad t \geq 0 \quad (2.52)$$

όπου τα \mathbf{u}_t και \mathbf{v}_t συμβολίζουν την ακολουθία εισόδου μήκους \tilde{k} και την ακολουθία εξόδου μήκους n_t αντίστοιχα, τη χρονική στιγμή t . Προφανώς ισχύει ότι $\mathbf{G}_j(t+T) = \mathbf{G}_j(t)$ και $n_{t+T} = n_t$ για όλα τα j και t . Ο περιοδικός κωδικοποιητής στην (2.52) έχει έναν ισοδύναμο χρονικά αμετάβλητο κωδικοποιητή που περιγράφεται από τους $T\tilde{k} \times \sum_{l=0}^{T-1} n_l$ δυαδικούς πίνακες

$$\mathbf{G}_j = \begin{bmatrix} \mathbf{G}_{jT}(0) & \mathbf{G}_{jT+1}(1) & \dots & \mathbf{G}_{jT+T-1}(T-1) \\ \mathbf{G}_{jT-1}(0) & \mathbf{G}_{jT}(1) & \dots & \mathbf{G}_{jT+T-2}(T-1) \\ \vdots & \vdots & & \vdots \\ \mathbf{G}_{jT-T+1}(0) & \mathbf{G}_{jT-T+2}(1) & \dots & \mathbf{G}_{jT}(T-1) \end{bmatrix} \quad (2.53)$$

για $0 \leq j \leq \nu^*$, έτσι ώστε

$$\mathbf{v}_t^* = \sum_{j=0}^{\nu^*} \mathbf{u}_{t-j}^* \mathbf{G}_j$$

όπου $\nu^* = \lceil \frac{\nu}{T} \rceil$ και \mathbf{u}_t^* και \mathbf{v}_t^* είναι η είσοδος και η έξοδος του κωδικοποιητή αντίστοιχα τη χρονική στιγμή t . Οι δύο κωδικοποιητές είναι ισοδύναμοι με την έννοια ότι για την ίδια λέξη πληροφορίας παράγουν την ίδια κωδική ακολουθία. Στην περίπτωση των PCC το \tilde{k} είναι ίσο με 1.

Για τον ισοδύναμο χρονικά αμετάβλητο κωδικοποιητή που παράγει τον PCC στο Παράδειγμα

2.17 έχουμε ότι $\nu^* = 1$,

$$\mathbf{G}_0 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{2.54}$$

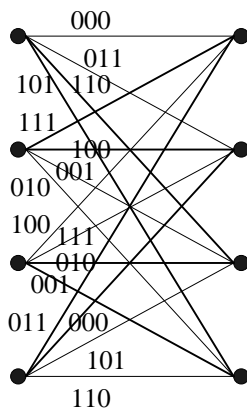
και

$$\mathbf{G}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{2.55}$$

Δηλαδή, ο αντίστοιχος πολυωνυμικός πίνακας γεννήτορας είναι ο

$$\mathbf{G}(D) = \begin{bmatrix} 1 + D & 1 & 0 \\ D & 1 + D & 1 + D \end{bmatrix}. \tag{2.56}$$

Το συμβατικό trellis module που προκύπτει από τον $\mathbf{G}(D)$ φαίνεται στο Σχ. 2.8. Τα διαγράμματα trellis που προκύπτουν από τα trellis modules στα Σχ. (2.7) και 2.8 αναπαριστούν τον ίδιο κώδικα.



Σχήμα 2.8. Το συμβατικό trellis module του $(3, 2, 2)$ PCC στο Παράδειγμα 2.17.

Ένα σημαντικό πλεονέκτημα των PCC είναι η δυνατότητα της κατασκευής σχήματος κωδικοποίησης μεταβλητού ρυθμού, χρησιμοποιώντας έναν $(n, 1, m)$ μπτρικό κώδικα, κι ένα σύνολο πινάκων απαλοιφής που οδηγούν σε διάφορους κωδικούς ρυθμούς [11, 12, 16, 17]. Ανάλογα με τις απαιτήσεις του συστήματος χρησιμοποιείται ο κατάλληλος πίνακας απαλοιφής, κάνοντας έτσι σωστή διαχείριση του διαθέσιμου εύρους ζώνης, ενώ όλοι οι παραγόμενοι κώδικες αποκωδικοποιούνται χρησιμοποιώντας τον αποκωδικοποιητή του μπτρικού κώδικα.

2.3 Το Ελάχιστο Διάγραμμα Trellis για Γραμμικούς Μπλοκ και Συγκραστικούς Κώδικες

Το διάγραμμα trellis \mathcal{T} είναι ένας κατευθυνόμενος γράφος (directed graph) του οποίου οι ακμές φέρουν ετικέτες (edge-labeled), με την ιδιότητα ότι κάθε κόμβος (κατάσταση) στο \mathcal{T}

βρίσκεται σε συγκεκριμένο βάθος (depth) (το βάθος έχει την έννοια της χρονικής στιγμής). Γι' αυτό το λόγο, το σύνολο των καταστάσεων Σ του \mathcal{T} χωρίζεται στα υποσύνολα $\Sigma_0, \Sigma_1, \Sigma_2, \dots$, των οποίων η τομή είναι το κενό σύνολο. Με Σ_i συμβολίζουμε το υποσύνολο των καταστάσεων του \mathcal{T} σε βάθος i . Όπως είναι προφανές, σε αντίστοιχα υποσύνολα B_0, B_1, B_2, \dots και $\Lambda_0, \Lambda_1, \Lambda_2, \dots$ χωρίζονται και το σύνολο των ακμών B και των ετικετών Λ^2 αντίστοιχα. Μία ακμή η οποία ανήκει στο υποσύνολο ακμών B_i του \mathcal{T} συνδέει μία κατάσταση σε βάθος i με μία κατάσταση σε βάθος $i + 1$, και φέρει ετικέτα η οποία ανήκει στο υποσύνολο Λ_i . Η δομή η οποία απαρτίζεται από το υποσύνολο των ακμών B_i , και τα υποσύνολα των καταστάσεων Σ_i και Σ_{i+1} (στις οποίες βρίσκεται η αρχική και η τελική κατάσταση αντίστοιχα μιας ακμής στο B_i) αποτελεί το i -οστό τμήμα trellis (trellis section) του \mathcal{T} . Η αλληλουχία L ακμών στο διάγραμμα trellis καλείται μονοπάτι (path) μήκους L .

Το διάγραμμα trellis πρωτοχρησιμοποιήθηκε στους συγκεραστικούς κώδικες, ως φυσική επέκταση του διαγράμματος του χώρου καταστάσεων των κωδικοποιητών τους στο χρόνο, προκειμένου να περιγραφεί αρτιότερα ο αλγόριθμος του Viterbi [65]. Η trellis αναπαράσταση των γραμμικών μπλοκ κωδίκων προτάθηκε από τους Bahl, Cocke, Jelinek και Raviv (BCJR) στο [23], με σκοπό την αποδοτικότερη αποκωδικοποίησή τους χρησιμοποιώντας αλγόριθμους που μέχρι τότε χρησιμοποιούνταν μόνο στους συγκεραστικούς κώδικες. Οι νέες δυνατότητες που προσέφερε αυτή η τεχνική ώθησε πολλούς ερευνητές στη μελέτη του διαγράμματος trellis για μπλοκ κώδικες και διάφορα χαρακτηριστικά που αφορούν στην κατασκευή, τη δομή και την πολυπλοκότητα του [24, 66, 67, 68, 69].

Όπως αρχικά ορίστηκε, το διάγραμμα trellis ενός (n, k) γραμμικού μπλοκ κώδικα \mathcal{C} αποτελείται από $n + 1$ επίπεδα καταστάσεων (βάθη), δηλαδή από n τμήματα trellis. Τα Σ_0 και Σ_n αποτελούνται από μία μόνο κατάσταση, ενώ η ετικέτα κάθε ακμής παίρνει τιμές στο F_2 (κάθε τμήμα trellis δηλαδή αντιστοιχεί σε ένα κωδικό bit). Η αλληλουχία των ετικετών των ακμών που απαρτίζουν ένα μονοπάτι το οποίο αναχωρεί από τη μοναδική κατάσταση σε βάθος 0 και καταλήγει στην επίσης μοναδική κατάσταση σε βάθος $n + 1$ αποτελεί κωδική λέξη του \mathcal{C} και αντιστρόφως.

Υπάρχουν αρκετές ποσότητες που χαρακτηρίζουν την πολυπλοκότητα του διαγράμματος trellis, όπως η διάσταση s_i του χώρου καταστάσεων σε κάθε βάθος i , η διάσταση i του χώρου ακμών του κάθε τμήματος trellis i , το σύνολο των ακμών, κ.ο.κ. Ένα από τα σημαντικότερα μέτρα της πολυπλοκότητας είναι το σύνολο των κωδικών συμβόλων (bit) που βρίσκονται στις ετικέτες των ακμών του διαγράμματος trellis, καθώς απ' αυτό εξαρτάται σημαντικά η υπολογιστική πολυπλοκότητα του αλγόριθμου του Viterbi. Ο McEliece το αναφέρει στο [24, 25] ως *trellis πολυπλοκότητα* (Trellis complexity-TC). Το διάγραμμα trellis του \mathcal{C} , όπως περιγράφεται παραπάνω, είναι *bit-oriented*, δηλαδή κάθε τμήμα trellis (και προφάνως κάθε ακμή) αντιστοιχεί σ'ένα κωδικό bit. Συνεπώς, σ'αυτήν την περίπτωση η trellis πολυπλοκότητα συμπίπτει με το

²Αναφερόμαστε σε ετικέτες εξόδου. Τις ετικέτες εισόδου προς το παρόν θα τις αγνοήσουμε, αφού δε μας ενδιαφέρει η αναπαράσταση συγκεκριμένου κωδικοποιητή.

πλήθος των ακμών του διαγράμματος trellis. Μία ακόμα ποσότητα από την οποία εξαρτάται η υπολογιστική πολυπλοκότητα του αλγόριθμου του Viterbi είναι ο αριθμός των *συνενώσεων* (merges) που περιέχονται στο διάγραμμα trellis. Ο αριθμός των συνενώσεων σε μία συγκεκριμένη κατάσταση ισούται με τον αριθμό των ακμών που καταλήγουν σε αυτήν μείον ένα. Η trellis πολυπλοκότητα και ο αριθμός των συνενώσεων του διαγράμματος trellis ισούται με τον αριθμό των πραγματικών προσθέσεων και συγκρίσεων που πρέπει να κάνει ο αλγόριθμος του Viterbi προκειμένου να αποκωδικοποιηθούν k bit πληροφορίας [24, 68].

Κάθε (n, k) γραμμικός μπλοκ κώδικας \mathcal{C} μπορεί να περιγραφεί από ένα *ελάχιστο διάγραμμα trellis* (minimal trellis diagram), το οποίο ελαχιστοποιεί ταυτόχρονα όλα τα μέτρα πολυπλοκότητας που αναφέραμε παραπάνω. Το ελάχιστο διάγραμμα είναι μοναδικό, τουλάχιστον σε επίπεδο ισομορφισμού. Όπως αποδείχθηκε μεταγενέστερα [24], το BCJR διάγραμμα trellis για τον \mathcal{C} είναι ελάχιστο.

Γενικότερα, ένα διάγραμμα trellis για τον \mathcal{C} μπορεί να κατασκευαστεί χρησιμοποιώντας έναν πίνακα γεννήτορα \mathbf{G} που τον παράγει, με τη μέθοδο που παρουσιάζεται στο [24]. Έστω \mathbf{g}_i μία γραμμή του \mathbf{G} . Συμβολίζουμε με $L(\mathbf{g}_i)$ και $R(\mathbf{g}_i)$ τις θέσεις του πρώτου και του τελευταίου αντίστοιχα μη μηδενικού στοιχείου της \mathbf{g}_i . Το διάστημα $[L(\mathbf{g}_i), R(\mathbf{g}_i)]$ καλείται *ενεργό διάστημα* (active span) της γραμμής, ενώ τα στοιχεία που περιέχονται σ' αυτό καλούνται επίσης *ενεργά*. Το πλήθος των στοιχείων που περιλαμβάνονται στο ενεργό διάστημα ονομάζεται *μήκος του ενεργού διαστήματος* (span length). Το στοιχείο της \mathbf{g}_i στη θέση $L(\mathbf{g}_i)$ καλείται και υπογραμμισμένο στοιχείο (διότι για πρακτικούς λόγους υπογραμμίζεται), ενώ αντίστοιχα το στοιχείο στη θέση $R(\mathbf{g}_i)$ καλείται και υπεργραμμισμένο στοιχείο. Το πλήθος των ενεργών στοιχείων της στήλης i του \mathbf{G} , το πλήθος δηλαδή των γραμμών των οποίων το ενεργό διάστημα περιλαμβάνει το i ισούται με τη διάσταση του χώρου ακμών b_i , στο τμήμα trellis i . Ο αριθμός των ενεργών στοιχείων της στήλης i χωρίς να συμπεριλάβουμε τα τελευταία μη μηδενικά στοιχεία των γραμμών ισούται με τη διάσταση του χώρου καταστάσεων s_{i+1} σε βάθος $i + 1$.

Ορισμός 2.18. Έστω ένας (n, k) γραμμικός μπλοκ κώδικας \mathcal{C} . Ανάμεσα σε όλους τους πίνακες γεννήτορες που τον παράγουν, αυτοί των οποίων το άθροισμα των μηκών των ενεργών διαστημάτων των γραμμών τους είναι ελάχιστο ονομάζονται *πίνακες γεννήτορες ελαχίστου ενεργού διαστήματος* (minimal span generator matrix-MSGM).

Ένα σύνολο δυαδικών διανυσμάτων $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k$ λέμε ότι έχει την *left-right* ιδιότητα (*LR property*) εάν $L(\mathbf{g}_i) \neq L(\mathbf{g}_j)$ και $R(\mathbf{g}_i) \neq R(\mathbf{g}_j)$ για $i \neq j$.

Θεώρημα 2.19. [67, 24] Ένας πίνακας \mathbf{G} είναι MSGM αν και μόνο αν έχει την ιδιότητα *LR*.

Χρησιμοποιώντας έναν MSGM, το διάγραμμα trellis του \mathcal{C} που κατασκευάζεται με τη μέθοδο στο [24] είναι ελάχιστο, και επί της ουσίας είναι το διάγραμμα trellis BCJR. Από έναν MSGM μπορούμε να υπολογίσουμε τις τιμές όλων των μέτρων πολυπλοκότητας του ελαχίστου διαγράμματος trellis του \mathcal{C} .

Παράδειγμα 2.20. Θεωρείστε τον παρακάτω πίνακα γεννήτορα του $(7, 4)$ γραμμικού μπλοκ

κώδικα Hamming

$$G = \begin{bmatrix} \underline{1} & 1 & 0 & \bar{1} & 0 & 0 & 0 \\ 0 & \underline{1} & 1 & 0 & \bar{1} & 0 & 0 \\ \underline{1} & 1 & 1 & 0 & 0 & \bar{1} & 0 \\ \underline{1} & 0 & 1 & 0 & 0 & 0 & \bar{1} \end{bmatrix}. \quad (2.57)$$

Ο τελευταίος δεν αποτελεί MSGM, καθώς $L(1) = L(3) = L(4) = 1$. Εάν τον χρησιμοποιήσουμε προκειμένου να κατασκευάσουμε ένα διάγραμμα trellis για τον κώδικα, τότε η διάσταση του χώρου καταστάσεων σε κάθε βάθος του εν λόγω διαγράμματος trellis θα είναι $[s_0, s_1, s_2, s_3, s_4, s_5, s_6] = [0, 3, 4, 4, 3, 2, 1, 0]$, ενώ η διάσταση του χώρου ακμών θα είναι $[b_0, b_1, b_2, b_3, b_4, b_5, b_6] = [3, 4, 4, 4, 3, 2, 1, 0]$. Η trellis πολυπλοκότητα είναι ίση με $TC = 78$, και υπολογίζεται με βάση τη σχέση

$$TC = \sum_{i=1}^n b_i \quad (2.58)$$

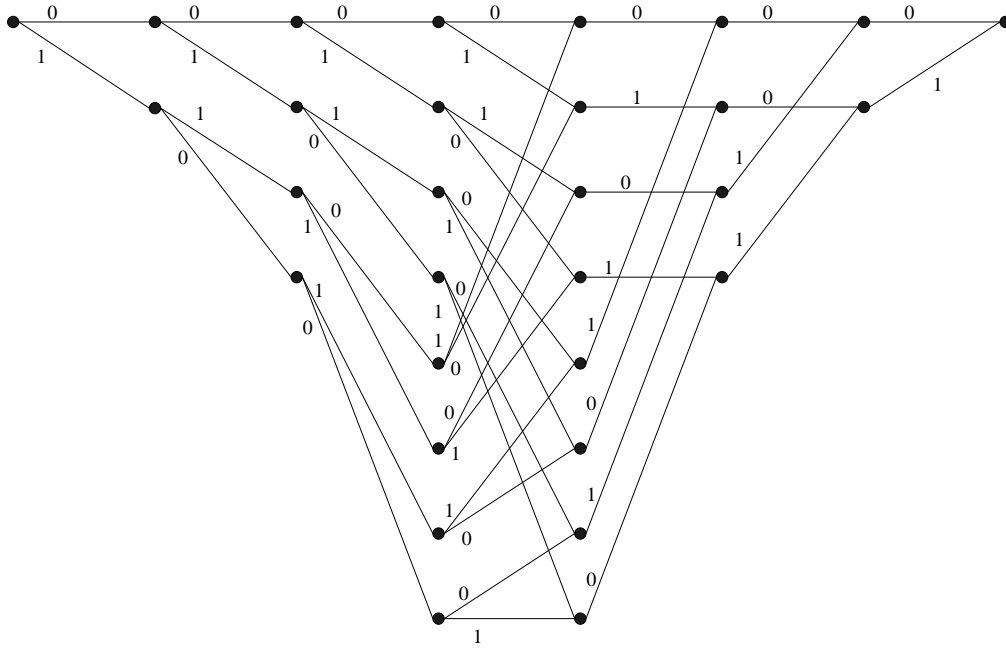
καθώς στην προκειμένη περίπτωση κάθε ακμή αντιστοιχεί σ' ένα κωδικό bit.

Πραγματοποιώντας γραμμοπράξεις, ο \mathbf{G} μετασχηματίζεται στον MSGM

$$\mathbf{G}_{ms} = \begin{bmatrix} \underline{1} & 1 & 0 & \bar{1} & 0 & 0 & 0 \\ 0 & \underline{1} & 1 & 0 & \bar{1} & 0 & 0 \\ 0 & 0 & \underline{1} & 1 & 0 & \bar{1} & 0 \\ 0 & 0 & 0 & \underline{1} & 1 & 0 & \bar{1} \end{bmatrix}. \quad (2.59)$$

Χρησιμοποιώντας τον G_{ms} κατασκευάζεται το ελάχιστο διάγραμμα trellis του κώδικα (Σχ. 2.9), για το οποίο ισχύει ότι $[s_0, s_1, s_2, s_3, s_4, s_5, s_6] = [0, 1, 2, 3, 3, 2, 1, 0]$, $[b_0, b_1, b_2, b_3, b_4, b_5, b_6] = [1, 2, 3, 4, 3, 2, 1]$ και $TC = 44$. Κάθε μονοπάτι το οποίο αναχωρεί από τη μοναδική κατάσταση σε βάθος 0 και καταλήγει στην επίσης μοναδική κατάσταση σε βάθος 6 αντιστοιχεί σε μία κωδική λέξη.

Αξίζει να σημειωθεί εδώ, πως μία μετάθεση των στηλών ενός πίνακα γεννήτορα \mathbf{G} , ο οποίος παράγει έναν γραμμικό μπλοκ κώδικα \mathcal{C} , οδηγεί σ' έναν πίνακα \mathbf{G}' ο οποίος παράγει έναν ισοδύναμο κώδικα \mathcal{C}' . Ο τελευταίος έχει ακριβώς τα ίδια φάσματα με τον \mathcal{C} , και επί της ουσίας κάθε λέξη του προκύπτει από τη μετάθεση των bit μιας λέξης του \mathcal{C} . Παρόλ' αυτά, τα ελάχιστα διαγράμματα trellis των δύο κωδίκων μπορεί να είναι τελείως διαφορετικά μεταξύ τους. Καλούμε βέλτιστο το ελάχιστο διάγραμμα trellis που ελαχιστοποιεί ένα συγκεκριμένο μέτρο πολυπλοκότητας (π.χ. την trellis πολυπλοκότητα), ως προς όλα τα ελάχιστα διαγράμματα trellis που αντιστοιχούν στους ισοδύναμους γραμμικούς μπλοκ κώδικες οι οποίοι προκύπτουν από όλες τις δυνατές μεταθέσεις του \mathbf{G} . Γενικότερα δεν υπάρχει εγγύηση για την ύπαρξη μιας μετάθεσης η οποία ελαχιστοποιεί όλες τις ποσότητες που σχετίζονται με την πολυπλοκότητα του διαγράμματος trellis. Η εύρεση της βέλτιστης μετάθεσης για κάποιο μέτρο πολυπλοκότητας πιστεύεται ότι είναι πρόβλημα *NP-hard*, παρόλο που δεν έχει αποδειχτεί ακόμα [68].



Σχήμα 2.9. Ελάχιστο διάγραμμα trellis του $(7, 4)$ γραμμικού μπλοκ κώδικα Hamming στο Παράδειγμα 2.20.

Η μελέτη του ελάχιστου διαγράμματος trellis των γραμμικών μπλοκ κωδίκων έδωσε το έναυσμα για τη διερεύνηση του ενδεχομένου της περιγραφής ενός συγκεραστικού κώδικα από λιγότερο πολύπλοκες δομές, εν συγκρίσει με το συμβατικό διάγραμμα trellis [25, 26]. Στο [26] κατασκευάζεται το ελάχιστο διάγραμμα trellis ενός συγκεραστικού κώδικα με τη χρήση του πίνακα ελέγχου ισοτιμίας, ενώ στο [25] (στο οποίο μελετάται ενδελεχώς η δομή και η πολυπλοκότητα του ελάχιστου διαγράμματος trellis) χρησιμοποιείται ο πίνακας γεννήτορας.

Θεωρήστε έναν κανονικό πίνακα γεννήτορα $\mathbf{G}(D)$ με μνήμη m , ο οποίος παράγει τον (n, k, m) συγκεραστικό κώδικα \mathcal{C} . Έστω επίσης ο αντίστοιχος ημι-άπειρος βαθμιδωτός πίνακας \mathbf{G}_{scalar} , ο οποίος περιγράφεται στην (2.15). Όπως είδαμε, ο τελευταίος θεωρείται ημι-άπειρος με την έννοια ότι το μήκος της ακολουθία πληροφορίας μπορεί να είναι οσοδήποτε μεγάλο. Έστω ότι κωδικοποιείται μία δυαδική ακολουθία πληροφορίας μήκους kL . Τότε, ο \mathbf{G}_{scalar} απαρτίζεται από kL γραμμές και $(L + \nu)n$ στήλες (γι' αυτό και θα τον συμβολίζουμε με $\mathbf{G}_{scalar}^{[L]}$), και αποτελεί τον πίνακα γεννήτορα ενός $((L + \nu)n, kL)$ γραμμικού μπλοκ κώδικα $\mathcal{C}^{[L]}$. Ο τελευταίος αποτελείται από όλα τα κωδικά μονοπάτια του \mathcal{C} τα οποία καταλήγουν στην κατάσταση μηδέν

σε βάθος $L + \nu$. Ο $\mathbf{G}_{scalar}^{[L]}$ είναι της μορφής

$$\mathbf{G}_{scalar}^{[L]} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu & \\ & & & & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_\nu \end{bmatrix}. \quad (2.60)$$

Σύμφωνα με αυτά που συζητήθηκαν στην αρχή της υποενότητας, ο $\mathbf{G}_{scalar}^{[L]}$ μπορεί να χρησιμοποιηθεί προκειμένου να κατασκευάσουμε ένα διάγραμμα trellis για τον γραμμικού μπλοκ κώδικα $\mathcal{C}^{[L]}$. Εάν η $\mathbf{G}_{scalar}^{[L]}$ είναι MSGM, εάν έχει την LR ιδιότητα δηλαδή, τότε το διάγραμμα trellis που θα προκύψει θα είναι ελάχιστο. Όπως φαίνεται από την (2.60), ο $\mathbf{G}_{scalar}^{[L]}$ (όπως άλλωστε και ο \mathbf{G}_{scalar}) έχει πολύ χαρακτηριστική δομή. Συγκεκριμένα κάθε k γραμμές του αποτελούν μια οριζόντια μετατόπιση του πίνακα

$$\tilde{\mathbf{G}} = [\mathbf{G}_0 \mathbf{G}_1 \dots \mathbf{G}_\nu]. \quad (2.61)$$

Αντίστοιχα, κάθε n στήλες του $\mathbf{G}_{scalar}^{[L]}$ αποτελούν μία κατακόρυφη μετατόπιση του πίνακα (matrix module)

$$\hat{\mathbf{G}} = \begin{bmatrix} \mathbf{G}_\nu \\ \vdots \\ \mathbf{G}_1 \\ \mathbf{G}_0 \end{bmatrix}. \quad (2.62)$$

Είναι προφανές ότι εάν ισχύει $L(i) \neq L(j) \bmod n$ και $R(i) \neq R(j) \bmod n$ για οποιοσδήποτε δύο γραμμές i και j του πίνακα $\tilde{\mathbf{G}}$ (ή ισοδύναμα εάν περιέχεται το πολύ ένα υπογραμμισμένο και το πολύ ένα υπεργραμμισμένο στοιχείο σε κάθε στήλη του $\hat{\mathbf{G}}$), τότε ο $\mathbf{G}_{scalar}^{[L]}$ έχει την LR -ιδιότητα και συνεπώς παράγει το (bit-oriented) ελάχιστο διάγραμμα trellis του $\mathcal{C}^{[L]}$. Το τελευταίο είναι περιοδικό με περίοδο n . Η δομή που αντιστοιχεί σε μία περίοδο του ελάχιστου διαγράμματος trellis είναι το ελάχιστο trellis module (minimum trellis module) \mathcal{M}_{min} . Η trellis πολυπλοκότητα TC του ελάχιστου trellis module δίνεται από τη σχέση

$$TC = \sum_{i=1}^n b_i \quad (2.63)$$

όπου b_i είναι το πλήθος των ενεργών στοιχείων που βρίσκονται στη στήλη i του πίνακα $\hat{\mathbf{G}}$. Η TC είναι το πλήθος των συμβόλων των ακμών που περιέχονται στο trellis module (πολλές φορές εκφράζεται ανά bit πληροφορίας).

Το ελάχιστο trellis module θα αποτελέσει το “κύτταρο” του ελαχίστου διαγράμματος trellis

για οποιαδήποτε τιμή του L . Συνήθως θεωρούμε το διάγραμμα trellis ενός συγκεραστικού κώδικα \mathcal{C} ημι-άπειρο, υπό την έννοια ότι ξεκινάει σε ένα συγκεκριμένο γνωστό χρονικό σημείο (όπου ξεκινάει και η κωδικοποίηση), ενώ το τέλος του μπορεί να βρίσκεται σε οποιοδήποτε βάθος, προκειμένου να μπορεί να κωδικοποιηθεί μια ακολουθία πληροφορίας οσοδήποτε μεγάλου μήκους. Επίσης, για τους ίδιους λόγους, αντί για τον $\mathbf{G}_{scalar}^{[L]}$ θα αναφερόμαστε στον ημι-άπειρο \mathbf{G}_{scalar} . Το ελάχιστο (ημι-άπειρο) διάγραμμα trellis του \mathcal{C} αποτελείται (έπειτα από ένα μικρό αρχικό τμήμα) από επαναλαμβανόμενα αντίτυπα του ελάχιστου trellis module.

Ορισμός 2.21. Ένας κανονικός πίνακας γεννήτορας $\mathbf{G}(D)$ του οποίου ο αντίστοιχος \mathbf{G}_{scalar} έχει την LR -ιδιότητα λέγεται *trellis κανονικός* (trellis canonical) [25, 70].

Χρησιμοποιώντας έναν trellis κανονικό πίνακα για έναν συγκεραστικό κώδικα \mathcal{C} μπορούμε να κατασκευάσουμε το ελάχιστο trellis module και να υπολογίσουμε την ελάχιστη trellis πολυπλοκότητα, καθώς και την ελάχιστη τιμή οποιουδήποτε άλλου μέτρου της πολυπλοκότητας. Επίσης, κάθε κανονικός πίνακας γεννήτορας μπορεί να μετασχηματιστεί σε trellis κανονικό πίνακα με βασικές γραμμοπράξεις [25].

Ένας (n, k, m) συγκεραστικός κώδικας \mathcal{C} μπορεί να περιγραφεί από διάφορα διαγράμματα trellis, και συνεπώς από διάφορα trellis modules. Όπως έχουμε αναφέρει και παραπάνω, το trellis module είναι η δομή του διαγράμματος trellis η οποία αντιστοιχεί σε μία περίοδο, και κατά τη διάρκεια της οποίας k bit πληροφορίας αντιστοιχίζονται σε n κωδικά bit. Για παράδειγμα, το πιο γνωστό trellis module είναι το συμβατικό που περιγράψαμε παραπάνω, το οποίο προκύπτει κατευθείαν από το διάγραμμα καταστάσεων του κωδικοποιητή. Η trellis πολυπλοκότητά του είναι ίση με $n2^{m+k}$.

Παράδειγμα 2.22. Θεωρήστε τον $(3, 2, 2)$ συγκεραστικό κώδικα που παράγεται από τον κανονικό πίνακα γεννήτορα

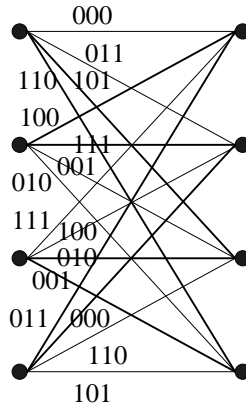
$$G(D) = \begin{bmatrix} 1 + D & D & 1 + D \\ D & 1 & 1 \end{bmatrix}. \quad (2.64)$$

Το συμβατικό trellis module του \mathcal{C} απεικονίζεται στο Σχ. 2.10. Η trellis πολυπλοκότητα είναι ίση με 48. Επίσης, στο Σχ. (2.11) φαίνεται το (συμβατικό) trellis module που προκύπτει από τον ισοδύναμο πίνακα γεννήτορα

$$G(D) = \begin{bmatrix} 1 + D & D & 1 + D \\ D^2 & D & D \end{bmatrix} \quad (2.65)$$

ο οποίος δεν είναι κανονικός. Η trellis πολυπλοκότητα είναι ίση με 96. Τέλος, το ελάχιστο trellis module του \mathcal{C} απεικονίζεται στο Σχ. 2.12, κι η αντίστοιχη trellis πολυπλοκότητα είναι ίση με 24.

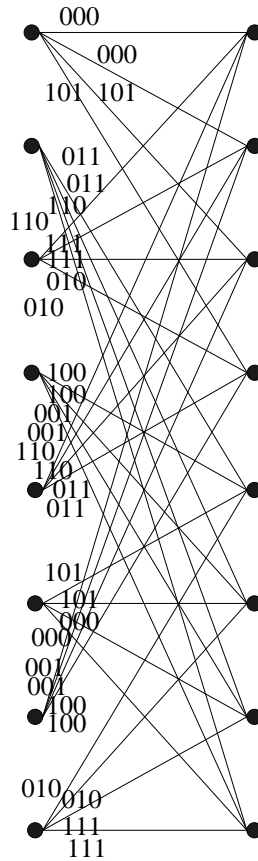
Εύκολα μπορεί να διαπιστωθεί ότι ο χρονικά αμετάβλητος πίνακας γεννήτορας του PCC στο Παράδειγμα (2.17) είναι σε trellis κανονική μορφή, συνεπώς μπορούμε να υπολογίσουμε την ελάχιστη trellis πολυπλοκότητα που μπορεί να έχει ένα trellis module που τον περιγράφει. Η



Σχήμα 2.10. Συμβατικό trellis module του συγκεραστικού κώδικα στο Παράδειγμα 2.22.

ελάχιστη TC είναι ίση με 24. Η trellis πολυπλοκότητα του trellis module ενός PCC δίνεται από τη σχέση $TC = n2^{m+1}$ [25], και στην προκειμένη περίπτωση είναι επίσης ίση με 24 (Σχ. 2.7). Δηλαδή, το trellis module στο Σχ. 2.7 αποτελεί ελάχιστο trellis module του κώδικα, παρόλο που δεν είναι bit oriented.

Θεωρείστε έναν (n,k,m) συγκεραστικό κώδικα \mathcal{C} και ένα trellis module \mathcal{M} που τον αναπαριστά, το οποίο θα το χρησιμοποιήσουμε προκειμένου να τον αποκωδικοποιήσουμε. Έστω ότι το \mathcal{M} αποτελείται από $\hat{n} \leq n$ τμήματα trellis. Εάν ο αλγόριθμος αποκωδικοποίησης είναι ο Viterbi, τότε θεωρούμε ότι n είσοδος του αποκωδικοποιητή είναι τα $\log p(r_{j,i}|v_{j,i})$ για κάθε ακμή, όπου $v_{j,i}$ είναι η τιμή του αντίστοιχου κωδικού bit i που μεταδίδεται κατά το τμήμα trellis j και $r_{j,i}$ το αντίστοιχο ληφθέν σύμβολο. Ο αριθμός των προσθέσεων και των συγκρίσεων που πρέπει να γίνουν για να αποκωδικοποιηθούν k bit πληροφορίας είναι ίσος με την trellis πολυπλοκότητα TC και τον αριθμό των συνενώσεων E του \mathcal{M} αντίστοιχα. Στην περίπτωση του Max-Log-MAP αλγόριθμου θα θεωρήσουμε ότι ο κωδικοποιητής είναι συστηματικός, καθώς ο εν λόγω αλγόριθμος χρησιμοποιείται σχεδόν αποκλειστικά στην επαναληπτική αποκωδικοποίηση των τούρμπο κωδικών, οι οποίοι χρησιμοποιούν συστηματικούς συγκεραστικούς κωδικοποιητές. Βέβαια, η υπολογιστική πολυπλοκότητα του αλγόριθμου δεν αλλάζει σε διαφορετική περίπτωση. Θεωρούμε ότι η είσοδος του αποκωδικοποιητή είναι τα $\ln(P(v_{j,i})p(r_{j,i}|v_{j,i}))$ (ή $v_{j,i}L_a(v_{j,i})/2 + \ln(p(r_{j,i}|v_{j,i}))$) εάν το $v_{j,i}$ είναι κωδικό bit και $\ln(p(r_{j,i}|v_{j,i}))$ εάν είναι bit ισοτιμίας. Ο υπολογισμός των μετρικών ακμής γ απαιτεί $\sum_{j=0}^{\hat{n}-1} (n_j - 1)2^{b_j}$ προσθέσεις πραγματικών αριθμών ανά k bit πληροφορίας, όπου n_j είναι το πλήθος των bit που αντιστοιχούν στο τμήμα trellis j , και b_j η διάσταση του χώρου ακμών. Στη συνέχεια, ο υπολογισμός των προς τα εμπρός μετρικών απαιτεί $\sum_{j=0}^{\hat{n}-1} 2^{b_j}$ προσθέσεις και E συγκρίσεις ανά k bit πληροφορίας, ενώ οι ίδιες πράξεις χρειάζονται και για τον υπολογισμό των προς τα πίσω μετρικών [71, 72]. Τέλος, για τον υπολογισμό των LLR χρειάζονται $\sum_{j=0}^{\hat{n}-1} (b_j - s_j)2^{b_j+1} + k$ προσθέσεις και $\sum_{j=0}^{\hat{n}-1} (b_j - s_j)2^{b_j} - 2k$ συγκρίσεις ανά k bit πληροφορίας. Συνολικά, ο αλγόριθμος Max-Log-MAP πρέπει να εκτελέσει $TC + \sum_{j=0}^{\hat{n}-1} (b_j - s_j + \frac{1}{2})2^{b_j+1} + k$ προσθέσεις και $2E + \sum_{j=0}^{\hat{n}-1} (b_j - s_j)2^{b_j} - 2k$ συγκρίσεις



Σχήμα 2.11. Συμβατικό trellis module του συγκεραστικού κώδικα στο Παράδειγμα 2.22, που προκύπτει από μη κανονικό πίνακα γεννήτορα.

ανά k bit πληροφορίας. Με ανάλογο τρόπο προκύπτει και η υπολογιστική πολυπλοκότητα του αλγόριθμου BCJR και του Log-MAP.

Η ανάλυση των McEliece και Lin στο [25] έδειξε πως ο λόγος για τον οποίο οι PCC μπορούν να περιγραφούν από trellis modules λιγότερο πολύπλοκα από το συμβατικό είναι η ιδιαίτερη δομή του (χρονικά αμετάβλητου) πίνακα γεννήτορα τους. Κάθε συγκεραστικός κώδικας μπορεί να αναπαρασταθεί από ένα ελάχιστο trellis module το οποίο είναι συνήθως λιγότερο πολύπλοκο από το συμβατικό. Έτσι, προτείνεται [25] η αναζήτηση και η κατασκευή συγκεραστικών κωδίκων με κριτήριο την ελάχιστη trellis πολυπλοκότητα, κι όχι το μέγεθος της μνήμης του κανονικού κωδικοποιητή που χρησιμοποιούνταν μέχρι τότε. Επίσης, τίθεται ως “ανοιχτό πρόβλημα” η ύπαρξη άλλων οικογενειών κωδίκων χαμηλής πολυπλοκότητας και καλής απόδοσης, όπως οι PCC. Προς τις παραπάνω κατευθύνσεις κινούνται τα άρθρα [29, 30, 28, 31, 32, 33]. Στα [29, 30] κατασκευάζονται $(n, n - 1)$ συγκεραστικοί κώδικες με κριτήριο την TC του ελαχίστου trellis module. Πρέπει να σημειωθεί εδώ πως κατ’ αντιστοιχία με τους γραμμικούς μπλοκ κώδικες, μία μετάθεση των σπλών του πίνακα γεννήτορα ενός συγκεραστικού κώδικα οδηγεί σε ισοδύναμο κώδικα. Βέβαια, τα ελάχιστα trellis modules των δύο κωδίκων μπορεί να διαφέρουν

Στον παραπάνω πίνακα τα g_0^i είναι μη μηδενικά μόνο για k τιμές του i . Όπως φαίνεται και στην (2.66), εάν $g_0^p = 0$, τότε το g_0^{p+1} βρίσκεται τοποθετημένο στην ίδια γραμμή του $\hat{\mathbf{G}}$ κι όχι στην επόμενη. Το GPCC trellis module \mathcal{M}_{GPCC} αποτελείται από n τμήματα trellis, ενώ η διάσταση του χώρου καταστάσεων σε βάθος i ισούται με \hat{s}_i . Το \mathcal{M}_{GPCC} δεν είναι απαραίτητα ελάχιστο trellis module.

Πολλοί κώδικες που δημοσιεύονται στα [29, 30] ανήκουν στην κατηγορία των GPCC. Ένα μειονέκτημα των κατασκευών στα [29, 30, 31, 32, 33] είναι ότι πρόκειται για αρκετά ευρείες οικογένειες με αποτέλεσμα όσο αυξάνεται η ελάχιστη trellis πολυπλοκότητα να αυξάνεται αρκετά και ο χώρος αναζήτησης κωδίκων.

3. ΚΑΤΑΣΚΕΥΕΣ ΣΥΓΚΕΡΑΣΤΙΚΩΝ ΚΩΔΙΚΩΝ ΧΑΜΗΛΗΣ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ

Σ' αυτό το κεφάλαιο θα παρουσιάσουμε μία οικογένεια συγκεραστικών κωδίκων χαμηλής πολυπλοκότητας [75, 76], οι οποίοι προκύπτουν τροποποιώντας το trellis module των PCC. Στόχος μας είναι να αυξήσουμε την απόδοση, με κόστος μια λογική αύξηση της trellis πολυπλοκότητας. Η αύξηση της πολυπλοκότητας επιτυγχάνεται μεταβάλλοντας τη διάσταση του χώρου καταστάσεων, σε διάφορα βάρη του trellis module. Επίσης, αναλύεται το προφίλ πολυπλοκότητας του ελάχιστου trellis module των εν λόγω κωδίκων, και προτείνεται μία αποδοτική τεχνική για την κατασκευή του. Στην περίπτωση των $(n, n - 1)$ κωδίκων αποδεικνύονται ικανές και αναγκαίες συνθήκες που πρέπει να ισχύουν προκειμένου το ελάχιστο trellis module να είναι βέλτιστο. Τέλος, παρουσιάζονται αρκετοί “καλοί” συγκεραστικοί κώδικες χαμηλής πολυπλοκότητας, ενώ αναπτύσσεται κι ένα άνω φράγμα για την ελάχιστη απόστασή τους.

3.1 Νέες Κατηγορίες Χρονικά Μεταβαλλόμενων Κωδίκων

Θεωρούμε έναν περιοδικό χρονικά μεταβαλλόμενο γραμμικό κωδικοποιητή με περίοδο k [64, 77, 62, 78, 63, 79, 80, 81], ο οποίος περιγράφεται από τους $1 \times n_t$ δυαδικούς πίνακες $\mathbf{G}_j(t)$, $0 \leq j \leq m_t$, έτσι ώστε

$$\mathbf{v}_t = \sum_{j=0}^{m_t} u_{t-j} \mathbf{G}_j(t), \quad t \geq 0 \quad (3.1)$$

όπου τα u_t και v_t συμβολίζουν την ακολουθία εισόδου μήκους 1 και την ακολουθία εξόδου μήκους n_t αντίστοιχα, τη χρονική στιγμή t . Επίσης, με m_t συμβολίζουμε τη μνήμη του κωδικοποιητή τη χρονική στιγμή t . Εφόσον ο κωδικοποιητής είναι περιοδικός, ισχύει ότι $\mathbf{G}_j(t+k) = \mathbf{G}_j(t)$, $m_{t+k} = m_t$, και $n_{t+k} = n_t$ για όλα τα j και t .

Το διάγραμμα trellis που αναπαριστά την παραπάνω κατασκευή είναι περιοδικό (μετά από μια σύντομη χρονική περίοδο), και το αντίστοιχο trellis module \mathcal{M} αποτελείται από k τμήματα. Κατά τη διάρκεια ενός trellis module, k bit πληροφορίας αντιστοιχίζονται σε n κωδικά bit, όπου $n = \sum_{l=0}^{k-1} n_l$. Η διάσταση του χώρου καταστάσεων του trellis module σε βάθος l είναι ίση με m_l , $0 \leq l \leq k-1$ ($m_k = m_0$). Το διάστημα του trellis module ανάμεσα στα βάρη l και $l+1$ ονομάζεται l -οστό τμήμα trellis (ή l -οστό τμήμα του trellis module). Με $v_{l,i}$ συμβολίζουμε το i -οστό κωδικό bit, το οποίο μεταδίδεται κατά τη διάρκεια του τμήματος l του trellis module, για $1 \leq i \leq n_l$, και για $0 \leq l \leq k-1$, και παράγεται από την ακολουθία γεννήτορα $\mathbf{g}^{(l,i)} = [g_0^{(l,i)} \ g_1^{(l,i)} \ \dots \ g_{m_l}^{(l,i)}]$.

Δοθέντος ενός θετικού ακεραίου m , επιβάλλουμε τους εξής περιορισμούς: $m_l \in \{m, m+1\}$, και $m_0 = m$. Συμβολίζουμε με T_{ζ_1, ζ_2} το σύνολο των τμημάτων l , $0 \leq l \leq k-1$, του trellis module, κατά τη διάρκεια των οποίων η διάσταση του χώρου καταστάσεων αλλάζει από ζ_1 σε ζ_2 , όπου $\zeta_1, \zeta_2 \in \{m, m+1\}$. Δηλαδή, $T_{\zeta_1, \zeta_2} = \{l | m_l = \zeta_1, \ m_{l+1} = \zeta_2, \ 0 \leq l \leq k-1\}$. Επίσης,

περιορίζουμε τη δομή του κώδικα έτσι ώστε οι πίνακες $\mathbf{G}_j(l) = [g_j^{(l,1)} \ g_j^{(l,2)} \ \dots \ g_j^{(l,n_l)}]$ να έχουν τις παρακάτω ιδιότητες:

$$P_1. \mathbf{G}_0(l) = \mathbf{1}, 0 \leq l \leq k - 1,$$

$$P_2. \mathbf{G}_m(l) = \mathbf{1}, l \in T_{m,m},$$

$$P_3. \mathbf{G}_{m+1}(l) = \mathbf{1}, l \in T_{m+1,m+1},$$

$P_4. g_m^{(l,i)} \neq 0$ ή $g_{m+1}^{(l,i)} \neq 0$, για όλα τα $1 \leq i \leq n_l$, ενώ οι $\mathbf{G}_m(l)$ και $\mathbf{G}_{m+1}(l)$ είναι γραμμικά ανεξάρτητοι, όταν το $l \in T_{m+1,m}$.

Από την ιδιότητα P_4 προκύπτει ότι $n_l \geq 2$, όταν $l \in T_{m+1,m}$. Συμβολίζουμε με p_l το μικρότερο ακέραιο, $1 \leq p_l < n_l$, τέτοιον ώστε

$$g_m^{(l,p_l+1)} = \dots = g_m^{(l,n_l)} \quad \text{και} \quad g_{m+1}^{(l,p_l+1)} = \dots = g_{m+1}^{(l,n_l)}$$

όταν $l \in T_{m+1,m}$.

Από την περιγραφή του trellis module \mathcal{M} προκύπτει άμεσα ότι η trellis πολυπλοκότητα TC , το πλήθος δηλαδή των συμβόλων στις ακμές του, δίνεται από τη σχέση

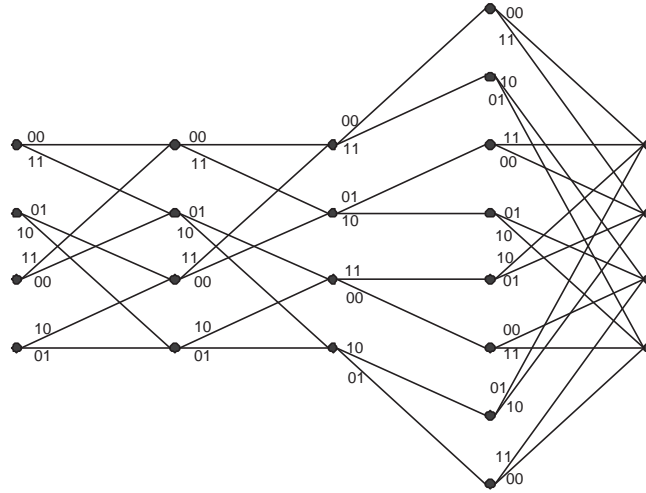
$$TC(\mathcal{M}) = \sum_{l \in T_{m,m}} n_l 2^{m+1} + \sum_{l \in T_{m,m+1}} n_l 2^{m+1} + \sum_{l \in T_{m+1,m+1}} n_l 2^{m+2} + \sum_{l \in T_{m+1,m}} n_l 2^{m+2}. \quad (3.2)$$

Στη συνέχεια του κεφαλαίου θα δούμε ότι το \mathcal{M} δεν είναι ελάχιστο. Θα υπολογίσουμε την ελάχιστη πολυπλοκότητα των νέων κωδίκων και θα αναπτύξουμε μια μέθοδο κατασκευής του ελαχίστου trellis module.

Θεωρούμε ως αφετηρία χρονικά μεταβαλλόμενες δομές με σταθερό προφίλ διαστάσεων του χώρου καταστάσεων, όπως οι (n, k, m) PCC. Κατασκευάζουμε νέους κώδικες επιτρέποντας μεταβολές στη διάσταση του χώρου καταστάσεων του κάθε τμήματος του trellis module. Είναι εύκολο να επιβεβαιωθεί ότι για $m_l = m$, $0 \leq l \leq k - 1$, το trellis module \mathcal{M} αντιστοιχεί σε έναν (n, k, m) PCC. Στόχος μας είναι να βρούμε κώδικες καλύτερης απόδοσης, αυξάνοντας την trellis πολυπλοκότητα. Θέλοντας να διατηρήσουμε την πολυπλοκότητα σε χαμηλά επίπεδα, περιορίζουμε τη μέγιστη τιμή της m_l να είναι ίση με $m + 1$. Είναι προφανές πως υπάρχουν πολλοί τρόποι να κατανεμηθούν τα n κωδικά bit στα k τμήματα του trellis module. Εδώ θα χρησιμοποιήσουμε την κατανομή που προέκυψε από τη μελέτη στο [15], αφού αυτή οδήγησε σε βέλτιστους χρονικά μεταβαλλόμενους κώδικες σταθερής μνήμης.

Παράδειγμα 3.1. Θεωρείστε έναν κώδικα που ανήκει στην προτεινόμενη κατηγορία, και παράγεται από έναν περιοδικό κωδικοποιητή με περίοδο 4. Υποθέστε επίσης ότι $n_l = 2$ για $0 \leq l \leq 3$, $m_l = 2$, για $0 \leq l \leq 2$, και $m_3 = 3$. Το αντίστοιχο trellis module απεικονίζεται στο Σχήμα 3.1. Αποτελείται από 4 τμήματα, εκ των οποίων τα πρώτα δύο ανήκουν στο σύνολο $T_{2,2}$. Τα δύο τελευταία ανήκουν στα σύνολα $T_{2,3}$ και $T_{3,2}$ αντίστοιχα. Κατά τη διάρκεια του trellis module, 4 bit πληροφορίας αντιστοιχίζονται σε 8 κωδικά bit, οπότε ο ρυθμός του κώδικα

είναι $4/8$. Οι γεννήτορες πίνακες είναι οι $\mathbf{G}_0(l) = [1 \ 1]$, $\mathbf{G}_1(l) = [0 \ 1]$, $\mathbf{G}_2(l) = [1 \ 1]$, για $0 \leq l \leq 2$, $\mathbf{G}_0(3) = \mathbf{G}_2(3) = [1 \ 1]$, $\mathbf{G}_1(3) = \mathbf{G}_3(3) = [1 \ 0]$, και ικανοποιούν τις ιδιότητες $P_1 - P_4$. Επίσης, $p_3 = 1$. Τέλος, από τη σχέση (3.2) προκύπτει ότι το trellis module έχει trellis πολυπλοκότητα $TC = 80$.



Σχήμα 3.1. Trellis module του $(8, 4)$ συγκεραστικού κώδικα στο παράδειγμα 3.1.

3.2 Ο Ισοδύναμος Χρονικά Αμετάβλητος Κωδικοποιητής

Ένας συγκεραστικός κώδικας \mathcal{C} , όπως παρουσιάστηκε στην ενότητα 3.1, παράγεται από έναν χρονικά μεταβαλλόμενο γραμμικό κωδικοποιητή με περίοδο k και μεταβλητή μνήμη m_l , ο οποίος περιγράφεται από τους $1 \times n_l$ πίνακες $\mathbf{G}_j(l)$, για $0 \leq j \leq m_l$ και $0 \leq l \leq k - 1$.

Λήμμα 3.2. *Ο περιοδικός χρονικά μεταβαλλόμενος κωδικοποιητής που παράγει τον κώδικα \mathcal{C} ικανοποιεί τις παρακάτω ιδιότητες.*

1. $\mathbf{G}_{m+1}(0) = \mathbf{0}$,
2. $\mathbf{G}_m(k - 1) \neq \mathbf{0}$,
3. $\mathbf{G}_m(l) \neq \mathbf{0}$ ή $\mathbf{G}_{m+1}(l + 1) \neq \mathbf{0}$, $0 \leq l \leq k - 2$.

Απόδειξη.

1. Εξ' ορισμού, $m_0 = m$. Άρα $\mathbf{G}_{m+1}(0) = \mathbf{0}$.
2. Εξ' ορισμού, $m_k = m_0 = m$. Επομένως, το τελευταίο τμήμα του trellis module ανήκει είτε στο σύνολο $T_{m,m}$ ή στο σύνολο $T_{m+1,m}$. Από τις ιδιότητες, P_2 και P_4 , έχουμε ότι $\mathbf{G}_m(k - 1) \neq \mathbf{0}$.

3. Από την ιδιότητα P_2 , όταν $m_l = m_{l+1} = m$, τότε $\mathbf{G}_m(l) \neq \mathbf{0}$. Όταν $m_l = m + 1$ και $m_{l+1} = m$, από την ιδιότητα P_4 , ισχύει ότι $\mathbf{G}_m(l) \neq \mathbf{0}$. Τέλος, όταν $m_{l+1} = m + 1$, από τις ιδιότητες P_3 και P_4 έχουμε ότι $\mathbf{G}_{m+1}(l + 1) \neq \mathbf{0}$.

□

Συμπληρώνοντας με μηδενικά τους πίνακες $\mathbf{G}_j(l)$, όπου αυτό είναι απαραίτητο, μπορούμε να υποθέσουμε ότι ο προτεινόμενος περιοδικός κωδικοποιητής που παράγει τον κώδικα \mathcal{C} , είναι ένας χρονικά μεταβαλλόμενος συγκεραστικός κωδικοποιητής με σταθερή μνήμη $m + 1$. Έτσι, ο ισοδύναμος χρονικά αμετάβλητος κωδικοποιητής περιγράφεται από τους $k \times n$ πίνακες \mathbf{G}_j , για $0 \leq j \leq \nu_f$, όπου $\nu_f = \lceil \frac{m+1}{k} \rceil$, η μορφή των οποίων δίνεται στη σχέση (2.53) [62]. Θα χρησιμοποιήσουμε την ειδική δομή των πινάκων $\mathbf{G}_j(l)$ για να υπολογίσουμε τη συνολική μνήμη του χρονικά αμετάβλητου κωδικοποιητή.

Θεώρημα 3.3. *Θεωρήστε τον περιοδικό χρονικά μεταβαλλόμενο κωδικοποιητή με περίοδο k που παράγει τον κώδικα \mathcal{C} . Ο ισοδύναμος χρονικά αμετάβλητος κωδικοποιητής έχει συνολική μνήμη $m_f = m$.*

Απόδειξη. Έστω ότι $\hat{m} = m + 1 \pmod{k}$. Από τη σχέση (2.53) και την υπόθεση ότι ο περιοδικός κωδικοποιητής έχει σταθερή μνήμη ίση με $m + 1$, έχουμε ότι

$$\mathbf{G}_{\nu_f} = \begin{bmatrix} \mathbf{G}_{m+1+\rho k-\hat{m}}(0) & \mathbf{G}_{m+1+\rho k-\hat{m}+1}(1) & \cdots & \mathbf{G}_{m+1+\rho k-\hat{m}+k-1}(k-1) \\ \mathbf{G}_{m+1+\rho k-\hat{m}-1}(0) & \mathbf{G}_{m+1+\rho k-\hat{m}}(1) & \cdots & \mathbf{G}_{m+1+\rho k-\hat{m}+k-2}(k-1) \\ \vdots & \vdots & & \vdots \\ \mathbf{G}_{m+1+\rho k-\hat{m}-k+1}(0) & \mathbf{G}_{m+1+\rho k-\hat{m}-k+2}(1) & \cdots & \mathbf{G}_{m+1+\rho k-\hat{m}}(k-1) \end{bmatrix} \quad (3.3)$$

και

$$\mathbf{G}_{\nu_f-1} = \begin{bmatrix} \mathbf{G}_{m+1+\rho k-\hat{m}-k}(0) & \mathbf{G}_{m+1+\rho k-\hat{m}+1-k}(1) & \cdots & \mathbf{G}_{m+1+\rho k-\hat{m}-1}(k-1) \\ \mathbf{G}_{m+1+\rho k-\hat{m}-1-k}(0) & \mathbf{G}_{m+1+\rho k-\hat{m}-k}(1) & \cdots & \mathbf{G}_{m+1+\rho k-\hat{m}-2}(k-1) \\ \vdots & \vdots & & \vdots \\ \mathbf{G}_{m+1+\rho k-\hat{m}-k+1-k}(0) & \mathbf{G}_{m+1+\rho k-\hat{m}-2k+2}(1) & \cdots & \mathbf{G}_{m+1+\rho k-\hat{m}-k}(k-1) \end{bmatrix}. \quad (3.4)$$

όπου

$$k \cdot \nu_f = k \cdot \left\lceil \frac{m+1}{k} \right\rceil = m + 1 + \rho k - \hat{m}$$

και $\rho = 0$, αν $\hat{m} = 0$, και $\rho = 1$, διαφορετικά. Εφόσον $\mathbf{G}_j(l) = \mathbf{0}$, για $j > m + 1$ και $\mathbf{G}_{m+1}(0) = \mathbf{0}$, μπορεί να επιβεβαιωθεί ότι οι πρώτες $R_0 = \rho k - \hat{m} + 1$ γραμμές του πίνακα \mathbf{G}_{ν_f} είναι μηδενικές. Έτσι, η συνολική μνήμη m_f του χρονικά αμετάβλητου κωδικοποιητή φράσσεται άνω από

$$\begin{aligned} m_f &\leq R_0(\nu_f - 1) + (k - R_0)\nu_f \\ &= k\nu_f - \rho k + \hat{m} - 1 = m. \end{aligned} \quad (3.5)$$

Η ισότητα ισχύει όταν οι πρώτες R_0 γραμμές του πίνακα \mathbf{G}_{ν_f-1} και οι τελευταίες $k - R_0$ γραμμές του \mathbf{G}_{ν_f} είναι μη μηδενικές. Εύκολα μπορεί να ελεγχθεί ότι ο $\mathbf{G}_m(r - \rho k + \hat{m} + k - 1)$ και ο $\mathbf{G}_{m+1}(r - \rho k + \hat{m} + k)$ εμφανίζονται στην r -οστή γραμμή του πίνακα \mathbf{G}_{ν_f-1} , για $0 \leq r < R_0 - 1$, και από το Λήμμα 3.2 τουλάχιστον ένας απ'αυτούς είναι μη μηδενικός. Επίσης, η $(R_0 - 1)$ -οστή γραμμή του \mathbf{G}_{ν_f-1} περιέχει τον $\mathbf{G}_m(k - 1)$ ο οποίος σύμφωνα με το Λήμμα 3.2 είναι μη μηδενικός. Τέλος, ο $\mathbf{G}_m(r - \rho k + \hat{m} - 1)$ και ο $\mathbf{G}_{m+1}(r - \rho k + \hat{m})$, βρίσκονται στην r -οστή γραμμή του πίνακα \mathbf{G}_{ν_f} για $R_0 \leq r \leq k - 1$ και σύμφωνα με το Λήμμα 3.2 τουλάχιστον ένας απ'αυτούς είναι μη μηδενικός. Έτσι, η συνολική μνήμη m_f του χρονικά αμετάβλητου κωδικοποιητή είναι ίση με m . \square

Θεώρημα 3.4. *Εάν ο ισοδύναμος χρονικά αμετάβλητος πίνακας γεννήτορας \mathbf{G} δεν είναι καταστροφικός, τότε είναι κανονικός.*

Απόδειξη. Ο πολυωνυμικός πίνακας γεννήτορας του χρονικά αμετάβλητου κωδικοποιητή είναι ο

$$\mathbf{G}(D) = \mathbf{G}_0 + \mathbf{G}_1 \cdot D + \dots + \mathbf{G}_{\nu_f} \cdot D^{\nu_f}.$$

Από το Θεώρημα 2.11, ένας πολυωνυμικός πίνακας γεννήτορας είναι κανονικός αν και μόνο αν είναι βασικός (basic), και ο αντίστοιχος πίνακας \mathbf{G}_h που περιέχει τους συντελεστές του όρου ν_j ($\nu_j \leq \nu_f$) των πολυωνύμων κάθε γραμμής j είναι βαθμού (rank) k . Πρώτα θα δείξουμε ότι ο \mathbf{G} είναι βασικός. Αυτό ισχύει αν και μόνο αν ο μέγιστος κοινός διαιρέτης $X(D)$ των οριζουσών όλων των $k \times k$ υποπινάκων του είναι ίσος με 1 (Θεώρημα 2.9). Εφόσον ο \mathbf{G} δεν είναι καταστροφικός, τότε το $X(D)$ είναι ίσο με D^l , για $l \geq 0$ (Θεώρημα 2.14). Από τη σχέση (2.53) και την ιδιότητα P_1 , ο πίνακας \mathbf{G}_0 δίνεται από τη σχέση

$$\mathbf{G}_0 = \begin{bmatrix} \mathbf{1}_{n_0} & \mathbf{G}_1(1) & \dots & \mathbf{G}_{k-1}(k-1) \\ \mathbf{0} & \mathbf{1}_{n_1} & \dots & \mathbf{G}_{k-2}(k-1) \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{1}_{n_{k-1}} \end{bmatrix} \quad (3.6)$$

όπου με $\mathbf{1}_x$ συμβολίζουμε το διάνυσμα μήκους x , του οποίου όλες οι συνιστώσες είναι ίσες με 1. Έτσι, μπορούμε να βρούμε έναν $k \times k$ υποπίνακα του \mathbf{G}_0 της μορφής

$$\mathbf{Q}_0 = \begin{bmatrix} 1 & * & \dots & * \\ 0 & 1 & \dots & * \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix}.$$

Με αστερίσκο συμβολίζουμε ένα στοιχείο που μπορεί να έχει είτε την τιμή 0, είτε την τιμή 1. Ο $k \times k$ (υπο)πίνακας \mathbf{Q}_0 είναι τριγωνικός άνω, κι η ορίζουσα του είναι ίση με 1. Έστω ότι ο

\mathbf{Q} είναι ο αντίστοιχος $k \times k$ υποπίνακας του \mathbf{G} . Η ορίζουσα του \mathbf{Q} είναι ένα πολυώνυμο του οποίου ο σταθερός όρος είναι η ορίζουσα του \mathbf{Q}_0 , η οποία είναι ίση με τη μονάδα, κι εφόσον είναι πολλαπλάσιο του $X(D) = D^l$, τότε $l = 0$ και ο \mathbf{G} είναι βασικός.

Στη συνέχεια θα δείξουμε ότι ο δυαδικός πίνακας \mathbf{G}_h είναι βαθμού (rank) k . Στην περίπτωση μας, ο \mathbf{G}_h αποτελείται από τις πρώτες R_0 γραμμές του πίνακα \mathbf{G}_{ν_f-1} και τις τελευταίες $k - R_0$ γραμμές του \mathbf{G}_{ν_f} . Αλλάζοντας τη σειρά των γραμμών του \mathbf{G}_h μπορούμε να παράγουμε τον πίνακα \mathbf{G}_{end} .

$$\mathbf{G}_{end} = \begin{bmatrix} \mathbf{G}_m(0) & \mathbf{G}_{m+1}(1) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{G}_{m-1}(0) & \mathbf{G}_m(1) & \mathbf{G}_{m+1}(2) & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{m-k+2}(0) & \mathbf{G}_{m-k+3}(1) & \mathbf{G}_{m-k+4}(2) & \dots & \mathbf{G}_{m+1}(k-1) \\ \mathbf{G}_{m-k+1}(0) & \mathbf{G}_{m-k+2}(1) & \mathbf{G}_{m-k+3}(2) & \dots & \mathbf{G}_m(k-1) \end{bmatrix}. \quad (3.7)$$

Ισχύει $\text{rank}(\mathbf{G}_h) = \text{rank}(\mathbf{G}_{end})$. Θα δείξουμε ότι ο \mathbf{G}_{end} μπορεί να μετασχηματιστεί σε μορφή κλιμακωτής διάταξης (row echelon form) με k μη μηδενικές γραμμές, δηλαδή το τελευταίο μη μηδενικό στοιχείο κάθε σειράς βρίσκεται (αυστηρά) στα αριστερά του τελευταίου μη μηδενικού στοιχείου της επόμενης γραμμής. Από το Λήμμα 3.2 προκύπτει ότι το τελευταίο μη μηδενικό στοιχείο της γραμμής r βρίσκεται είτε στον $\mathbf{G}_m(r)$, ή στον $\mathbf{G}_{m+1}(r+1)$, για $0 \leq r < k-1$, ενώ για $r = k-1$, το τελευταίο μη μηδενικό στοιχείο βρίσκεται στον $\mathbf{G}_m(k-1)$.

Υποθέστε ότι οι πρώτες $r+1$ γραμμές του πίνακα (από τη γραμμή r και πάνω) είναι σε μορφή κλιμακωτής διάταξης. Επίσης, υποθέστε ότι το τελευταίο μη μηδενικό στοιχείο της γραμμής r δεν βρίσκεται αριστερά του τελευταίου μη μηδενικού στοιχείου της γραμμής $r+1$. Αυτό είναι πιθανό μόνο αν, $\mathbf{G}_{m+1}(r+1) \neq \mathbf{0}$, $\mathbf{G}_m(r+1) \neq \mathbf{0}$ και $\mathbf{G}_{m+1}(r+2) = \mathbf{0}$. Δηλαδή, το τμήμα $r+1$ του trellis module ανήκει στο σύνολο $T_{m+1,m}$. Σ'αυτήν την περίπτωση, από την ιδιότητα P_4 , έχουμε ότι $\mathbf{G}_{m+1}(r+1) \neq \mathbf{G}_m(r+1)$, ενώ και τα δύο είναι μη μηδενικά. Διακρίνουμε δύο περιπτώσεις.

Εάν $g_{m+1}^{(r+1, p_{r+1}+j)} = g_m^{(r+1, p_{r+1}+j)} = 1$, για $1 \leq j \leq n_{r+1} - p_{r+1}$, τότε τα τελευταία μη μηδενικά στοιχεία και των δύο γραμμών βρίσκονται στην ίδια στήλη. Προσθέτοντας τη γραμμή $(r+1)$ στη γραμμή r , το τελευταίο μη μηδενικό στοιχείο της γραμμής r βρίσκεται αριστερά του τελευταίου μη μηδενικού στοιχείου της γραμμής $r+1$. Εάν $g_{m+1}^{(r+1, p_{r+1}+j)} = 1$ και $g_m^{(r+1, p_{r+1}+j)} = 0$, για $1 \leq j \leq n_{r+1} - p_{r+1}$, τότε το τελευταίο μη μηδενικό στοιχείο της γραμμής r βρίσκεται δεξιά του αντίστοιχου στοιχείου της γραμμής $(r+1)$. Γι' αυτό, πρέπει ν' αλλάξουμε τη θέση μεταξύ των γραμμών r και $(r+1)$. Σε οποιαδήποτε άλλη περίπτωση, οι πρώτες $r+2$ γραμμές είναι σε μορφή κλιμακωτής διάταξης. Έτσι, ο \mathbf{G}_{end} μπορεί εύκολα να μετασχηματιστεί σε μορφή κλιμακωτής διάταξης με k μη μηδενικές γραμμές, άρα $\text{rank}(\mathbf{G}_{end}) = k$. Έχουμε δείξει ότι ο \mathbf{G} είναι βασικός και ότι ο αντίστοιχος \mathbf{G}_h είναι βαθμού k , άρα η απόδειξη είναι ολοκληρωμένη. \square

Όπως είδαμε, οι κώδικες της προτεινόμενης κατηγορίας μπορούν επίσης να παραχθούν από

(χρονικά αμετάβλητους) κανονικούς πίνακες γεννήτορες συνολικής μνήμης m , γι' αυτό θα τους αποκαλούμε (n, k, m) συγκεραστικούς κώδικες.

3.3 Η Ελάχιστη Trellis Πολυπλοκότητα και Το Αντίστοιχο Ελάχιστο Trellis Module

Οι προτεινόμενοι κώδικες κατασκευάζονται τροποποιώντας το trellis module των PCC. Παρακάτω θα δείξουμε ότι το trellis module που προκύπτει δεν είναι ελάχιστο. Ένα ελάχιστο trellis module έχει την ελάχιστη δυνατή trellis πολυπλοκότητα. Όπως είδαμε στην ενότητα 2.3, το ελάχιστο trellis module ενός συγκεραστικού κώδικα μπορεί να παραχθεί απευθείας από έναν trellis-κανονικό πίνακα γεννήτορα [25]. Επίσης, κάθε κανονικός πίνακας γεννήτορας μπορεί να μετασχηματιστεί σε trellis-κανονικό πίνακα, εκτελώντας βασικές γραμμοπράξεις. Σ' αυτή την ενότητα θα υπολογίσουμε την ελάχιστη trellis πολυπλοκότητα των νέων κωδίκων και θα παρουσιάσουμε μία αποδοτική διαδικασία για την κατασκευή του ελάχιστου trellis module απευθείας από το αρχικό τους trellis module, αποφεύγοντας τη μέθοδο που παρουσιάστηκε στα [25], [24].

Από τον πίνακα γεννήτορα \mathbf{G} κατασκευάζουμε τον πίνακα

$$\tilde{\mathbf{G}} = [\mathbf{G}_0 \cdots \mathbf{G}_{\nu_f-1} \mathbf{G}_{\nu_f}]. \quad (3.8)$$

Όλα τα στοιχεία που βρίσκονται στο ενεργό διάστημα μιας γραμμής του πίνακα $\tilde{\mathbf{G}}$, δηλαδή ανάμεσα στο πρώτο και το τελευταίο μη μηδενικό στοιχείο (συμπεριλαμβανομένων και αυτών), καλούνται ενεργά στοιχεία (το πρώτο και το τελευταίο ενεργό στοιχείο αναφέρονται και ως “υπογραμμισμένο” και “υπεργραμμισμένο” στοιχείο αντίστοιχα). Ο βαθμιδωτός πίνακας γεννήτορας (ο οποίος αποτελείται από μετατοπισμένα αντίγραφα του $\tilde{\mathbf{G}}$)

$$\mathbf{G}_{scalar} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_{\nu_f} & & \\ & \mathbf{G}_0 & \cdots & \mathbf{G}_{\nu_f-1} & \ddots & \\ & & \ddots & \vdots & & \\ & & & \mathbf{G}_0 & & \\ & & & & & \ddots \end{bmatrix} \quad (3.9)$$

έχει την LR ιδιότητα αν δεν υπάρχουν δύο γραμμές των οποίων το ενεργό διάστημα να ξεκινάει (L -ιδιότητα) ή να τελειώνει (R -ιδιότητα) στην ίδια στήλη. Ισοδύναμα, ο πίνακας \mathbf{G}_{scalar} έχει την LR ιδιότητα εάν έχει το πολύ ένα υπογραμμισμένο και ένα υπεργραμμισμένο στοιχείο σε κάθε στήλη. Όταν ο \mathbf{G} είναι κανονικός και ο \mathbf{G}_{scalar} έχει την LR ιδιότητα, τότε ο \mathbf{G} είναι trellis-κανονικός πίνακας γεννήτορας. Αυτό σημαίνει ότι από τον \mathbf{G} μπορούμε να κατασκευάσουμε το ελάχιστο trellis module για το συγκεκριμένο κώδικα, ενώ η ελάχιστη trellis πολυπλοκότητα

(ανά bit πληροφορίας) είναι ίση με

$$TC = \sum_{i=1}^n 2^{b_i} \quad (3.10)$$

όπου b_i είναι το πλήθος των ενεργών στοιχείων που περιέχονται στη στήλη i του πίνακα $\hat{\mathbf{G}}$ που προκύπτει από τον \mathbf{G}

$$\hat{\mathbf{G}} = \begin{bmatrix} \mathbf{G}_{\nu_f} \\ \mathbf{G}_{\nu_f-1} \\ \vdots \\ \mathbf{G}_0 \end{bmatrix}. \quad (3.11)$$

Λήμμα 3.5. *Εάν ο πίνακας γεννήτορας \mathbf{G} είναι μη καταστροφικός, τότε μπορεί να μετασχηματιστεί σε έναν trellis-κανονικό πίνακα, τέτοιον ώστε,*

- για κάθε τιμή $l \notin T_{m+1,m}$ του trellis module, όλοι οι πίνακες $\mathbf{G}_j(l)$, για $0 \leq j \leq m_l$ περιέχουν μόνο ενεργά στοιχεία. Οι πίνακες $\mathbf{G}_j(l)$, για $j > m_l$ δεν περιέχουν ενεργά στοιχεία.
- για κάθε τιμή $l \in T_{m+1,m}$ του trellis module, οι πίνακες $\mathbf{G}_j(l)$, για $0 \leq j \leq m-1$, περιέχουν μόνο ενεργά στοιχεία. Επίσης, τα πρώτα p_l στοιχεία ενός εκ των πινάκων $\mathbf{G}_m(l)$ και $\mathbf{G}_{m+1}(l)$ και όλα τα n_l στοιχεία του άλλου πίνακα είναι ενεργά. Οι πίνακες $\mathbf{G}_j(l)$, για $j > m+1$ δεν περιέχουν ενεργά στοιχεία.

Θεώρημα 3.6. *Η trellis πολυπλοκότητα του ελάχιστου trellis module \mathcal{M}_{min} του κώδικα \mathcal{C} δίνεται από τη σχέση*

$$TC(\mathcal{M}_{min}) = \sum_{l \in T_{m,m}} n_l 2^{m+1} + \sum_{l \in T_{m,m+1}} n_l 2^{m+1} + \sum_{l \in T_{m+1,m+1}} n_l 2^{m+2} + \sum_{l \in T_{m+1,m}} (n_l + p_l) 2^{m+1}. \quad (3.12)$$

Απόδειξη. Όταν ο πίνακας γεννήτορας \mathbf{G} είναι σε trellis κανονική μορφή, τότε μπορούμε να υπολογίσουμε την ελάχιστη πολυπλοκότητα από τον πίνακα $\hat{\mathbf{G}}$, χρησιμοποιώντας τη σχέση (3.10). Θεωρείστε τις n_l στήλες του $\hat{\mathbf{G}}$, οι οποίες αντιστοιχούν στο τμήμα l του (αρχικού) trellis module, για $0 \leq l \leq k-1$

$$\begin{bmatrix} \mathbf{G}_{m_l}(l) \\ \mathbf{G}_{m_l-1}(l) \\ \vdots \\ \mathbf{G}_0(l) \end{bmatrix}. \quad (3.13)$$

Στη σχέση (3.13), έχουμε παραλείψει τις γραμμές πάνω από τον πίνακα $\mathbf{G}_{m_l}(l)$ και κάτω από τον $\mathbf{G}_0(l)$ οι οποίοι δεν περιέχουν ενεργά στοιχεία.

Εάν το $l \in T_{m+1,m}$ τότε, από το Λήμμα 3.5, η στήλη i του πίνακα στη σχέση (3.13), για $1 \leq i \leq p_l$ περιέχει $m+2$ ενεργά στοιχεία, ενώ για $p_l+1 \leq i \leq n_l$ περιέχει $m+1$ ενεργά στοιχεία. Οπότε,

$$\sum_{i=1}^{n_l} 2^{b_i} = \sum_{i=1}^{p_l} 2^{m+2} + \sum_{i=p_l+1}^{n_l} 2^{m+1} = (n_l + p_l)2^{m+1}. \quad (3.14)$$

Εάν το l ανήκει σε οποιοδήποτε άλλο σύνολο, από το Λήμμα 3.5, η στήλη i του πίνακα στη σχέση (3.13), για $1 \leq i \leq n_l$ περιέχει m_l+1 ενεργά στοιχεία. Οπότε,

$$\sum_{i=1}^{n_l} 2^{b_i} = n_l 2^{m_l+1}. \quad (3.15)$$

Από τις σχέσεις (3.14) και (3.15), αθροίζοντας για όλες τις τιμές του l , η σχέση (3.12) προκύπτει εύκολα. \square

Από τη σύγκριση των σχέσεων (3.12) και (3.2) προκύπτει ότι το αρχικό trellis module δεν έχει την ελάχιστη δυνατή trellis πολυπλοκότητα. Στη συνέχεια θα δείξουμε πως αν αντικαταστήσουμε κάθε τμήμα $l \in T_{m+1,m}$ του αρχικού trellis module με ένα ισοδύναμο ζεύγος τμημάτων, το αποτέλεσμα είναι ένα trellis module ελάχιστης trellis πολυπλοκότητας.

Θεώρημα 3.7. Το τμήμα $l \in T_{m+1,m}$ του trellis module είναι ισοδύναμο με ένα ζεύγος τμημάτων, όπου το πρώτο τμήμα l_1 αντιστοιχεί στα πρώτα p_l κωδικά bit και σε ένα bit πληροφορίας. Επίσης, έχει 2^{m+1} αρχικές καταστάσεις και 2^{m+1} τελικές καταστάσεις. Το δεύτερο τμήμα l_2 αποτελείται από 2^{m+1} αρχικές καταστάσεις και 2^m τελικές καταστάσεις, παράγει τα τελευταία $n_l - p_l$ κωδικά bit του αρχικού τμήματος l , ενώ δεν αντιστοιχεί σε κανένα bit πληροφορίας.

Απόδειξη. Κάθε τμήμα $l \in T_{m+1,m}$ αντιστοιχεί σε ένα bit πληροφορίας και παράγει n_l κωδικά bit. Επίσης, εκ κατασκευής κάθε αρχική κατάσταση $\mathbf{S} = (S_1, S_2, \dots, S_{m-1}, S_m, S_{m+1})$ συνδέεται με την τελική κατάσταση $\mathbf{S}' = (u, S_1, S_2, \dots, S_{m-1})$, όπου u είναι το bit πληροφορίας. Ορίζουμε τα l_1 και l_2 όπως παρακάτω. Κάθε αρχική κατάσταση \mathbf{S} του τμήματος l_1 συνδέεται με την κατάσταση $\mathbf{S}^{int} = (u, S_1, S_2, \dots, S_{m-1}, h(S_m, S_{m+1}))$ του δεύτερου τμήματος l_2 . Επίσης, η (αρχική) κατάσταση \mathbf{S}^{int} (του τμήματος l_2) συνδέεται με την κατάσταση $\mathbf{S}' = (u, S_1, S_2, \dots, S_{m-1})$. Η συνάρτηση h είναι γραμμική και είναι της μορφής

$$h(x_1, x_2) = h_1 \cdot x_1 + h_2 \cdot x_2$$

όπου $h_1 = g_m^{(l, p_l+1)} = \dots = g_m^{(l, n_l)}$ και $h_2 = g_{m+1}^{(l, p_l+1)} = \dots = g_{m+1}^{(l, n_l)}$. Το πρώτο τμήμα l_1 αντιστοιχεί σε ένα bit πληροφορίας και παράγει p_l κωδικά bit. Χρησιμοποιώντας τους ίδιους γεννήτορες με το αρχικό τμήμα l , μπορούμε να παράγουμε τα ίδια κωδικά bit, για την ίδια αρχική κατάσταση και το ίδιο bit πληροφορίας.

Το δεύτερο τμήμα l_2 φέρει τα τελευταία $n_l - p_l$ κωδικά bit, ενώ δεν αντιστοιχεί σε κανένα bit πληροφορίας. Έστω $v_{l_2, i}$ το i κωδικό bit του δεύτερου τμήματος l_2 , για $1 \leq i \leq n_l - p_l$, το

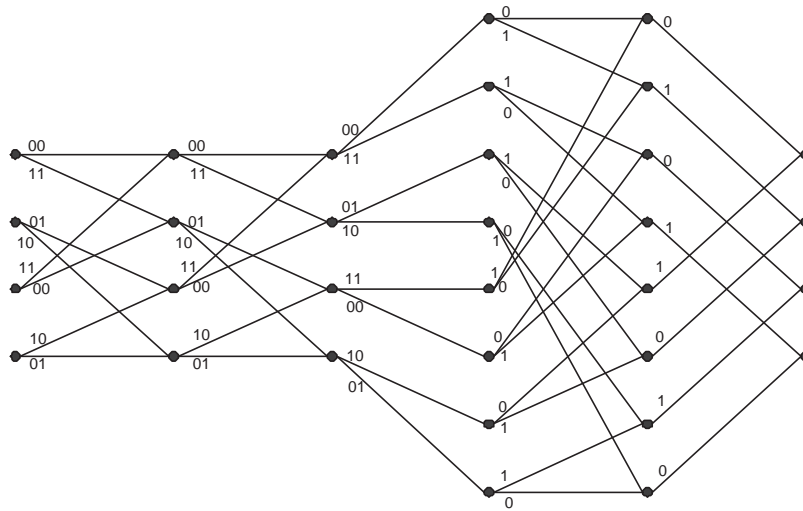
οποίο παράγεται από τη σχέση

$$\begin{aligned} v_{l_2,i} &= \left[g_0^{(l_2,i)} g_1^{(l_2,i)} \dots g_m^{(l_2,i)} \right] \cdot (\mathbf{S}^{int})^T \\ &= g_0^{(l_2,i)} u + \sum_{j=1}^{m-1} g_j^{(l_2,i)} S_j + g_m^{(l_2,i)} h(S_m, S_{m+1}). \end{aligned}$$

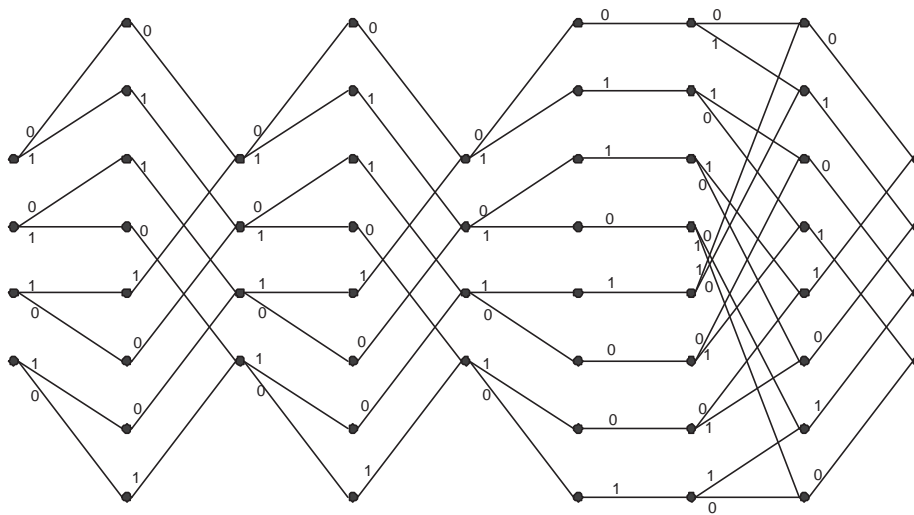
Θέτοντας $g_j^{(l_2,i)} = g_j^{(l,i+p_l)}$, για $0 \leq j \leq m-1$, και $g_m^{(l_2,i)} = 1$, μπορεί εύκολα να επαληθευθεί ότι $v_{l_2,i} = v_{l,i+p_l}$, για $1 \leq i \leq n_l - p_l$. Έτσι, το αρχικό τμήμα $l \in T_{m+1,m}$ είναι ισοδύναμο με το προτεινόμενο ζεύγος τμημάτων. \square

Το πρώτο και το δεύτερο τμήμα του ισοδύναμου ζεύγους περιέχουν $p_l 2^{m+2}$ και $(n_l - p_l) 2^{m+1}$ κωδικά bit στις ακμές τους αντίστοιχα (το δεύτερο τμήμα δεν αντιστοιχεί σε bit πληροφορίας, οπότε μία μόνο ακμή ξεκινάει από κάθε αρχική κατάσταση). Έτσι, το σύνολο των κωδικών bit στις ακμές και των δύο τμημάτων του ισοδύναμου ζεύγους είναι $(n_l + p_l) 2^{m+1}$. Εάν στη σχέση (3.2) αντικαταστήσουμε τον όρο $\sum_{l \in T_{m+1,m}} n_l 2^{m+2}$ με τον όρο $\sum_{l \in T_{m+1,m}} (n_l + p_l) 2^{m+1}$, τότε προκύπτει η σχέση (3.12). Δηλαδή, αν απλά αντικαταστήσουμε στο αρχικό trellis module ενός κώδικα \mathcal{C} κάθε ένα από τα τμήματα $l \in T_{m+1,m}$ με το ισοδύναμό του ζεύγος τμημάτων, το αποτέλεσμα είναι ένα ελάχιστο trellis module για τον \mathcal{C} .

Παράδειγμα 1.1 (συνέχεια) Από τη σχέση (3.12) προκύπτει πως η ελάχιστη trellis πολυπλοκότητα του $(8, 4)$ συγκεραστικού κώδικα που κατασκευάστηκε στην ενότητα 3.1 είναι ίση με $TC = 72$. Το ελάχιστο trellis module του κώδικα (Σχ. 3.2) προκύπτει εύκολα αν αντικαταστήσουμε στο αρχικό trellis module (Σχ. 3.1) το τελευταίο τμήμα (το οποίο ανήκει στο σύνολο $T_{3,2}$) με το ισοδύναμό του ζεύγος τμημάτων. Το ελάχιστο trellis module που κατασκευάζεται σύμφωνα με την τεχνική του McEliece [24, 25] φαίνεται στο Σχήμα 3.3. Κάθε ένα από τα 8 κωδικά bit αντιστοιχεί σε ένα τμήμα του trellis module. Στο trellis module του Σχ. 3.2, κάποια τμήματα φέρουν περισσότερα από ένα κωδικά bit, και γι' αυτό το λόγο λέμε ότι είναι *τμηματοποιημένο* (sectionalized). Αυτό το φαινόμενο συμβαίνει στους περισσότερους (προτεινόμενους) κώδικες με $k \neq n - 1$. Εφόσον με την προαναφερθείσα τεχνική, αλλάζουμε μόνο τα τμήματα $l \in T_{m+1,m}$ προκειμένου να κατασκευάσουμε ένα ελάχιστο trellis module, όλα τα άλλα τμήματα διατηρούν τον αρχικό τους αριθμό κωδικοποιημένων bit. Επίσης, αφού κάθε τμήμα $l \in T_{m+1,m}$ αντικαθίσταται από δύο νέα τμήματα, τότε αν $n_l > 2$, τουλάχιστον ένα από τα νέα τμήματα θα φέρει περισσότερα από ένα κωδικά bit. Αντίθετα, μπορεί εύκολα να επιβεβαιωθεί ότι για $k = n - 1$, το ελάχιστο trellis module που προκύπτει έχει ένα τμήμα για κάθε κωδικό bit (είναι bit-oriented). Στη συνέχεια του κεφαλαίου θα δείξουμε ότι όταν το ελάχιστο trellis module (όπως προκύπτει σ' αυτήν την εργασία) είναι τμηματοποιημένο, τότε ο αριθμός πράξεων που εκτελεί ο αλγόριθμος του Viterbi μπορεί να μειωθεί.



Σχήμα 3.2. Ελάχιστο trellis module του $(8, 4)$ συγκεραστικού κώδικα στο Παράδειγμα 3.1.



Σχήμα 3.3. Bit-oriented ελάχιστο trellis module του $(8, 4)$ συγκεραστικού κώδικα στο Παράδειγμα 3.1.

3.4 Αποτελέσματα Αναζήτησης Κωδίκων και Συγκρίσεις

Κατά τη διάρκεια της διαδικασίας αναζήτησης χρησιμοποιείται ως σημείο αφετηρίας το trellis module ενός (n, k, m) PCC, και αυξάνεται η διάσταση του χώρου καταστάσεων κάποιων τμημάτων του από m σε $m+1$. Στη συνέχεια αναζητούνται οι γεννιότερες $\mathbf{g}^{(l,i)}$ που θα παράγουν τον καλύτερο δυνατό κώδικα. Στόχος είναι η κατασκευή καλύτερων κωδίκων με κόστος μικρή αύξηση της πολυπλοκότητας. Σημειώνεται ότι η πολυπλοκότητα διατηρείται σε λογικά επίπεδα, αφού παραμένει μικρότερη από την πολυπλοκότητα του αντίστοιχου $(n, k, m+1)$ PCC. Εφόσον σε κάποια τμήματα του καινούργιου trellis module η διάσταση του χώρου καταστάσεων είναι ίση με m και σε κάποια ίση με $m+1$, ο χώρος αναζήτησης (για συγκεκριμένο trellis module) είναι μεγαλύτερος από το χώρο αναζήτησης του αντίστοιχου PCC με μνήμη m και μικρότερος από αυτόν του PCC με μνήμη $m+1$. Επίσης, οι ιδιότητες $P1-P4$ έχουν ως αποτέλεσμα τη μείωση του αρχικού χώρου αναζήτησης. Οι ομοιότητες των νέων κωδίκων με την κατηγορία των PCCs μας επιτρέπουν να χρησιμοποιήσουμε γνωστές τεχνικές [15] για να επιταχύνουμε τη διαδικασία αναζήτησης. Για παράδειγμα, αν όλοι οι γεννιότερες που απαρτίζουν έναν κωδικοποιητή έχουν άρτιο βάρος, τότε ο κωδικοποιητής απορρίπτεται ως καταστροφικός. Επίσης, μεταθέσεις των γεννητόρων που αντιστοιχούν στο ίδιο τμήμα του trellis module οδηγούν σε ισοδύναμο κώδικα, κ.τ.λ.

Η αναζήτηση έγινε για ένα μεγάλο εύρος ρυθμών και μεγεθών μνήμης, και οδήγησε σε πολλούς καλούς κώδικες. Πάνω από το 80% των κωδίκων που παρατίθενται στους Πίνακες 3.1, 3.2 και 3.3 παρουσιάζονται για πρώτη φορά και έχουν καλύτερα φάσματα από τους αντίστοιχους PCC. Από μια σκοπιά αυτό το γεγονός δεν προκαλεί έκπληξη. Είναι ευρέως αποδεκτό στη βιβλιογραφία [31], [29, 30, 32, 33], ότι ο βέλτιστος κώδικας με πολυπλοκότητα TC_1 θα είναι καλύτερος από το βέλτιστο κώδικα με πολυπλοκότητα TC_2 , για $TC_1 > TC_2$. Αυτό όμως δεν ισχύει για οποιονδήποτε κώδικα με πολυπλοκότητα TC_1 . Οι Πίνακες 3.1, 3.2 και 3.3 περιέχουν τα αποτελέσματα της αναζήτησης, καθώς και πληροφορίες για το ρυθμό τους, τη μνήμη, την (ελάχιστη) trellis πολυπλοκότητα TC (ανά bit πληροφορίας), την ελάχιστη απόσταση d_f , τους γεννιότερες, και τους πρώτους έξι όρους των φασμάτων τους. Οι γεννιότερες παρουσιάζονται στο οχταδικό σύστημα. Για παράδειγμα, ο αριθμός 173 αντιστοιχεί στην ακολουθία [1111011], όπου η πρώτη συνιστώσα αντιστοιχεί στο τρέχον bit πληροφορίας. Η τελευταία στήλη του κάθε πίνακα περιέχει την d_f και τους πρώτους όρους των φασμάτων των καλύτερων PCC που παρουσιάζονται στο [15] και αποτέλεσαν το σημείο της αφετηρίας μας. Η σύγκριση με τους αντίστοιχους PCC μας δίνει τη δυνατότητα να παρατηρήσουμε τη βελτίωση (των φασμάτων) που επιτεύχθηκε με την αύξηση της trellis πολυπλοκότητας. Σε μερικές περιπτώσεις, μικρή αύξηση της trellis πολυπλοκότητας οδήγησε σε σημαντική βελτίωση των φασμάτων. Αναφορικά με τα αποτελέσματα σημειώνουμε τα εξής:

- Μερικοί από τους νέους κώδικες έχουν μεγαλύτερη ελάχιστη απόσταση d_f από τους αντίστοιχους PCC με μνήμη m , αυξάνοντας ελάχιστα την trellis πολυπλοκότητα TC .

Τέτοιοι κώδικες είναι για παράδειγμα οι $(5, 3, 3)$, $(5, 3, 5)$, $(5, 3, 8)$, $(7, 4, 7)$, $(6, 5, 8)$ με $TC = 716.8$, $(7, 6, 3)$ με $TC = 21.3$, $(7, 6, 4)$, $(8, 7, 8)$ με $TC = 731.4$.

- Κάποιοι από τους νέους κώδικες έχουν την ίδια d_f με τους αντίστοιχους PCC μνήμης $m + 1$. Έτσι, έχουν παρόμοια ασυμπτωτική απόδοση (για μεγάλο λόγο σήματος προς θόρυβο) με τους αντίστοιχους $(n, k, m + 1)$ PCC και μικρότερη trellis πολυπλοκότητα. Κώδικες που ανήκουν σε αυτήν την κατηγορία είναι οι: $(7, 4, 3)$, $(6, 5, 6)$, $(7, 5, 2)$, $(7, 5, 4)$ με $TC = 70.4$, $(8, 7, 3)$ με $TC = 27.4$, $(8, 7, 4)$ με $TC = 54.9$, $(7, 6, 7)$ με $TC = 384$, $(7, 3, 3)$.
- Πολλοί από τους νέους κώδικες που περιέχονται στους Πίνακες 3.1, 3.2 και 3.3 επιτυγχάνουν μία συγκεκριμένη τιμή της ελάχιστης απόστασης d_f , έχοντας τη μικρότερη (γνωστή) πολυπλοκότητα trellis (για συγκεκριμένο ρυθμό). Παραδείγματα αποτελούν οι: $(7, 3, 7)$, $(7, 4, 7)$, $(7, 4, 8)$, $(7, 5, 2)$, $(5, 3, 3)$, $(8, 7, 8)$ με $TC = 731.4$, $(7, 5, 5)$, $(7, 5, 8)$.
- Σε πολλές άλλες περιπτώσεις, μικρή αύξηση της TC ήταν αρκετή για να βελτιώσει τα φάσματα (για την ίδια d_f) των καλύτερων PCC με μνήμη m ($(4, 3, 3)$, $(4, 3, 4)$, $(4, 3, 7)$, $(5, 4, 4)$, $(6, 5, 3)$ με $TC = 22.4$, $(7, 6, 7)$ με $TC = 341.3$, $(8, 7, 6)$, $(8, 7, 7)$).

Συγκρίσεις των αποτελεσμάτων μας με κώδικες που δημοσιεύονται στη βιβλιογραφία συνοψίζονται στον Πίνακα 3.4. Στα [29] και [30], οι συγγραφείς σχεδιάζουν $(n-1, n)$ συγκεραστικούς κώδικες διαφόρων ρυθμών και τιμών της πολυπλοκότητας trellis. Οι κώδικες που προέκυψαν από την αναζήτηση στην παρούσα εργασία, και οι κώδικες που δημοσιεύονται στα [29] και [30], για τις ίδιες παραμέτρους (ρυθμό και TC) έχουν στις περισσότερες περιπτώσεις παρόμοια φάσματα, όπως φαίνεται και στον Πίνακα 3.4. Η αναζήτηση στην παρούσα εργασία γίνεται σε μία περιορισμένη κατηγορία κωδίκων, ενώ στα [29] και [30] εξετάζονται όλοι οι κώδικες για συγκεκριμένο ρυθμό και πολυπλοκότητα. Παρόλ' αυτά, τα αποτελέσματα είναι συγκρίσιμα. Επιπλέον, η αναζήτηση σε μια περιορισμένη κατηγορία κωδίκων μας έδωσε τη δυνατότητα να εξετάσουμε πολύ μεγαλύτερες τιμές μνήμης και πολυπλοκότητας (σε κάποιες περιπτώσεις διπλασιάσαμε τη μέγιστη τιμή μνήμης που εξετάζεται στα [29] και [30]).

Στα [32] και [33], οι συγγραφείς αναζητούν καλούς κώδικες διαφόρων ρυθμών και τιμών της TC στην κατηγορία των GPCC. Από τη μορφή των αντίστοιχων \mathbf{G} και $\hat{\mathbf{G}}$, εύκολα προκύπτει πως η προτεινόμενη κατηγορία κωδίκων αποτελεί υποσύνολο των GPCC. Συγκρίσεις (Πίνακας 3.4) οδηγούν στο συμπέρασμα ότι οι προτεινόμενοι κώδικες έχουν συγκρίσιμα φάσματα με τους καλύτερους GPCC που δημοσιεύονται στα προαναφερθέντα άρθρα. Επίσης, η αναζήτηση στη συγκεκριμένη κατηγορία μας επέτρεψε να εξετάσουμε (για κάποιους ρυθμούς) περιπτώσεις κωδίκων με ελάχιστη TC μεγαλύτερη κατά μία τάξη μεγέθους, από τις περιπτώσεις που εξετάζονται στα [32] και [33].

Συγκεραστικοί κώδικες χαμηλής πολυπλοκότητας σχεδιάζονται επίσης στα [31], [28] και [27]. Σχεδόν όλοι οι κώδικες που βρήκαμε στην αναζήτησή μας έχουν καλύτερη απόδοση. Αυτό είναι

Πίνακας 3.1. Αποτελέσματα αναζήτησης

k/n	m	TC	d_f	γεννήτορες ^{1 2}	φάσματα $\frac{A_{d_f}, A_{d_f+1} \dots}{I_{d_f}, I_{d_f+1} \dots}$	PCC $d_f, \frac{A_{d_f}}{I_{d_f}}$
2/3	3	32	4	11,(32, 27)	$\frac{1,5,24,71,238,862}{1,12,102,412,1821,7804}$	4, $\frac{2}{5}$
2/3	4	64	5	31,(76, 43)	$\frac{3,11,39,135,519,1902}{5,45,218,949,4518,19355}$	5, $\frac{5}{15}$
2/3	5	128	6	51,(156, 117)	$\frac{6,27,70,285,1103,4063}{26,129,494,2446,10878,46500}$	6, $\frac{16}{56}$
2/3	6	256	7*	155,(236, 357)	$\frac{17,53,133,569,2327,8624}{86,360,1148,5767,27277,114524}$	6, $\frac{1}{1}$
2/3	7	512	8	275,(712, 513)	$\frac{43,0,547,0,8139,0}{230,0,5296,0,103967,0}$	8, $\frac{66}{395}$
2/3	8	1024	8	675,(1036, 1753)	$\frac{6,42,153,510,1853,7338}{20,284,1312,5164,22192,99382}$	8, $\frac{9}{38}$
2/5	4	96	10*	(26,36),(77, 51, 55)	$\frac{7,0,32,0,66,0}{12,0,134,0,328,0}$	9, $\frac{3}{5}$
2/5	5	224	11	(67,76),(111, 132, 157)	$\frac{4,10,14,21,39,73}{7,32,62,108,241,491}$	11, $\frac{6}{14}$
2/5	6	448	12	(131,143),(276, 235, 353)	$\frac{1,10,18,18,30,67}{1,31,78,95,158,433}$	12, $\frac{2}{3}$
2/5	7	896	14	(235,373),(736, 551, 457)	$\frac{10,0,45,0,134,0}{26,0,212,0,855,0}$	14, $\frac{15}{42}$
2/5	8	1792	15	(435,747),(1762, 1153, 1245)	$\frac{5,16,16,28,68,102}{11,60,70,152,388,660}$	15, $\frac{11}{43}$
3/4	3	26.7	4	15,16,(25, 37)	$\frac{10,46,202,949,4444,20840}{31,237,1565,9389,53751,297976}$	4, $\frac{29}{124}$
3/4	4	53.3	4	25,31,(42, 67)	$\frac{2,22,104,489,2436,12207}{2,92,734,4742,29645,178472}$	4, $\frac{5}{10}$
3/4	5	106.7	5	57,41,(123, 146)	$\frac{9,55,276,1308,6740,34757}{27,376,2524,14860,92772,555164}$	5, $\frac{13}{51}$
3/4	6	256	6	127,245,(316, 247)	$\frac{27,118,529,2978,15201,79518}{159,910,5536,37465,223804,1348149}$	6, $\frac{45}{276}$
3/4	7	512	6	341,663,(422, 667)	$\frac{5,73,305,1515,8172,42055}{13,691,3396,20848,133658,780830}$	6, $\frac{10}{35}$
3/4	8	1024	7*	661,1353,(1422, 1353)	$\frac{36,182,814,4190,22343,119005}{260,1838,10520,63364,391151,2367368}$	6, $\frac{1}{1}$
3/5	3	32	5*	15,(27, 36),(15,17)	$\frac{4,13,34,91,270,786}{7,39,151,524,1860,6360}$	4, $\frac{1}{1}$
3/5	4	64	6	26,(41, 73),(27,35)	$\frac{13,0,99,0,961,0}{35,0,586,0,8153,0}$	6, $\frac{18}{54}$
3/5	5	128	7*	45,(166, 127),(43,67)	$\frac{13,39,86,269,847,2528}{43,198,572,2054,7495,25672}$	6, $\frac{1}{1}$
3/5	6	298.7	8	(165,127),243,(276, 317)	$\frac{19,53,120,374,1166,3692}{83,277,920,3210,11476,41021}$	8, $\frac{36}{142}$
3/5	7	512	8	221,(636, 503),(271,375)	$\frac{4,29,71,196,615,1828}{6,137,450,1531,5570,18388}$	8, $\frac{5}{18}$
3/5	8	1024	9*	545,(1676, 1023),(561,725)	$\frac{7,41,97,317,937,2924}{19,232,682,2728,9145,32598}$	8, $\frac{1}{3}$
3/6	4	74.7	8*	(26,32),(75, 57),(27,35)	$\frac{21,0,74,0,461,0}{66,0,364,0,3256,0}$	7, $\frac{5}{8}$
3/7	2	21.3	6	(5,7),(5,7),(17, 12, 16)	$\frac{3,9,4,18,40,33}{4,21,12,72,194,187}$	6, $\frac{6}{9}$
3/7	3	40	8*	(13,17),(13,17),(25, 26, 36)	$\frac{13,0,32,0,137,0}{29,0,132,0,724,0}$	7, $\frac{3}{4}$
3/7	4	117.3	9*	(24,35),(47, 73),(51, 76, 67)	$\frac{6,10,14,30,46,117}{11,32,62,134,249,719}$	8, $\frac{1}{1}$
3/7	5	234.6	10	(43,67),(145, 155),(105, 152, 177)	$\frac{2,13,21,28,63,119}{2,40,97,139,368,776}$	10, $\frac{11}{22}$
3/7	6	469.2	12	(165,141),(353, 277),(316, 263, 351)	$\frac{15,0,65,0,239,0}{41,0,313,0,1578,0}$	12, $\frac{32}{103}$
3/7	7	938.4	13*	(255,271),(567, 633),(576, 723, 625)	$\frac{9,21,32,70,121,223}{25,88,166,389,790,1617}$	12, $\frac{2}{4}$
3/7	8	1876.8	14	(545,731),(1263, 1767),(1762, 1273, 1061)	$\frac{8,25,23,57,141,247}{22,95,102,325,926,1747}$	14, $\frac{25}{81}$

¹ Οι γεννήτορες σε παρένθεση αντιστοιχούν σε κωδικά bit του ίδιου τμήματος του trellis module.

² Οι υπεργραμμισμένοι γεννήτορες αντιστοιχούν σε τμήματα l με $m_l = m + 1$.

* Αυτός ο κώδικας έχει μεγαλύτερη d_f από τον καλύτερο PCC με μήνιμη m .

Πίνακας 3.2. Αποτελέσματα αναζήτησης (συνέχεια)

k/n	m	TC	d_f	γεννήτορες ^{1 2}	φάσματα $\frac{A_{d_f}, A_{d_f+1}, \dots}{I_{d_f}, I_{d_f+1}, \dots}$	PCC $d_f, \frac{A_{d_f}}{I_{d_f}}$
4/5	3	24	3	17,15,11,(22, 37)	$\frac{3,23,154,873,5023,29400}{5,100,1114,8916,65769,468010}$	$3, \frac{6}{11}$
4/5	3	32	3	11,37,31,(22, 27)	$\frac{1,21,143,780,4645,27881}{1,71,942,7766,59834,440359}$	
4/5	4	56	4	31,35,51,(41, 77)	$\frac{8,72,427,2598,16843,106461}{21,443,4048,32299,258779,1950187}$	$4, \frac{30}{159}$
4/5	5	96	4	57,73,71,(102, 157)	$\frac{2,43,282,1655,10853,70203}{4,249,2635,20899,169414,1306213}$	$4, \frac{4}{11}$
4/5	6	192	5	165,165,151,(202, 307)	$\frac{16,143,883,5480,36515,242254}{79,1309,10922,84630,675872,5222766}$	$5, \frac{22}{99}$
4/5	6	224	5	173,171,243,(312, 263)	$\frac{14,121,755,4794,31777,211392}{61,1060,9019,72129,574891,4472826}$	
4/5	7	448	6	243,207,735,(562, 773)	$\frac{106,0,4741,0,208623,0}{822,0,66691,0,4216577,0}$	$6, \frac{134}{1015}$
4/5	8	896	6	645,561,1623,(1772, 1307)	$\frac{24,184,1155,7889,52253,349075}{157,1920,16177,135006,1053523,8122683}$	$6, \frac{55}{364}$
4/6	4	80	6*	32,(47, 65),(67,(46, 67)	$\frac{46,0,545,0,7458,0}{181,0,4027,0,77903,0}$	$5, \frac{10}{28}$
4/7	2	20	4	5,(13, 17),(13, 16),(5,7)	$\frac{1,10,23,48,135,328}{1,27,87,268,916,2592}$	$4, \frac{3}{4}$
4/7	3	40	6*	13,(35, 27),(36, 27),(13,17)	$\frac{22,0,148,0,1033,0}{59,0,790,0,7819,0}$	$5, \frac{5}{8}$
4/7	4	64	6	25,(52, 67),(33,37),(23,35)	$\frac{5,23,44,110,315,921}{11,91,211,687,2246,7481}$	$6, \frac{8}{18}$
4/7	4	80	6	25,(47, 61),(46, 67),(25,33)	$\frac{3,19,41,107,291,786}{3,68,207,642,2051,6498}$	$6, \frac{8}{18}$
4/7	5	176	7	(57,75),(151, 133) ,163,(122, 177)	$\frac{5,27,69,144,428,1230}{13,114,383,976,3391,11007}$	$7, \frac{9}{21}$
4/7	6	320	8	155,(367, 251),(246, 323),(175,133)	$\frac{7,41,70,175,549,1582}{18,204,446,1316,4491,14930}$	$8, \frac{18}{66}$
4/7	7	512	9*	(271,227),(225,363) ,317,(422, 667)	$\frac{29,68,131,379,1131,3466}{133,410,1001,3282,11089,37026}$	$8, \frac{1}{1}$
4/7	8	1152	10*	(545,635),(445,731) ,1563,(1772, 1307)	$\frac{52,0,363,0,2903,0}{267,0,2781,0,28855,0}$	$9, \frac{6}{18}$
4/8	4	72	8	(31,33),(33,37),(25,37),(72, 53)	$\frac{31,0,111,0,610,0}{88,0,599,0,4015,0}$	$8, \frac{49}{178}$
5/6	3	22.4	3	17,11,15,16,(25, 37)	$\frac{8,73,487,3270,22475,153734}{21,498,4903,44543,383395,3149553}$	$3, \frac{15}{61}$
5/6	3	28.8	3	13,11,33,33,(22, 27)	$\frac{4,53,398,2720,19440,139219}{9,279,3629,35762,327057,2854421}$	
5/6	4	51.2	4	33,21,27,63,(46, 67)	$\frac{64,0,3724,0,211188,0}{338,0,49781,0,4488299}$	$4, \frac{111}{754}$
5/6	5	102.4	4	55,61,71,127,(132, 141)	$\frac{10,122,918,6991,55282,434855}{46,999,11268,111107,1078411,10048936}$	$4, \frac{15}{58}$
5/6	6	230.4	5*	113,155,373,271,(246, 323)	$\frac{48,370,2783,22386,180413,1450732}{384,4279,43852,437790,4209934,39381311}$	$4, \frac{3}{5}$
5/6	7	358.4	5	237,327,357,321,(436, 633)	$\frac{23,238,1752,14115,114373,923499}{124,2618,25129,262388,2553616,24203617}$	$5, \frac{29}{202}$
5/6	8	716.8	6*	545,771,645,575,(1516, 1277)	$\frac{263,0,16530,0,1087894,0}{3021,0,317457,0,28983900,0}$	$5, \frac{11}{66}$
5/6	8	921.6	6*	541,531,1563,1567,(1572, 1143)	$\frac{209,0,13411,0,883735,0}{2160,0,244500,0,22828475,0}$	

¹ Οι γεννήτορες σε παρένθεση αντιστοιχούν σε κωδικά bit του ίδιου τμήματος του trellis module.

² Οι υπεργραμμισμένοι γεννήτορες αντιστοιχούν σε τμήματα l με $m_l = m + 1$.

* Αυτός ο κώδικας έχει μεγαλύτερη d_f από τον καλύτερο PCC με μνήμη m .

Πίνακας 3.3. Αποτελέσματα αναζήτησης (συνέχεια)

k/n	m	TC	d_f	γεννήτορες ^{1 2}	φάσματα $\frac{A_{d_f}, A_{d_f+1}, \dots}{I_{d_f}, I_{d_f+1}, \dots}$	PCC $d_f, \frac{A_{d_f}}{I_{d_f}}$
5/7	2	19.2	4*	4, (15, 13), 13, 17, (16, 13)	$\frac{43, 0, 501, 0, 6608, 0}{162, 0, 3608, 0, 67471, 0}$	3, $\frac{4}{6}$
5/7	3	25.6	4	15, (36, 27), 13, 13, (13, 15)	$\frac{10, 52, 169, 712, 3060, 12489}{22, 274, 1365, 7146, 37193, 181546}$	4, $\frac{17}{64}$
5/7	4	57.6	4	21, 37, (27, 35), 51, (55, 63)	$\frac{1, 27, 113, 380, 1752, 7548}{1, 119, 704, 3227, 18138, 91296}$	4, $\frac{2}{2}$
5/7	4	70.4	5*	25, 27, (51, 65), 75, (56, 73)	$\frac{25, 112, 388, 1654, 7579, 33203}{111, 775, 3689, 19088, 101825, 514540}$	
5/7	4	76.8	5*	25, (57, 73), 51, 71, (46, 63)	$\frac{21, 86, 315, 1392, 6236, 27245}{71, 500, 2619, 14090, 74774, 381640}$	
5/7	5	128	6*	45, (57, 73), 165, 133, (132, 147)	$\frac{96, 0, 1649, 0, 32659, 0}{572, 0, 17007, 0, 465097, 0}$	5, $\frac{10}{28}$
5/7	6	256	6	145, (167, 113), 335, 277, (316, 263)	$\frac{22, 102, 384, 1833, 8252, 37310}{97, 727, 3633, 20508, 109692, 571194}$	6, $\frac{48}{223}$
5/7	7	409.5	6	207, (321, 277), 325, 341, (422, 667)	$\frac{7, 53, 208, 958, 4298, 19142}{13, 344, 1712, 9977, 53240, 273515}$	6, $\frac{20}{45}$
5/7	7	563.2	6	267, 317, (523, 735), 765, (532, 763)	$\frac{4, 47, 208, 837, 3791, 17453}{7, 313, 1820, 8560, 47387, 254845}$	
5/7	8	921.6	7*	545, (725, 531), 645, 1107, (1672, 1077)	$\frac{37, 127, 461, 2314, 10567, 47860}{233, 1065, 4938, 29340, 154941, 796354}$	6, $\frac{2}{4}$
6/7	3	21.3	3*	11, 13, 17, 13, 15, (26, 37)	$\frac{21, 169, 1291, 10101, 78907, 616091}{97, 1405, 15901, 162398, 1566130, 14559919}$	2, $\frac{1}{1}$
6/7	3	26.7	3*	15, 13, 12, 27, 37, (26, 37)	$\frac{12, 124, 1024, 8271, 67903, 557409}{38, 906, 12120, 135099, 1404087, 13953554}$	
6/7	4	42.6	4*	35, 21, 25, 35, 31, (62, 57)	$\frac{190, 0, 13720, 0, 1011065, 0}{1575, 0, 223811, 0, 24416278, 0}$	3, $\frac{6}{11}$
6/7	5	96	4	65, 51, 53, 63, 177, (115, 147)	$\frac{25, 281, 1341, 21796, 203063, 1875784}{113, 2934, 35898, 427087, 4883763, 53575361}$	4, $\frac{39}{223}$
6/7	6	234.6	4	177, 163, 341, 243, 337, (266, 347)	$\frac{5, 114, 967, 8938, 85299, 805267}{14, 1050, 13292, 161280, 1899272, 21400055}$	4, $\frac{7}{36}$
6/7	7	341.3	4	313, 277, 315, 225, 251, (722, 437)	$\frac{1, 76, 685, 6225, 59281, 563189}{4, 725, 9573, 116284, 1371685, 15569093}$	4, $\frac{4}{10}$
6/7	7	384	5*	337, 235, 221, 251, 563, (436, 713)	$\frac{74, 702, 6285, 59886, 568643, 5393396}{703, 10012, 119937, 1412721, 1598754, 175674996}$	
6/7	8	682.6	5	531, 725, 761, 625, 471, (1672, 1077)	$\frac{27, 322, 2989, 28434, 269581, 2578988}{204, 3848, 50664, 615684, 7037073, 79082189}$	5, $\frac{39}{314}$
6/7	8	938.6	5	551, 545, 1227, 1523, 1747, (1756, 1243)	$\frac{15, 262, 2280, 21854, 207207, 1986819}{102, 3417, 40021, 484145, 5527775, 62001438}$	
7/8	3	20.6	2	15, 13, 15, 17, 15, 16, (31, 23)	$\frac{1, 31, 277, 2463, 22125, 198704}{1, 178, 2788, 35580, 416721, 4617255}$	2, $\frac{2}{2}$
7/8	3	27.4	3*	15, 17, 13, 33, 33, 21, (25, 37)	$\frac{28, 274, 2456, 22412, 205112, 1875123}{139, 2488, 32760, 390333, 4406235, 47937296}$	
7/8	4	41.2	3	31, 33, 27, 21, 27, 25, (42, 73)	$\frac{9, 133, 1340, 13105, 131822, 1324499}{24, 1239, 18855, 248814, 3143048, 38002760}$	3, $\frac{12}{37}$
7/8	4	45.7	3	37, 31, 21, 37, 35, 41, (43, 71)	$\frac{6, 132, 1230, 12218, 124120, 1252635}{14, 1176, 16351, 221576, 2816610, 34207936}$	
7/8	4	54.9	4*	21, 35, 27, 77, 67, 71, (46, 77)	$\frac{302, 0, 29488, 0, 2945565, 0}{2896, 0, 556936, 0, 82387730, 0}$	
7/8	5	91.4	4	41, 63, 63, 63, 55, 137, (172, 113)	$\frac{60, 646, 6490, 68580, 727911, 7697452}{415, 8174, 115833, 1546982, 19964469, 248677631}$	4, $\frac{144}{1104}$
7/8	6	182.8	4	163, 125, 125, 143, 153, 261, (306, 277)	$\frac{20, 290, 2987, 32367, 348615, 3766973}{102, 3311, 49853, 705461, 9362763, 120305198}$	4, $\frac{31}{227}$
7/8	7	329.1	4	373, 375, 343, 221, 315, 235, (662, 573)	$\frac{6, 161, 1625, 17212, 190253, 2073239}{23, 1691, 26021, 364957, 5001886, 65125727}$	4, $\frac{13}{94}$
7/8	8	731.4	5*	545, 545, 625, 625, 771, 1603, (1536, 1053)	$\frac{89, 927, 9737, 106778, 1174169, 12904560}{1132, 15928, 222768, 3003521, 39177854, 498615773}$	4, $\frac{4}{11}$
7/8	8	877.7	5*	545, 545, 645, 1307, 1153, 1257, (1673, 1023)	$\frac{57, 709, 7443, 80924, 892652, 9810774}{601, 12164, 165708, 2225463, 29235346, 372938822}$	

¹ Οι γεννήτορες σε παρένθεση αντιστοιχούν σε κωδικά bit του ίδιου τμήματος του trellis module.

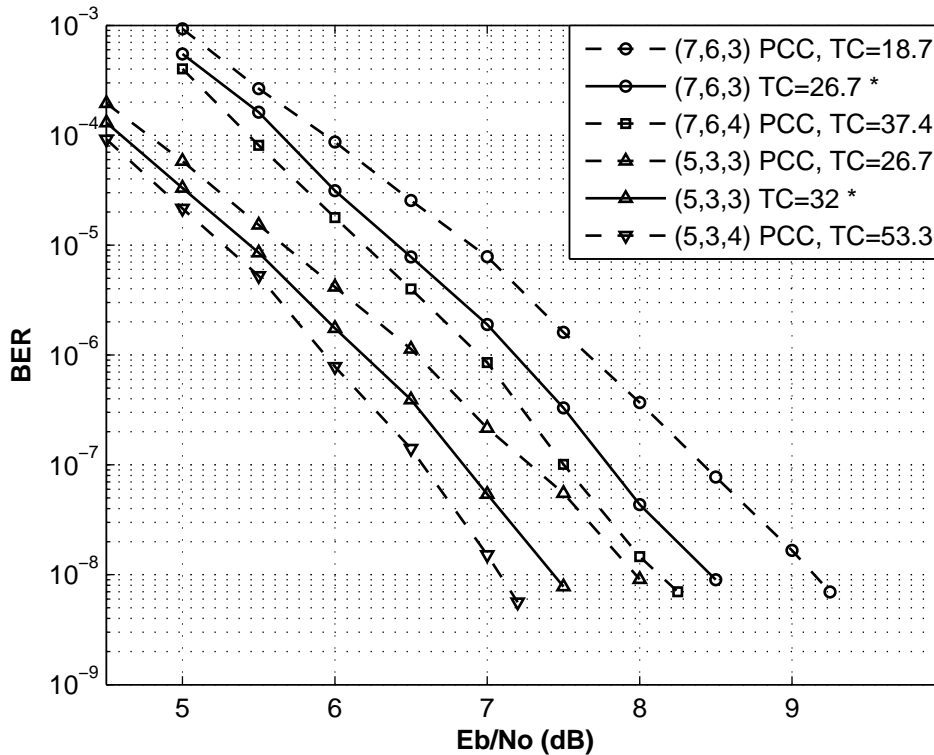
² Οι υπεργραμμισμένοι γεννήτορες αντιστοιχούν σε τμήματα l με $m_l = m + 1$.

* Αυτός ο κώδικας έχει μεγαλύτερη d_f από τον καλύτερο PCC με μήνιμη m .

Πίνακας 3.4. Συγκρίσεις με κώδικες από τη βιβλιογραφία

k/n	κώδικας από την παρούσα μελέτη $TC, d_f, \frac{A_{d_f}, A_{d_f+1}, A_{d_f+2}}{I_{d_f}, I_{d_f+1}, I_{d_f+2}}$	κώδικας από τη βιβλιογραφία $TC, d_f, \frac{A_{d_f}, A_{d_f+1}, A_{d_f+2}}{I_{d_f}, I_{d_f+1}, I_{d_f+2}}$
3/4	26.7, 4, $\frac{10,46,202}{31,237,1565}$	26.7, 4, $\frac{10,46,202}{31,237,1565}$ [29]
5/6	51.2, 4, $\frac{64,0,3724}{338,0,49781}$	51.2, 4, $\frac{25,225,1576}{135,2026,20913}$ [29]
7/8	27.4, 3, $\frac{28,274,2456}{139,2488,32760}$	27.4, 3, $\frac{28,274,2456}{134,2408,31736}$ [29]
2/3	64, 5, $\frac{3,11,39}{5,45,218}$	64, 5, $\frac{2,13,45}{5,51,248}$ [30]
2/3	128, 6, $\frac{6,27,70}{26,129,494}$	128, 6, $\frac{6,27,70}{26,129,494}$ [30]
2/3	256, 7, $\frac{17,53,133}{86,360,1148}$	256, 7, $\frac{17,53,133}{86,360,1148}$ [30]
2/3	1024, 8, $\frac{6,42,153}{20,284,1312}$	1024, 8, $\frac{6,42,153}{20,284,1312}$ [30]
3/4	53.3, 4, $\frac{2,22,104}{2,92,734}$	53.3, 4, $\frac{2,22,96}{3,93,680}$ [30]
3/5	32, 5, $\frac{4,13,34}{7,39,151}$	32, 4, $\frac{1,2,14}{1,3,53}$ [32]
3/5	64, 6, $\frac{13,0,99}{35,0,586}$	64, 6, $\frac{13,0,105}{40,0,610}$ [32]
3/6	74.7, 8, $\frac{21,0,74}{66,0,364}$	74.7, 8, $\frac{21,0,74}{72,0,373}$ [32]
2/5	96, 10, $\frac{7,0,32}{12,0,134}$	96, 10, $\frac{7,0,32}{16,0,145}$ [33]
3/5	128, 7, $\frac{13,39,86}{43,198,572}$	128, 7, $\frac{11,42,82}{40,248,549}$ [33]
3/7	21.3, 6, $\frac{3,9,4}{4,21,12}$	21.3, 6, $\frac{3,9,4}{4,21,12}$ [33]
3/7	117.3, 9, $\frac{6,10,14}{11,32,62}$	106.7, 9, $\frac{6,7,13}{16,21,45}$ [33]
4/7	20, 4, $\frac{1,10,23}{1,27,87}$	22, 4, $\frac{2,12,20}{3,34,80}$ [33]

αναμενόμενο, αφού οι (n, k, m) συγκεραστικοί κώδικες που κατασκευάζονται στα προαναφερθέντα άρθρα έχουν μικρότερη trellis πολυπλοκότητα. Στα Σχήματα 3.4 και 3.5, περιέχονται αποτελέσματα προσομοίωσης (ρυθμός λανθασμένων bit (Bit Error Rate - BER) συναρτήσει του λόγου σήματος προς θόρυβο) κάποιων από τους νέους κώδικες, για το AWGN κανάλι, χρησιμοποιώντας διαμόρφωση BPSK. Εκεί φαίνεται η καλύτερη απόδοση των νέων κωδίκων σε σχέση με τους PCC. Για παράδειγμα, ο κώδικας $(7, 6, 3)$ με $TC = 26.7$ (ανά bit πληροφορίας) εξοικονομεί 0.6dB και 0.7dB, συγκρινόμενος με τον αντίστοιχο PCC με μνήμη 3, για πιθανότητες σφάλματος 10^{-7} και 10^{-8} , αντίστοιχα (Σχ. 3.4). Αυτό επιτυγχάνεται αυξάνοντας κατά 43% την trellis πολυπλοκότητα TC . Για τις ίδιες πιθανότητες σφάλματος, ο $(7, 6, 4)$ PCC εξοικονομεί 0.9dB και 1dB αντίστοιχα, συγκρινόμενος με τον αντίστοιχο PCC μνήμης 3 (Σχ. 3.4). Αυτό επιτυγχάνεται αυξάνοντας κατά 100% την TC . Επίσης, ο κώδικας $(5, 4, 3)$ με $TC = 32$ (Σχ. 3.5(α')) εξοικονομεί 0.45dB και 0.6dB, συγκρινόμενος με τον αντίστοιχο PCC μνήμης 3, για πιθανότητες σφάλματος 10^{-5} και 10^{-6} αντίστοιχα (αυξάνοντας κατά 60% την TC), ενώ έχει και ελαφρώς καλύτερη απόδοση από τον $(5, 4, 4)$ PCC με $TC = 40$ (για τις ίδιες πιθανότητες σφάλματος). Ο κώδικας $(6, 5, 3)$ με $TC = 28.8$ (Σχ. 3.5(β')) εξοικονομεί 0.55dB και 0.6dB, συγκρινόμενος με τον αντίστοιχο PCC με μνήμη 3, για πιθανότητες σφάλματος 10^{-6} και 10^{-7} , αντίστοιχα (αυξάνοντας κατά 50% την TC). Ο ίδιος κώδικας έχει παρόμοια απόδοση με τον $(6, 5, 4)$ PCC (Σχ. 3.5(β')) για πιθανότητα σφάλματος από 10^{-3} μέχρι 10^{-7} .



Σχήμα 3.4. Προσομοιώσεις (AWGN κανάλι και διαμόρφωση BPSK) κωδίκων στους Πίνακες 3.1, 3.2 και 3.3 (σημειωμένοι με αστερίσκο) και PCCs.

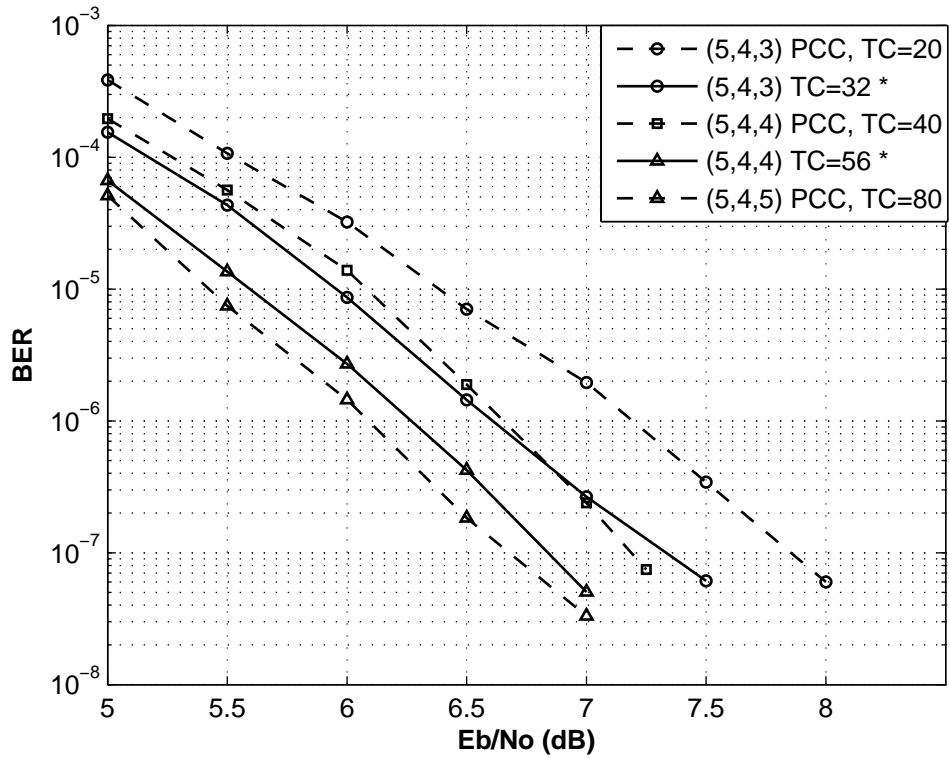
3.5 Ζητήματα Αποκωδικοποίησης

Στη συνέχεια θα δείξουμε πως το trellis module του κώδικα \mathcal{C} που κατασκευάσαμε στην ενότητα 3.3, πέραν από ελάχιστη trellis πολυπλοκότητα περιέχει και ελάχιστο αριθμό συνενώσεων. Κάθε τμήμα trellis $l \in T_{m,m}$ περιέχει 2^m συνενώσεις, αφού εκ κατασκευής δύο ακμές εισέρχονται σε κάθε μία από τις 2^m “τελικές καταστάσεις” του, δηλαδή τις καταστάσεις σε βάθος $l+1$. Αντίστοιχα, κάθε τμήμα trellis $l \in T_{m+1,m+1}$ περιέχει 2^{m+1} συνενώσεις, ενώ τα τμήματα trellis $l \in T_{m,m+1}$ δεν περιέχουν καθόλου συνενώσεις. Τέλος, για κάθε τμήμα $l \in T_{m+1,m}$, το ισοδύναμο ζευγάρι τμημάτων trellis περιέχει $2^{m+1} + 2^m$ συνενώσεις, αφού στο πρώτο και στο δεύτερο τμήμα δύο ακμές εισέρχονται σε κάθε μία από τις 2^{m+1} και 2^m “τελικές καταστάσεις” αντίστοιχα. Συνολικά, το trellis module που κατασκευάστηκε στην ενότητα 3.3 περιέχει

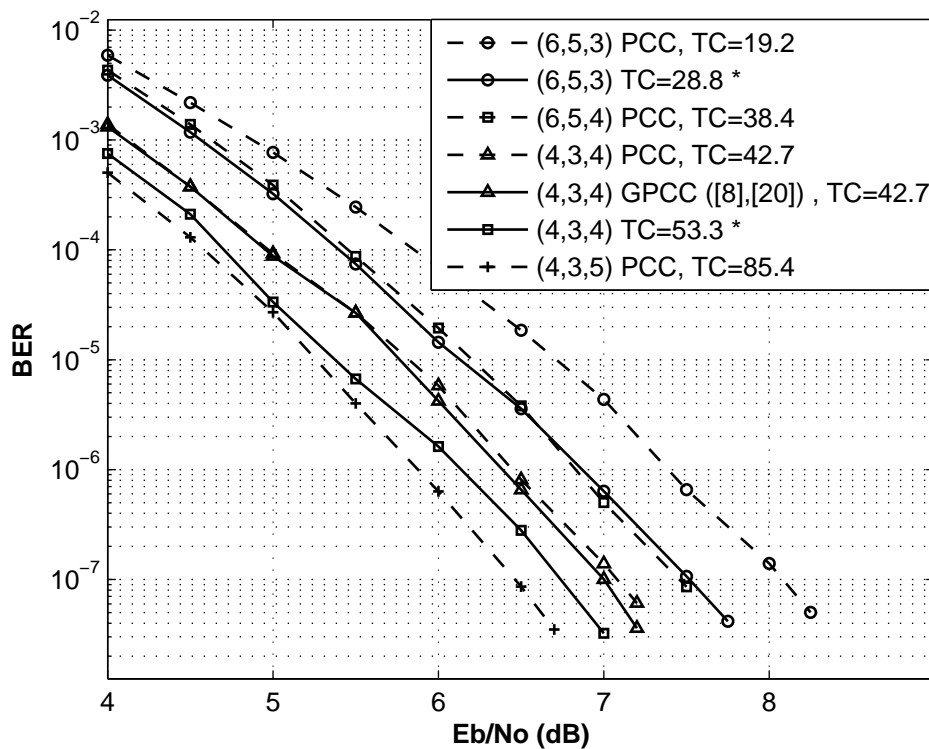
$$\tilde{E} = |T_{m,m}| \cdot 2^m + |T_{m+1,m+1}| \cdot 2^{m+1} + |T_{m+1,m}| \cdot (2^{m+1} + 2^m). \quad (3.16)$$

συνενώσεις.

Παρακάτω θα υπολογίσουμε τον ελάχιστο αριθμό συνενώσεων E που μπορεί να περιέχει ένα trellis module που αναπαριστά τον κώδικα \mathcal{C} . Γενικότερα, ο συνολικός αριθμός των συνενώσεων σε ένα trellis module \mathcal{M} ισούται με το συνολικό αριθμό ακμών μείον τον αριθμό των



(α')



(β')

Σχήμα 3.5. Προσομοιώσεις (AWGN κανάλι και διαμόρφωση BPSK) κωδικών στους Πίνακες 3.1, 3.2 και 3.3 (σημειωμένοι με αστερίσκο) και PCCs.

περιεχόμενων καταστάσεων [24, 25, 68]. Για τον υπολογισμό θα χρησιμοποιήσουμε τον trellis-κανονικό πίνακα γεννήτορα που κατασκευάστηκε στο Λήμμα 3.5, ο οποίος περιέχει όλες τις πληροφορίες που αφορούν το “προφίλ πολυπλοκότητας” του ελάχιστου trellis module. Πιο συγκεκριμένα, σε πρώτη φάση θα υπολογίσουμε τον συνολικό αριθμό καταστάσεων $|\Sigma|$ του ελάχιστου bit-oriented trellis module. Στη συνέχεια, εφόσον γνωρίζουμε την trellis πολυπλοκότητα TC του τελευταίου από το Θεώρημα 3.6 και εφόσον η trellis πολυπλοκότητα ενός bit-oriented trellis module ισούται με το πλήθος των ακμών που περιέχει, τότε ισχύει ότι

$$E = TC - |\Sigma|. \quad (3.17)$$

Η διάσταση του χώρου καταστάσεων σε βάθος $i+1$ ισούται με το σύνολο των ενεργών στοιχείων της στήλης i του πίνακα $\hat{\mathbf{G}}$, εξαιρουμένου του υπεργραμμισμένου στοιχείου (εάν υπάρχει). Με λίγα λόγια, το τελευταίο μη μηδενικό στοιχείο του ενεργού διαστήματος κάθε γραμμής δεν συνεισφέρει στη διάσταση των χώρων καταστάσεων. Όπως προκύπτει από την απόδειξη του Λήμματος 3.5, για $l \in T_{m,m}$, το στοιχείο n_l του πίνακα $\mathbf{G}_m(l)$ αποτελεί το τελευταίο μη μηδενικό στοιχείο του ενεργού διαστήματος κάποιας γραμμής του πίνακα $\tilde{\mathbf{G}}$. Το ίδιο ισχύει και για το στοιχείο n_l του πίνακα $\mathbf{G}_{m+1}(l)$, για $l \in T_{m,m}$. Τέλος, για $l \in T_{m+1,m}$, το στοιχείο p_l ενός εκ των πινάκων $\mathbf{G}_m(l)$ και $\mathbf{G}_{m+1}(l)$ και το στοιχείο n_l του άλλου πίνακα αποτελούν επίσης τα τελευταία μη μηδενικά στοιχεία των ενεργών διαστημάτων δύο γραμμών του πίνακα $\tilde{\mathbf{G}}$. Έτσι, δουλεύοντας ακριβώς όπως στην απόδειξη του Θεωρήματος 3.6 (χωρίς να λαμβάνουμε υπόψιν μας τα ενεργά στοιχεία που περιγράψαμε παραπάνω), προκύπτει ότι το σύνολο των καταστάσεων του ελάχιστου bit-oriented trellis module που περιγράφει τον \mathcal{C} δίνεται από τη σχέση

$$\begin{aligned} |\Sigma| = & \sum_{l \in T_{m,m}} ((n_l - 1)2^{m+1} + 2^m) + \sum_{l \in T_{m,m+1}} n_l 2^{m+1} + \sum_{l \in T_{m+1,m+1}} ((n_l - 1)2^{m+2} + 2^{m+1}) \\ & + \sum_{l \in T_{m+1,m}} ((n_l + p_l - 2)2^{m+1} + 2^m). \end{aligned} \quad (3.18)$$

Συνδυάζοντας τώρα τις σχέσεις (3.12), (3.17) και (3.18) προκύπτει ότι το E είναι ίσο με το \tilde{E} στη σχέση (3.16). Με λίγα λόγια, το trellis module που κατασκευάστηκε στην ενότητα 3.3 έχει τον ελάχιστο δυνατό αριθμό συνενώσεων, καθώς και ελάχιστη trellis πολυπλοκότητα. Μπορεί εύκολα ναδειχθεί ότι επί της ουσίας αποτελεί *τμηματοποίηση* του ελάχιστου bit-oriented trellis module, που προφανώς δεν αυξάνει ούτε την trellis πολυπλοκότητα, ούτε τον αριθμό των συνενώσεων.

Η υπολογιστική πολυπλοκότητα είναι το κατάλληλο μέτρο της πολυπλοκότητας αποκωδι-

Ή στον υπολογισμό του συνολικού αριθμού καταστάσεων ενός trellis module που απαρτίζεται από T τμήματα trellis προσμετράμε είτε τις καταστάσεις σε βάθος 0 ή τις καταστάσεις σε βάθος T (όχι και τα δύο σύνολα ταυτόχρονα), καθώς το βάθος T αποτελεί το βάθος 0 του επόμενου trellis module στο διάγραμμα trellis.

κοποίησης, ειδικά όταν η αποκωδικοποίηση πραγματοποιείται από λογισμικό που εκτελείται από κάποιον κεντρικό επεξεργαστή (software implementation).

Παρόλο που η υπολογιστική πολυπλοκότητα είναι μια καλά ορισμένη ποσότητα που μπορεί να υπολογιστεί σχετικά εύκολα για κάποιον κώδικα (ή πιο σωστά, για ένα συγκεκριμένο trellis module κάποιου κώδικα), η πολυπλοκότητα ενός αποκωδικοποιητή υλοποιημένου σε ολοκληρωμένο κύκλωμα (integrated circuit - IC) (hardware implementation) είναι ένα πολυδιάστατο πρόβλημα, το οποίο εξαρτάται ισχυρά από τη συγκεκριμένη αρχιτεκτονική της υλοποίησης. Παράμετροι κλειδιά μιας τέτοιας υλοποίησης αποτελούν η ταχύτητα της αποκωδικοποίησης (αυτή η παράμετρος εξαρτάται ισχυρά από την υπολογιστική πολυπλοκότητα) και το μέγεθος του ολοκληρωμένου κυκλώματος. Για παράδειγμα, θεωρήστε την υλοποίηση που βασίζεται στην αρχιτεκτονική πρόσθεσης-σύγκρισης-επιλογής (add-compare-select-ACS), όπου κάθε μονάδα ACS αντιστοιχεί σε μία κατάσταση του διαγράμματος trellis του κώδικα (οι ίδιες μονάδες ACS χρησιμοποιούνται σε όλα τα βήθη του trellis) [82, 83, 73]. Ένα χαρακτηριστικό του trellis module που επηρεάζει την πολυπλοκότητα αυτής της αρχιτεκτονικής είναι ο αριθμός των ακμών (ή κλαδιών) που καταλήγουν σε μία κατάσταση του trellis, και αντιστοιχεί στον αριθμό των ακμών που πρέπει να διαχειρίζονται ταυτόχρονα οι μονάδες ACS. Αυτό το χαρακτηριστικό επηρεάζει την ταχύτητα αποκωδικοποίησης και την πολυπλοκότητα υλοποίησης των μονάδων ACS. Άλλο χαρακτηριστικό είναι η *συνδεσιμότητα* των καταστάσεων (state connectivity). Οι μονάδες ACS συνδέονται μεταξύ τους, όπως οι καταστάσεις στο trellis, για την ανταλλαγή πληροφοριών (π.χ. μετρικές μονοπατιών) κατά τη διάρκεια της αποκωδικοποίησης. Ο αριθμός των ACS μονάδων με τις οποίες θα πρέπει να συνδέεται μία συγκεκριμένη μονάδα ACS εξαρτάται από τη δομή του trellis module. Ο συνολικός αριθμός αυτών των διασυνδέσεων επηρεάζει σημαντικά το μέγεθος της συγκεκριμένης αρχιτεκτονικής [82]. Συνοψίζοντας, οι προτεινόμενες κατασκευές βελτιστοποιούνται και σχεδιάζονται με βασικό κριτήριο την trellis πολυπλοκότητα, οπότε είναι κατάλληλες για υλοποίηση του αποκωδικοποιητή με τη χρήση λογισμικού. Για υλοποιήσεις του αποκωδικοποιητή σε ολοκληρωμένα κυκλώματα, το βέλτιστο trellis module διαφοροποιείται από αρχιτεκτονική σε αρχιτεκτονική.

Στη συνέχεια θα δείξουμε πως αν εκμεταλλευτούμε το γεγονός ότι το ελάχιστο trellis module (όπως κατασκευάζεται εδώ) για τους περισσότερους (n, k, m) κώδικες, με $k \neq n - 1$, είναι τμηματοποιημένο, μπορούμε να μειώσουμε τον αριθμό των προσθέσεων που πραγματοποιούνται από τον αλγόριθμο του Viterbi. Πράγματι, ας υποθέσουμε πως μια κωδική λέξη μήκους n' ενός δυαδικού γραμμικού κώδικα \mathcal{C} (μπλοκ ή συγκεραστικού) μεταδίδεται σε ένα κανάλι χωρίς μνήμη, δυαδικής εισόδου - συνεχούς εξόδου, και έστω ότι $p(r|j)$ είναι οι υπό συνθήκη pdf του καναλιού, $j \in GF_2$, $r \in \mathbb{R}$. Επίσης, έστω ότι $\mathbf{r} = [r_1, r_2, \dots, r_{n'}] \in \mathbb{R}^{n'}$ είναι η έξοδος του καναλιού. Ένας αποκωδικοποιητής μεγίστης πιθανοφάνειας αναζητάει την κωδική λέξη $\mathbf{v} = [v_1, v_2, \dots, v_{n'}] \in \mathcal{C}$ η οποία μεγιστοποιεί τη μετρική $\log P(\mathbf{r}|\mathbf{v})$, ή ισοδύναμα τη μετρική

[84]:

$$M(\mathbf{v}) = \sum_{i=1}^{n'} (-1)^{v_i} \mu_i$$

όπου $\mu_i = \log p(r_i|0) - \log p(r_i|1)$ είναι η μετρική του bit i . Οι μετρικές μ_i , $1 \leq i \leq n'$ αποτελούν την είσοδο του αποκωδικοποιητή. Υποθέστε τώρα ότι ο αλγόριθμος του Viterbi εφαρμόζεται χρησιμοποιώντας το ελάχιστο trellis module ενός (n, k, m) , $k \neq n - 1$ συγκεραστικού κώδικα, που ανήκει στη νέα κατηγορία. Υποθέστε επίσης ότι ένα από τα τμήματα trellis φέρει $n_l > 1$ κωδικά bit. Επίσης, έστω ότι το συγκεκριμένο τμήμα περιέχει 2^τ ακμές. Εκ κατασκευής $\tau \in \{m + 1, m + 2\}$. Ο αλγόριθμος του Viterbi θα εκτελέσει $n_l 2^\tau$ προσθέσεις στο συγκεκριμένο τμήμα $((n_l - 1)2^\tau$ για να αθροίσει τις μετρικές των bit της κάθε ακμής (μετρικές ακμών) και 2^τ για να ενημερώσει τις μετρικές των μονοπατιών). Επειδή το τμήμα φέρει περισσότερα από ένα κωδικά bit, πιο αποδοτικές μέθοδοι μπορούν να χρησιμοποιηθούν για τον υπολογισμό των μετρικών ακμής, όπως η *τεχνική εξαντλητικού υπολογισμού* (exhaustive computation technique - ECT) που παρουσιάστηκε στο [84]. Ακολουθώντας την ECT, για $n_l \leq \tau$, το τμήμα του trellis module περιέχει το πολύ 2^{n_l} διαφορετικές ετικέτες ακμής, κι επομένως 2^{n_l} διαφορετικές τιμές της μετρικής ακμής. Κάθε μία από αυτές τις τιμές υπολογίζεται από τις μετρικές των bit με $n_l - 1$ προσθέσεις. Εφόσον συμπληρωματικές ετικέτες ακμής αντιστοιχούν σε αντίθετες μετρικές ακμής, χρειάζεται να υπολογίσουμε μόνο 2^{n_l-1} μετρικές ακμής. Οι υπόλοιπες 2^{n_l-1} προκύπτουν με αλλαγή προσήμου, η οποία θεωρείται πράξη χωρίς κόστος. Επομένως, ο αριθμός προσθέσεων που πρέπει να γίνουν για τον υπολογισμό των μετρικών ακμής του συγκεκριμένου τμήματος είναι $\theta \leq 2^{n_l-1} \cdot (n_l - 1)$. Εάν $n_l > \tau$, ο μέγιστος αριθμός των διαφορετικών ετικετών ακμής είναι 2^τ . Από την ιδιότητα P_1 προκύπτει πως εάν μία (δυναμική) ετικέτα ακμής μήκους n_l περιέχεται στο συγκεκριμένο τμήμα του trellis module, τότε θα περιέχεται κι η συμπληρωματική της ετικέτα. Οπότε, ο αριθμός των προσθέσεων που πρέπει να γίνουν για τον υπολογισμό των μετρικών ακμής είναι $\theta \leq 2^{\tau-1} \cdot (n_l - 1)$. Επομένως, σε κάθε περίπτωση $\theta \leq 2^{\psi-1} (n_l - 1)$, όπου $\psi = \min(\tau, n_l)$. Επίσης, 2^τ προσθέσεις χρειάζονται για τον υπολογισμό των μετρικών των μονοπατιών. Έτσι, χρειάζονται συνολικά $2^{\psi-1} \cdot (n_l - 1) + 2^\tau$ προσθέσεις το πολύ, όταν χρησιμοποιείται η ECT. Εύκολα προκύπτει ότι $n_l 2^\tau > 2^{\psi-1} \cdot (n_l - 1) + 2^\tau$, για $n_l > 1$, και για τις δύο τιμές του ψ . Με άλλα λόγια, η ύπαρξη, στο trellis module, τμημάτων που φέρουν περισσότερα από ένα κωδικά bit μπορεί να μειώσει τον αριθμό των προσθέσεων (μεταξύ πραγματικών αριθμών) που πραγματοποιούνται από τον αλγόριθμο του Viterbi.

Για παράδειγμα, η ελάχιστη trellis πολυπλοκότητα του $(8, 4, 2)$ συγκεραστικού κώδικα που παρουσιάστηκε στο Παράδειγμα 3.1 είναι $TC = 18$ (ανά bit πληροφορίας), δηλαδή, 18 προσθέσεις χρειάζονται για να αποκωδικοποιηθεί ένα bit πληροφορίας όταν για την αποκωδικοποίηση χρησιμοποιείται το bit oriented ελάχιστο trellis. Εφόσον κάθε ένα από τα τρία τμήματα του ελάχιστου trellis module (Σχ. 3.2) φέρει δύο κωδικά bit, με τη χρήση της ECT χρειάζονται 13.5 προσθέσεις για την αποκωδικοποίηση ενός bit πληροφορίας, δηλαδή 25% λιγότερες προσθέσεις.

Επίσης, θεωρήστε τον $(5, 7, 4)$ συγκεραστικό κώδικα (Πίνακας 3.3), του οποίου η trellis πολυπλοκότητα είναι $TC = 70.4$ (ανά bit πληροφορίας). Το ελάχιστο trellis module αποτελείται από έξι τμήματα. Πέντε από αυτά φέρουν από ένα κωδικό bit, ενώ ένα φέρει δύο. Παρόλο που μόνο ένα τμήμα φέρει περισσότερα από ένα κωδικά bit, οι απαιτούμενες προσθήσεις είναι 58, όταν για την αποκωδικοποίηση χρησιμοποιείται το ελάχιστο trellis module (όπως κατασκευάζεται στην παρούσα μελέτη).

Αξίζει να σημειωθεί εδώ πως μπορεί να υπάρχει μία καλύτερη τμηματοποίηση (δηλαδή μία τμηματοποίηση που να οδηγεί σε λιγότερες πράξεις). Η βέλτιστη τμηματοποίηση είναι χαρακτηριστικό του κώδικα, οπότε διαφορετικοί κώδικες πιθανότατα έχουν διαφορετικά βέλτιστα τμηματοποιημένα trellis modules [84]. Από την άλλη όμως, η προτεινόμενη τμηματοποίηση (όταν συμβαίνει, δηλαδή για τους περισσότερους (n, k, m) κώδικες με $k \neq n - 1$) είναι καλή για όλους τους κώδικες της προτεινόμενης κατηγορίας, με την έννοια ότι μπορούν να αποκωδικοποιηθούν εκτελώντας λιγότερες πράξεις, απ' ότι όταν χρησιμοποιείται το bit oriented ελάχιστο trellis module.

3.6 Βέλτιστα Trellis Modules για $(n, n - 1)$ Συγκεραστικούς Κώδικες

Όπως είδαμε και στο κεφάλαιο 2, από τη μετάθεση των n κωδικών bit ενός συγκεραστικού κώδικα [25], [29] (από τη μετάθεση δηλαδή των στηλών ενός πίνακα γεννήτορα αυτού) προκύπτει ένας ισοδύναμος κώδικας, με την έννοια ότι οι δύο κώδικες έχουν πανομοιότυπα φάσματα. Παρόλ' αυτά, τα αντίστοιχα ελάχιστα trellis modules των δύο κωδίκων μπορεί να είναι εντελώς διαφορετικά μεταξύ τους. Ονομάζουμε βέλτιστο ελάχιστο trellis module ενός κώδικα αυτό το οποίο ελαχιστοποιεί ένα συγκεκριμένο μέτρο πολυπλοκότητας (π.χ. trellis πολυπλοκότητα, αριθμό καταστάσεων, μέγιστη διάσταση του χώρου καταστάσεων, αριθμό συνενώσεων, κλπ.) ανάμεσα στα ελάχιστα trellis modules όλων των δυνατών μεταθέσεων του κώδικα (γενικότερα δεν υπάρχει εγγύηση για την ύπαρξη μιας μετάθεσης, η οποία ελαχιστοποιεί όλες τις ποσότητες που σχετίζονται με την πολυπλοκότητα του trellis module). Σ' αυτή τη μελέτη θεωρούμε ως βέλτιστο trellis module, αυτό που ελαχιστοποιεί την trellis πολυπλοκότητα και τον αριθμό των συνενώσεων (θα δείξουμε ότι για έναν $(n, n - 1)$ συγκεραστικό κώδικα αυτές οι δύο ποσότητες μπορούν να ελαχιστοποιηθούν ταυτόχρονα), δηλαδή την υπολογιστική πολυπλοκότητα του αλγόριθμου του Viterbi.

Θεωρήστε έναν $(n, n - 1)$ συγκεραστικό κώδικα \mathcal{C} και τον αντίστοιχο $(n, 1)$ δυϊκό του κώδικα \mathcal{C}^\perp . Έστω $\mathbf{G}(D)$ ένας κανονικός πολυωνυμικός πίνακας γεννήτορας συνολικής μνήμης m ο οποίος παράγει τον \mathcal{C} . Τότε ισχύει ότι ο $\mathbf{H}(D) = [H^{(0)}(D) \ H^{(1)}(D) \ \dots \ H^{(n-1)}(D)]$ είναι ένας κανονικός πίνακας γεννήτορας με μνήμη m , ο οποίος παράγει τον \mathcal{C}^\perp [49], όπου

$$H^{(i)}(D) = h_0^{(i)} + h_1^{(i)}D + \dots + h_m^{(i)}D^m \quad (3.19)$$

είναι η ορίζουσα του $(n-1) \times (n-1)$ υποπίνακα που προκύπτει από τον $\mathbf{G}(D)$ εάν διαγράψουμε τη στήλη i . Επίσης, $\widehat{\mathbf{H}}(D) = \left[\widehat{H}^{(0)}(D) \quad \widehat{H}^{(1)}(D) \dots \widehat{H}^{(n-1)}(D) \right]$ είναι ένας κανονικός πίνακας γεννήτορας με μνήμη m , του *ανάστροφου*² (reciprocal) κώδικα $\widehat{\mathcal{C}}^\perp$ του \mathcal{C}^\perp , όπου

$$\widehat{H}^{(i)}(D) = D^m H^{(i)}(D^{-1}). \quad (3.20)$$

Συνδυάζοντας τις σχέσεις (3.19) και (3.20), οι πίνακες συντελεστών των όρων ελάχιστου και μέγιστου βαθμού του $\widehat{\mathbf{H}}(D)$ (τους οποίους θα συμβολίζουμε με $\widehat{\mathbf{H}}_0$ και $\widehat{\mathbf{H}}_m$ ³ αντίστοιχα), δίνονται από τις σχέσεις

$$\widehat{\mathbf{H}}_0 = \left[h_m^{(0)} \quad h_m^{(1)} \dots h_m^{(n-1)} \right] \quad (3.21)$$

$$\widehat{\mathbf{H}}_m = \left[h_0^{(0)} \quad h_0^{(1)} \dots h_0^{(n-1)} \right]. \quad (3.22)$$

Έστω $\chi = L(\widehat{\mathbf{H}}_0) - R(\widehat{\mathbf{H}}_m)$, $1 - n \leq \chi \leq n - 1$, όπου $L(\widehat{\mathbf{H}}_0)$ είναι η θέση του πρώτου μη μηδενικού στοιχείου του $\widehat{\mathbf{H}}_0$ και $R(\widehat{\mathbf{H}}_m)$ είναι η θέση του τελευταίου μη μηδενικού στοιχείου του $\widehat{\mathbf{H}}_m$, $0 \leq L(\widehat{\mathbf{H}}_0), R(\widehat{\mathbf{H}}_m) \leq n - 1$ [29]. Αποδεικνύεται στα [29], [30] ότι η trellis πολυπλοκότητα TC και ο συνολικός αριθμός καταστάσεων $|\Sigma|$ του bit oriented ελάχιστου trellis module του \mathcal{C} δίνεται από τις σχέσεις

$$TC = \begin{cases} (n - \chi)2^{m+1} + (\chi - 1)2^m, & \chi \geq 0 \\ (n - \chi - 1)2^{m+1}, & \chi < 0 \end{cases} \quad (3.23)$$

$$|\Sigma| = \begin{cases} n2^m - \chi2^{m-1}, & \chi \geq 0 \\ (n - \chi)2^m, & \chi < 0. \end{cases} \quad (3.24)$$

Επίσης, τα bit oriented ελάχιστα trellis modules των \mathcal{C} και $\widehat{\mathcal{C}}^\perp$ έχουν μεταξύ τους την ίδια τιμή της διάστασης του χώρου καταστάσεων σε όλα τα βάρη. Στο [29], μία μετάθεση ενός κώδικα καλείται βέλτιστη όταν παράγει έναν ισοδύναμο κώδικα, του οποίου το bit oriented ελάχιστο trellis module έχει τον ελάχιστο δυνατό συνολικό αριθμό καταστάσεων. Μία βέλτιστη μετάθεση για τον $\widehat{\mathcal{C}}^\perp$ είναι βέλτιστη και για τον \mathcal{C} . Επίσης, αποδεικνύεται ότι βέλτιστη (και για τους δύο κώδικες) είναι η μετάθεση που μεγιστοποιεί το χ . Αξίζει να σημειωθεί πως μία μετάθεση που μεγιστοποιεί το χ , ελαχιστοποιεί και την trellis πολυπλοκότητα TC του \mathcal{C} [29] (αυτό προκύπτει επίσης κι από τη σχέση (3.23)), ενώ δύο μεταθέσεις που αντιστοιχούν στην ίδια τιμή του χ είναι ισοδύναμες, όσον αφορά τις τιμές των TC και Σ . Μία βέλτιστη μετάθεση για τους $\widehat{\mathcal{C}}^\perp$ και

²Δοθέντος ενός συγκραστικού κώδικα \mathcal{C} και ενός κανονικού πίνακα γεννήτορα $\mathbf{G}(D)$ που τον παράγει, ο *ανάστροφος* του κώδικα $\widehat{\mathcal{C}}$ είναι ο κώδικας που παράγεται από τον (κανονικό) πίνακα γεννήτορα $\widehat{\mathbf{G}}(D)$ για τον οποίο ισχύει $\widehat{g}_j^{(l)}(D) = g_j^{(l)}(D^{-1})D^{\nu_j}$

³Ο πίνακας $\widehat{\mathbf{H}}_h$ συμπίπτει με τον $\widehat{\mathbf{H}}_m$ στην περίπτωση των $(n, n - 1, m)$ κωδίκων.

\mathcal{C} δίνεται στο [29],

$$\pi = \begin{cases} (A - B) \wedge (A \cap B) \wedge (I - A), & \text{εάν } A \cap B \neq \emptyset \\ A \wedge (I - (A \cup B)) \wedge B, & \text{εάν } A \cap B = \emptyset \end{cases} \quad (3.25)$$

όπου $A = \{i : \hat{\mathbf{H}}_m(i) \neq 0, i \in I = \{0, 1, \dots, n-1\}\}$, $B = \{i : \hat{\mathbf{H}}_0(i) \neq 0, i \in I = \{0, 1, \dots, n-1\}\}$, ενώ τα $\hat{\mathbf{H}}_0(i)$ και $\hat{\mathbf{H}}_m(i)$ συμβολίζουν τα i -οστά στοιχεία των $\hat{\mathbf{H}}_0$ και $\hat{\mathbf{H}}_m$ αντίστοιχα, και το $X \wedge Y$ συμβολίζει την αλύσωση δύο διατεταγμένων συνόλων X και Y .

Στη συνέχεια θα δείξουμε πως ο συνολικός αριθμός συνενώσεων E στο bit oriented ελάχιστο trellis module του \mathcal{C} μειώνεται, καθώς αυξάνεται το χ . Έτσι, μια μετάθεση που μεγιστοποιεί το χ , ελαχιστοποιεί και το συνολικό αριθμό συνενώσεων E . Όσον αφορά το bit-oriented trellis module ισχύει ότι $E = TC - |\Sigma|$ [24, 25], οπότε συνδυάζοντας τις σχέσεις (3.23) και (3.24) έχουμε

$$E = \begin{cases} (2n - 2 - \chi)2^{m-1}, & \chi \geq 0 \\ (n - 2 - \chi)2^m, & \chi < 0. \end{cases} \quad (3.26)$$

Από την (3.26) προκύπτει ότι η E είναι φθίνουσα συνάρτηση του χ . Έτσι, η μετάθεση στην (3.25), η οποία μεγιστοποιεί το χ (και κάθε άλλη μετάθεση που μεγιστοποιεί το χ) παράγει έναν ισοδύναμο κώδικα, του οποίου το bit-oriented ελάχιστο trellis module έχει τις μικρότερες δυνατές τιμές των TC , $|\Sigma|$ και E , ανάμεσα στα bit-oriented ελάχιστα trellis modules όλων των ισοδύναμων κωδίκων. Έτσι, αποτελεί βέλτιστη μετάθεση για το μέτρο της πολυπλοκότητας που χρησιμοποιείται σ' αυτήν την εργασία, δηλαδή την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης μεγίστης πιθανοφάνειας (Viterbi). Πρέπει να σημειωθεί εδώ πως αφού η E δεν είναι γνησίως φθίνουσα συνάρτηση του χ , μπορεί να πάρει την ελάχιστη τιμή της για διάφορες τιμές του χ πέραν της μέγιστης. Από την άλλη, η TC είναι γνησίως φθίνουσα συνάρτηση του χ (αυτό προκύπτει από την (3.23)), συνεπώς ελαχιστοποιείται μόνο όταν το χ πάρει τη μέγιστη τιμή του. Έτσι, μόνο μία μετάθεση που μεγιστοποιεί το χ , ελαχιστοποιεί την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης, δηλαδή ελαχιστοποιεί ταυτόχρονα την trellis πολυπλοκότητα και τον αριθμό των συνενώσεων.

Θεωρείστε τώρα έναν $(n, n-1, m)$ συγκεραστικό κώδικα \mathcal{C} της προτεινόμενης οικογένειας, όπως περιγράφεται στην ενότητα 3.1. Χωρίς βλάβη της γενικότητας θα υποθέσουμε ότι στο τελευταίο τμήμα trellis του αρχικού trellis module (το οποίο απαρτίζεται από $k = n-1$ τμήματα trellis) αντιστοιχούν δύο κωδικά bit, ενώ σε κάθε ένα από τα υπόλοιπα τμήματα αντιστοιχεί ένα bit. Σύμφωνα με την ιδιότητα P_4 , μόνο το τελευταίο τμήμα $l \in T_{m+1, m}$, καθώς πρέπει να ισχύει $n_l > 1$. Επομένως, υπάρχει $0 < j < n-2$, τέτοιος ώστε

1. $l \in T_{m, m}$, για $0 \leq l < j$,
2. $l \in T_{m+1, m+1}$, για $j < l < n-2$,

3. $j \in T_{m,m+1}$.

Σ'αυτό το σημείο υπενθυμίζουμε ότι στην περίπτωση των $(n, n-1, m)$ κωδίκων, η τεχνική που περιγράφεται στην ενότητα 3.3 παράγει το bit oriented ελάχιστο trellis module, αφού στο αρχικό module μόνο το τμήμα που ανήκει στο $T_{m,m+1}$ φέρει δύο κωδικά bit, το οποίο αντικαθίσταται από ισοδύναμο ζεύγος τμημάτων καθένα εκ των οποίων φέρει ένα κωδικό bit.

Σύμφωνα με το Θεώρημα 3.4, ο $(n-1) \times n$ (ισοδύναμος χρονικά αμετάβλητος) πίνακας γεννήτορας του \mathcal{C} είναι κανονικός (προφανώς θεωρούμε ότι δεν είναι καταστροφικός). Από τη σχέση (2.53) και την ιδιότητα P_1 , ο πίνακας των σταθερών όρων \mathbf{G}_0 είναι της μορφής

$$\mathbf{G}_0 = \begin{bmatrix} 1 & * & * & \dots & * & * \\ 0 & 1 & * & \dots & * & * \\ 0 & 0 & 1 & \dots & * & * \\ \vdots & \vdots & \ddots & \ddots & \vdots & \\ 0 & 0 & \dots & 0 & 1 & 1 \end{bmatrix}. \quad (3.27)$$

Με αστερίσκο συμβολίζουμε ένα στοιχείο που μπορεί να έχει είτε την τιμή 0, είτε την τιμή 1. Επίσης, όπως είδαμε και στην ενότητα 3.2, ο πίνακας \mathbf{G}_h μετατρέπεται έπειτα από κατάλληλη μετάθεση των γραμμών του στον πίνακα \mathbf{G}_{end} στην (3.7). Στην προκειμένη περίπτωση, ο \mathbf{G}_{end} είναι της μορφής

$$\mathbf{G}_{end} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ * & 1 & & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & & \\ * & * & \dots & 1 & 0 & 0 & \dots & 0 & 0 \\ * & * & \dots & * & * & 1 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots & \\ * & * & \dots & * & * & * & \dots & 1 & 0 \\ * & * & \dots & * & * & * & \dots & * & 1 \end{bmatrix}. \quad (3.28)$$

Στην (3.28) έχουμε υποθέσει ότι

$$\begin{bmatrix} \mathbf{G}_{m+1}(n-2) \\ \mathbf{G}_m(n-2) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ * & 1 \end{bmatrix}. \quad (3.29)$$

Παρόλ' αυτά, οποιαδήποτε άλλη μορφή του πίνακα (ο οποίος είναι μέγιστου βαθμού (full rank), διότι το τμήμα trellis $n-2$ είναι της μορφής $T_{m+1,m}$ και συνεπώς ισχύει η ιδιότητα $P4$) στο αριστερό μέλος της (3.29) μπορεί να μετασχηματιστεί στον πίνακα του δεξιού μέλους της (3.29) με στοιχειώδεις γραμμοπράξεις μεταξύ των γραμμών $n-3$ και $n-2$ στην (3.28).

Λήμμα 3.8. Για τον κανονικό πίνακα γεννήτορα $\mathbf{H}(D)$ μήμης m του δυϊκού κώδικα \mathcal{C}^\perp του \mathcal{C} , που περιγράφεται παραπάνω, ισχύει ότι

1. $h_0^{(n-2)} = h_0^{(n-1)} = 1$.
2. $h_m^{(i)} = 0$, για $0 \leq i < j$ και $h_m^{(j)} = 1$.

Απόδειξη. Το $h_0^{(i)}$, $0 \leq i \leq n - 1$, είναι ο σταθερός όρος της οριζουσας $H^{(i)}(D)$ του υποπίνακα του $\mathbf{G}(D)$, ο οποίος απαρτίζεται από όλες τις στήλες του τελευταίου πλιν της i -οστής. Επομένως, είναι εύκολο να επιβεβαιωθεί ότι το $h_0^{(i)}$ ισούται με την οριζουσα του υποπίνακα του \mathbf{G}_0 στην (3.27), ο οποίος απαρτίζεται από όλες τις στήλες του τελευταίου πλιν της i -οστής. Αντίστοιχα, το $h_m^{(i)}$, $0 \leq i \leq n - 1$ είναι ο συντελεστής του όρου με βαθμό m της $H^{(i)}(D)$. Οπότε, το $h_m^{(i)}$ είναι η οριζουσα του υποπίνακα του πίνακα συντελεστών των μεγιστοβάθμιων όρων \mathbf{G}_h του $\mathbf{G}(D)$, και συνεπώς του \mathbf{G}_{end} στην (3.28) (ο \mathbf{G}_{end} έχει προκύψει από το \mathbf{G}_h μεταθέτοντας κάποιες γραμμές), ο οποίος απαρτίζεται από όλες τις στήλες του τελευταίου πλιν της i -οστής. Επομένως,

1. Εάν διαγράψουμε οποιαδήποτε από τις δύο τελευταίες στήλες του \mathbf{G}_0 στην (3.27), προκύπτει ένας $(n - 1) \times (n - 1)$ άνω τριγωνικός υποπίνακας, του οποίου η κύρια διαγώνιος αποτελείται από μονάδες, οπότε η οριζουσα του ισούται με 1.
2. Εάν διαγράψουμε τη στήλη j (η μοναδική στήλη που περιέχει μόνο αστερίσκους και μηδενικά) από τον πίνακα \mathbf{G}_{end} στην (3.28), προκύπτει ένας $(n - 1) \times (n - 1)$ κάτω τριγωνικός υποπίνακας, του οποίου η κύρια διαγώνιος αποτελείται από μονάδες, οπότε η οριζουσα του ισούται με 1. Εάν διαγράψουμε οποιαδήποτε στήλη $0 \leq i < j$, τότε προκύπτει ένας $(n - 1) \times (n - 1)$ κάτω τριγωνικός υποπίνακας, του οποίου το στοιχείο i της κύρια διαγωνίου είναι μηδενικό, και συνεπώς η οριζουσα του ισούται με 0.

□

Συνδυάζοντας τις σχέσεις (3.21), (3.22) και το Λήμμα 3.8, προκύπτει ότι ο $\widehat{\mathbf{H}}_0$ και ο $\widehat{\mathbf{H}}_m$ είναι της μορφής

$$\widehat{\mathbf{H}}_0 = \begin{bmatrix} \underbrace{0 \ 0 \ \dots \ 0}_j & 1 & h_m^{(j+1)} & \dots & h_m^{(n-1)} \end{bmatrix} \quad (3.30)$$

$$\widehat{\mathbf{H}}_m = \begin{bmatrix} h_0^{(0)} & h_0^{(1)} & \dots & 1 & 1 \end{bmatrix}. \quad (3.31)$$

Όπως είδαμε και παραπάνω, η μετάθεση του \mathcal{C} που δίνεται στην (3.25) μεγιστοποιεί το χ (θα συμβολίζουμε τη μέγιστη τιμή του χ με χ_{opt}), και συνεπώς οδηγεί στο bit oriented ελάχιστο trellis module με τα ελάχιστα δυνατά TC , $|\Sigma|$ και E . Επίσης, οποιαδήποτε άλλη μετάθεση μεγιστοποιεί το χ είναι βέλτιστη. Στη συνέχεια θα υπολογίσουμε το χ_{opt} για έναν $(n, n - 1, m)$ κώδικα (δεδομένου ότι ο $\widehat{\mathbf{H}}(D)$ είναι ένας κανονικός πίνακας γεννήτορας που παράγει τον ανάστροφο του δυϊκού του κώδικα), βασιζόμενοι στη μετάθεση (3.25). Έστω ότι $\widehat{\mathbf{H}}'(D)$, $\widehat{\mathbf{H}}'_0$

και $\hat{\mathbf{H}}'_m$ είναι οι πίνακες $\hat{\mathbf{H}}(D)$, $\hat{\mathbf{H}}_0$ και $\hat{\mathbf{H}}_m$ αντίστοιχα, μετά την εφαρμογή της μετάθεσης (3.25). Υποθέστε ότι $A \cap B \neq \emptyset$. Τότε, από την (3.25) και τους ορισμούς των συνόλων A και B , προκύπτει ότι το πρώτο μη μηδενικό στοιχείο του $\hat{\mathbf{H}}'_0$ βρίσκεται στη θέση $L(\hat{\mathbf{H}}'_0) = |A| - |A \cap B|$. Επίσης, το τελευταίο μη μηδενικό στοιχείο του $\hat{\mathbf{H}}'_m$ βρίσκεται στη θέση $R(\hat{\mathbf{H}}'_m) = |A| - 1$. Αντίστοιχα, εάν $A \cap B = \emptyset$ ισχύει ότι $L(\hat{\mathbf{H}}'_0) = n - |B|$ και $R(\hat{\mathbf{H}}'_m) = |A| - 1$ (ισχύει $|I| = n$). Έτσι, έχουμε ότι

$$\chi_{opt} = \begin{cases} 1 - |A \cap B|, & \text{if } A \cap B \neq \emptyset \\ n - |A| - |B| + 1, & \text{if } A \cap B = \emptyset \end{cases}. \quad (3.32)$$

Θεώρημα 3.9. *Το ελάχιστο trellis module του \mathcal{C} είναι βέλτιστο, αν και μόνο αν*

$$h_0^{(i)} = h_m^{(i)} = 1, \quad \text{για όλα τα } j \leq i \leq n - 1 \quad (3.33)$$

Απόδειξη. Από τις (3.30) και (3.31), η τιμή του χ που αντιστοιχεί στον πίνακα γεννήτορα $\hat{\mathbf{H}}(D)$ του $\hat{\mathcal{C}}^\perp$ (την οποία θα συμβολίζουμε με $\tilde{\chi}$) είναι

$$\tilde{\chi} = j - n + 1. \quad (3.34)$$

Το ελάχιστο trellis module του \mathcal{C} είναι βέλτιστο (δηλαδή η αρχική, η εκ κατασκευής, μετάθεση του κώδικα είναι βέλτιστη), αν και μόνο αν $\tilde{\chi} = \chi_{opt}$.

Διακρίνουμε δύο περιπτώσεις:

- $A \cap B \neq \emptyset$: Σύμφωνα με τις (3.32) και (3.34), $\tilde{\chi} = \chi_{opt}$, αν και μόνο αν

$$j - n + 1 = 1 - |A \cap B| \Leftrightarrow |A \cap B| = n - j. \quad (3.35)$$

Όπως φαίνεται στην (3.30), τα j πρώτα στοιχεία του \hat{H}_0 είναι μηδενικά. Έτσι, η (3.35) ισχύει αν και μόνο αν $h_0^i = h_m^i = 1$, για όλα τα $j \leq i \leq n - 1$.

- $A \cap B = \emptyset$: Εφόσον η τομή των A και B είναι το κενό σύνολο, τότε $|A| + |B| \leq n$. Έτσι, από την (3.32), $\chi_{opt} = n - |A| - |B| + 1 \geq n - n + 1 = 1 > j - n + 1 = \tilde{\chi}$ (υπενθυμίζουμε ότι $0 < j < n - 2$). Με λίγα λόγια, σ' αυτήν την περίπτωση υπάρχει μετάθεση η οποία παράγει έναν κώδικα ισοδύναμο με τον \mathcal{C} , του οποίου το ελάχιστο trellis module όμως έχει μικρότερες τιμές όσον αφορά στα TC , $|\Sigma|$ και E .

□

Σ' αυτήν την ενότητα, προέκυψαν οι ικανές και αναγκαίες συνθήκες που πρέπει να ικανοποιούν οι γεννήτορες ενός $(n, n - 1)$ κώδικα \mathcal{C} της προτεινόμενης οικογένειας, προκειμένου το ελάχιστο trellis module του να είναι βέλτιστο. Πιο συγκεκριμένα, το ελάχιστο trellis module του \mathcal{C} είναι βέλτιστο, αν και μόνο αν οι ορίζουσες των $(n - 1) \times (n - 1)$ υποπινάκων που προκύπτουν

από τους πίνακες \mathbf{G}_0 και \mathbf{G}_{end} , διαγράφοντας τη στήλη i , είναι ίσες με τη μονάδα, για όλα τα $j \leq i \leq n-1$. Τα αποτελέσματα αυτά αποτελούν γενίκευση της ανάλυσης που παρουσιάστηκε στο [85].

Οι συνθήκες αυτές μπορούν να φανούν χρήσιμες στην αναζήτηση νέων κωδίκων. Για παράδειγμα, έστω η περίπτωση όπου τα m και n είναι αρκετά μεγάλα. Τότε η εξαντλητική αναζήτηση του καλύτερου $(n, n-1)$ κώδικα \mathcal{C} είναι αδύνατη. Λαμβάνοντας υπόψιν μας και την επικρατούσα άποψη στη βιβλιογραφία, η οποία λέει ότι όσο αυξάνεται η trellis πολυπλοκότητα, καθώς και τα υπόλοιπα μέτρα πολυπλοκότητας του ελάχιστου trellis module, αυξάνεται και η απόδοση των κωδίκων που δύναται να κατασκευάσουμε, μπορούμε να περιορίσουμε το χώρο αναζήτησης κωδίκων, εξετάζοντας μόνο αυτούς που ικανοποιούν τις προαναφερθείσες συνθήκες. Αυτό γιατί οι υπόλοιποι κώδικες μπορούν να περιγραφούν από λιγότερο πολύπλοκα trellis modules (για την ακρίβεια υπάρχουν ισοδύναμοι με αυτούς κώδικες που μπορούν να περιγραφούν από λιγότερο πολύπλοκα trellis modules). Με λίγα λόγια, χρησιμοποιούμε τις παραπάνω συνθήκες για να εντοπίσουμε τους κώδικες που δεν έχουν αντίστοιχες ισοδύναμες μορφές με απλούστερα trellis modules.

3.7 Άνω Φράγματα για την Ελεύθερη Απόσταση των Νέων Κωδίκων

Όπως αναφέραμε στην ενότητα 3.1, υπάρχουν διάφοροι τρόποι (όταν $k < n-1$) να κατανεμηθούν τα n κωδικά bit στα k τμήματα του trellis module \mathcal{M} ενός (n, k, m) συγκεραστικού κώδικα \mathcal{C} της προτεινόμενης κατηγορίας. Σε αυτό το κεφάλαιο, καθώς και στην αναζήτηση από την οποία προέκυψαν οι Πίνακες 3.1, 3.2 και 3.3, χρησιμοποιήσαμε για κάθε (n, k, m) κώδικα την κατανομή που προέκυψε (κατόπιν εξαντλητικής αναζήτησης στις περιπτώσεις που αυτό ήταν εφικτό) από τη μελέτη στο [15], και οδήγησε στον βέλτιστο (n, k, m) PCC. Σε κάποιες περιπτώσεις όμως, είναι πιθανό μια διαφορετική κατανομή των κωδικών bit να οδηγήσει σε καλύτερο κώδικα. Εάν λάβει κανείς υπόψιν του όλες τις δυνατές κατανομές κωδικών bit που οδηγούν σ' έναν (n, k, m) συγκεραστικό κώδικα \mathcal{C} συγκεκριμένης trellis πολυπλοκότητας TC , τότε πιθανότατα να αυξηθεί υπερβολικά το μέγεθος του χώρου αναζήτησης, το σύνολο δηλαδή των κωδίκων που πρέπει να εξεταστούν προκειμένου να βρεθεί ο καλύτερος. Σ' αυτή την ενότητα θα αναπτύξουμε ένα άνω φράγμα για την ελάχιστη απόσταση του \mathcal{C} , βασιζόμενοι στο φράγμα του Heller. Το εν λόγω φράγμα θα εξαρτάται από την κατανομή των κωδικών bit στα τμήματα trellis, επομένως μπορούμε να γνωρίζουμε εξ αρχής ποιες κατανομές κωδικών bit έχουν πιθανότητα να οδηγήσουν σε κώδικες μεγάλης ελάχιστης απόστασης και ποιες όχι.

Θεωρήστε έναν (n, k, m) κώδικα \mathcal{C} της προτεινόμενης οικογένειας, και το ελάχιστο trellis module \mathcal{M}_{min} αυτού, το οποίο κατασκευάζεται στην ενότητα 3.3. Έστω \hat{n}_i και \hat{b}_i το πλήθος των κωδικών bit και η διάσταση του χώρου ακμών αντίστοιχα του τμήματος trellis \hat{l} του \mathcal{M}_{min} , όπου $0 \leq \hat{l} \leq \hat{\mu} - 1$ και $k < \hat{\mu} \leq n$ το πλήθος των τμημάτων trellis που απαρτίζουν το \mathcal{M}_{min} . Έστω επίσης \hat{s}_i η διάσταση του χώρου καταστάσεων του \mathcal{M}_{min} σε βάθος \hat{l} . Σύμφωνα με όσα

συζητήθηκαν στην ενότητα 3.3, είναι προφανές πως τα \hat{n}_i , \hat{b}_i και \hat{s}_i προκύπτουν άμεσα από τη διάσταση του χώρου καταστάσεων του αρχικού trellis module \mathcal{M} σε κάθε βάθος l , καθώς και από την κατανομή των κωδικών του bit.

Θεώρημα 3.10. Η ελεύθερη απόσταση d_f του \mathcal{C} ικανοποιεί τη σχέση

$$d_f \leq \min_{\substack{0 \leq \hat{l} \leq \hat{\mu}-1 \\ i \geq \theta}} \left\{ \frac{\sum_{j=0}^{i-1} \hat{n}_{j+\hat{l} \bmod \hat{\mu}}}{2(1 - 2^{\hat{s}_{i+\hat{l} \bmod \hat{\mu}} - \sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}})})} \right\} \quad (3.36)$$

όπου θ είναι ο μικρότερος ακέραιος για τον οποίο ισχύει

$$\sum_{j=0}^{\theta-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}}) > \hat{s}_{\theta+\hat{l} \bmod \hat{\mu}}. \quad (3.37)$$

Απόδειξη. Θεωρούμε τα κωδικά μονοπάτια τα οποία ξεκινούν από την κατάσταση 0 σε βάθος \hat{l} του ελάχιστου trellis. Σε βάθος $i + \hat{l}$ το πλήθος τους είναι ίσο με $2^{\sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}})}$. Αυτό ισχύει γιατί το πλήθος των ακμών του τμήματος p του \mathcal{M}_{min} που αναχωρούν από μία συγκεκριμένη κατάσταση σε βάθος p είναι ίσο με $2^{\hat{b}_p - \hat{s}_p}$. Έτσι, το πλήθος των μονοπατιών που καταλήγουν σε όλες τις καταστάσεις σε βάθος $i + \hat{l}$ του διαγράμματος trellis ισούται με το πλήθος των μονοπατιών που καταλήγουν σε όλες τις καταστάσεις σε βάθος $i + \hat{l} - 1$ επί τον συντελεστή $2^{\hat{b}_{i+\hat{l}-1 \bmod \hat{\mu}} - \hat{s}_{i+\hat{l}-1 \bmod \hat{\mu}}}$. Ισοδύναμα, $\hat{b}_p - \hat{s}_p$ είναι το σύνολο των bit πληροφορίας που αντιστοιχούν στο τμήμα trellis p του \mathcal{M}_{min} . Επομένως, οι ακολουθίες πληροφορίας που αντιστοιχούν στα μονοπάτια που καταλήγουν σε όλες τις καταστάσεις σε βάθος $i + \hat{l}$ του διαγράμματος trellis απαρτίζονται από $\sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}})$ bit και συνεπώς το πλήθος τους είναι ίσο με $2^{\sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l}-1 \bmod \hat{\mu}} - \hat{s}_{j+\hat{l}-1 \bmod \hat{\mu}})}$. Ο αριθμός των μονοπατιών που καταλήγουν στην κατάσταση 0 σε βάθος $i + \hat{l}$ είναι ίσος με $2^{\sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}}) - \hat{s}_{i+\hat{l} \bmod \hat{\mu}}}$. Κάθε μονοπάτι αντιστοιχεί σε μία κωδική λέξη, η οποία απαρτίζεται από $\sum_{l=0}^{i-1} \hat{n}_{j+\hat{l} \bmod \hat{\mu}}$ κωδικά bit. Το σύνολο των κωδικών λέξεων αποτελεί έναν $(\sum_{l=0}^{i-1} \hat{n}_{j+\hat{l} \bmod \hat{\mu}}, \sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}}) - \hat{s}_{i+\hat{l} \bmod \hat{\mu}})$ γραμμικό μπλοκ κώδικα, η ελάχιστη απόσταση του οποίου φράσσεται άνω από το φράγμα Plotkin στην (2.25). Η σχέση (3.37) προσδιορίζει την ελάχιστη τιμή του i (την οποία συμβολίζουμε με θ), για την οποία ο αριθμός των μονοπατιών που καταλήγουν στην κατάσταση 0 σε βάθος $\theta + \hat{l}$ είναι μεγαλύτερος του 1. \square

Αξίζει να σημειωθεί πως εάν στην απόδειξη του Θεωρήματος (3.37) φράξουμε την ελάχιστη απόσταση του $(\sum_{l=0}^{i-1} \hat{n}_{j+\hat{l} \bmod \hat{\mu}}, \sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}}) - \hat{s}_{i+\hat{l} \bmod \hat{\mu}})$ γραμμικού μπλοκ κώδικα χρησιμοποιώντας το φράγμα Griesmer στην (2.26), τότε ενδεχομένως να παρατηρήσουμε μικρή βελτίωση σε κάποιες περιπτώσεις. Κατ' αυτόν τον τρόπο, προκύπτει η σχέση

$$\sum_{j=0}^{i-1} (\hat{b}_{j+\hat{l} \bmod \hat{\mu}} - \hat{s}_{j+\hat{l} \bmod \hat{\mu}}) - \hat{s}_{i+\hat{l} \bmod \hat{\mu}}^{-1} \left[\frac{d_f}{2^w} \right] \leq \sum_{l=0}^{i-1} \hat{n}_{j+\hat{l} \bmod \hat{\mu}} \quad (3.38)$$

την οποία ικανοποιεί η ελεύθερη απόσταση d_f του \mathcal{C} , για κάθε $0 \leq \hat{l} \leq \hat{\mu} - 1$ και $i \geq \theta$.

Για κάθε (n, k, m) κώδικα στους Πίνακες 3.1, 3.2 και 3.3, όπου $k < n - 1$, χρησιμοποιούμε τα παραπάνω φράγματα, για να διαπιστώσουμε εάν υπάρχει κατανομή των κωδικών bit στα τμήματα trellis του αρχικού trellis module, η οποία θα μπορούσε να οδηγήσει σε κώδικα με την ίδια ελάχιστη trellis πολυπλοκότητα και μεγαλύτερη ελεύθερη απόσταση. Έτσι, προκύπτει ότι για τους κώδικες $(5, 2, 8)$, $(5, 3, 8)$ του Πίνακα 3.1, $(7, 4, 5)$ του Πίνακα 3.2 και $(7, 5, 6)$, $(7, 5, 7)$, $(7, 5, 8)$ του Πίνακα 3.3, υπάρχουν κατανομές που θα μπορούσαν να οδηγήσουν σε αντίστοιχους κώδικες με μεγαλύτερη ελεύθερη απόσταση. Στην περίπτωση του $(7, 4, 5)$ βρήκαμε έναν κώδικα με $d_f = 8$, ενώ ο αντίστοιχος κώδικας στον Πίνακα 3.2 έχει $d_f = 7$. Οι γεννίτορες ακολουθίες του καινούργιου κώδικα (στη μορφή που χρησιμοποιείται στους Πίνακες 3.1, 3.2 και 3.3) είναι οι $51, 43, \overline{(145, 173)}, \overline{(175, 156, 117)}$.

Κλείνοντας αυτήν την ενότητα, αξίζει να παρατηρήσουμε ότι στις περισσότερες περιπτώσεις, η αρχική κατανομή των κωδικών bit που χρησιμοποιήθηκε (η κατανομή δηλαδή που οδήγησε στον καλύτερο PCC στο [15]), οδήγησε σε βέλτιστους (όσον αφορά στην ελάχιστη απόσταση) κώδικες.

3.A Απόδειξη Λήμματος 3.5

Σύμφωνα με το Θεώρημα 3.4, αν ο πολυωνυμικός πίνακας \mathbf{G} δεν είναι καταστροφικός, τότε είναι κανονικός. Άρα, [25] ο \mathbf{G} είναι trellis κανονικός πίνακας γεννίτορας, αν ο \mathbf{G}_{scalar} έχει την LR -ιδιότητα [25]. Επίσης, ο \mathbf{G} είτε είναι ήδη σε trellis κανονική μορφή, ή μερικές γραμμοπράξεις είναι αρκετές για να μετασχηματιστεί σε trellis κανονικό πίνακα. Από την (3.11), ο πίνακας $\hat{\mathbf{G}}$ είναι ένα κατακόρυφο τμήμα του \mathbf{G}_{scalar} (για την ακρίβεια ο \mathbf{G}_{scalar} απαρτίζεται από μετατοπισμένα αντίγραφα του $\hat{\mathbf{G}}$), οπότε αν κάθε στήλη του $\hat{\mathbf{G}}$ περιέχει το πολύ ένα υπεργραμμισμένο κι ένα υπογραμμισμένο στοιχείο, τότε ο \mathbf{G}_{scalar} έχει την LR -ιδιότητα.

Από την (3.6), όλα τα υπογραμμισμένα στοιχεία εμφανίζονται σε διαφορετικές στήλες του \mathbf{G}_0 . Έτσι, κάθε στήλη του $\hat{\mathbf{G}}$ έχει το πολύ ένα υπογραμμισμένο στοιχείο, και \mathbf{G}_{scalar} έχει την L -ιδιότητα.

Από την απόδειξη του Θεωρήματος 3.3, προκύπτει πως οι πρώτες R_0 γραμμές του \mathbf{G}_{ν_f} είναι μηδενικές, ενώ οι πρώτες R_0 γραμμές του \mathbf{G}_{ν_f-1} και οι τελευταίες $k - R_0$ γραμμές του \mathbf{G}_{ν_f} είναι μη μηδενικές. Οπότε, τα τελευταία μη μηδενικά στοιχεία (υπεργραμμισμένα στοιχεία) των πρώτων R_0 γραμμών του $\tilde{\mathbf{G}}$ περιέχονται στον \mathbf{G}_{ν_f-1} , ενώ τα τελευταία μη μηδενικά στοιχεία των τελευταίων $k - R_0$ γραμμών του $\tilde{\mathbf{G}}$ περιέχονται στον \mathbf{G}_{ν_f} . Αξίζει να θυμηθούμε εδώ ότι ο \mathbf{G}_{end} στην (3.7) αποτελείται από τις τελευταίες $k - R_0$ γραμμές του \mathbf{G}_{ν_f} και τις πρώτες R_0 γραμμές του \mathbf{G}_{ν_f-1} . Έτσι, αν ο \mathbf{G}_{end} περιέχει το πολύ ένα υπεργραμμισμένο στοιχείο σε κάθε μία στήλη του, τότε το ίδιο ισχύει και για τον $\hat{\mathbf{G}}$, οπότε ο \mathbf{G}_{scalar} έχει την R -ιδιότητα.

Η γραμμή r του \mathbf{G}_{end} , για $0 \leq r \leq k - R_0 - 1$ είναι η γραμμή $(r + R_0)$ του \mathbf{G}_{ν_f} . Επίσης, η γραμμή r του \mathbf{G}_{end} , για $k - R_0 \leq r \leq k - 1$ είναι η γραμμή $(r - k + R_0)$ του \mathbf{G}_{ν_f-1} .

Μπορεί να επιβεβαιωθεί ότι η γραμμή r του \mathbf{G}_{end} , για $0 \leq r < k - 1$, περιέχει τους $\mathbf{G}_m(r)$ και $\mathbf{G}_{m+1}(r+1)$. Από το Λήμμα 3.2, ένας από τους δύο περιέχει το τελευταίο μη μηδενικό στοιχείο της συγκεκριμένης γραμμής, ενώ το τελευταίο μη μηδενικό στοιχείο της γραμμής $(k - 1)$ του \mathbf{G}_{end} περιέχεται στο $\mathbf{G}_m(k - 1)$.

Στη συνέχεια θεωρείστε δύο γραμμές r_1 και r_2 του \mathbf{G}_{end} , όπου $0 \leq r_1 < r_2 \leq k - 1$. Υποθέστε πως τα τελευταία μη μηδενικά στοιχεία των δύο γραμμών βρίσκονται στην ίδια στήλη του \mathbf{G}_{end} . Από τη δομή του \mathbf{G}_{end} , αυτό είναι πιθανό μόνον αν $r_2 = r_1 + 1$, $\mathbf{G}_{m+1}(r_2 + 1) = \mathbf{0}$, ενώ ο $\mathbf{G}_{m+1}(r_2)$ και ο $\mathbf{G}_m(r_2)$ είναι μη μηδενικοί. Δηλαδή, το τμήμα r_2 του αρχικού trellis module ανήκει στο σύνολο $T_{m+1,m}$. Διακρίνουμε δύο περιπτώσεις.

1. Αν $g_{m+1}^{(r_2, p_{r_2} + l)} = 0$ και $g_m^{(r_2, p_{r_2} + l)} = 1$ ή $g_{m+1}^{(r_2, p_{r_2} + l)} = 1$ και $g_m^{(r_2, p_{r_2} + l)} = 0$, για $1 \leq l \leq n_{r_2} - p_{r_2}$ τότε το τελευταίο μη μηδενικό στοιχείο της γραμμής r_1 βρίσκεται στη στήλη p_{r_2} (ή n_{r_2}) του $\mathbf{G}_{m+1}(r_2)$ (αυτό προκύπτει από την ιδιότητα P4 και τον ορισμό του p_{r_2}), ενώ το τελευταίο μη μηδενικό στοιχείο της γραμμής r_2 βρίσκεται στη στήλη n_{r_2} (ή p_{r_2}) του $\mathbf{G}_m(r_2)$. Δηλαδή, τα τελευταία μη μηδενικά στοιχεία των γραμμών r_1 και r_2 του \mathbf{G}_{end} βρίσκονται σε διαφορετικές στήλες. Οπότε, ο \mathbf{G}_{scalar} έχει την R -ιδιότητα.
2. Αν $g_{m+1}^{(r_2, p_{r_2} + l)} = g_m^{(r_2, p_{r_2} + l)} = 1$, για $1 \leq l \leq n_{r_2} - p_{r_2}$, τότε το τελευταίο μη μηδενικό στοιχείο της γραμμής r_1 βρίσκεται στη στήλη n_{r_2} του $\mathbf{G}_{m+1}(r_2)$, και το τελευταίο μη μηδενικό στοιχείο της γραμμής r_2 βρίσκεται στη στήλη n_{r_2} του $\mathbf{G}_m(r_2)$. Οπότε, και τα δύο περιέχονται στην ίδια στήλη του \mathbf{G}_{end} . Σ'αυτή την περίπτωση, πρέπει γίνουν γραμμοπράξεις στον \mathbf{G} , έτσι ώστε ο \mathbf{G}_{scalar} να αποκτήσει την R -ιδιότητα [25]. Πιο συγκεκριμένα, αν $0 \leq r_1 < r_2 \leq k - R_0 - 1$, δηλαδή, και οι δύο γραμμές περιέχονται στον \mathbf{G}_{ν_f} , τότε πρέπει να προσθέσουμε τη γραμμή $(r_2 + R_0)$ στη γραμμή $(r_1 + R_0)$ του \mathbf{G} . Αξίζει να σημειωθεί πως εφόσον $r_2 + R_0 > r_1 + R_0$, τότε αυτή η γραμμοπράξη δεν θα επηρεάσει την L -ιδιότητα. Όντως, στον \mathbf{G}_0 το πρώτο μη μηδενικό στοιχείο της γραμμής $(r_2 + R_0)$ βρίσκεται δεξιά του πρώτου μη μηδενικού στοιχείου της γραμμής $(r_1 + R_0)$. Μετά την πρόσθεση, το τελευταίο μη μηδενικό στοιχείο της γραμμής r_1 του \mathbf{G}_{end} είναι το στοιχείο p_{r_2} του $\mathbf{G}_{m+1}(r_2)$ (Αυτό προκύπτει από την ιδιότητα P4 και τον ορισμό του p_{r_2}), ενώ το τελευταίο μη μηδενικό στοιχείο της γραμμής r_2 του \mathbf{G}_{end} είναι το στοιχείο n_{r_2} του $\mathbf{G}_m(r_2)$. Εάν $k - R_0 \leq r_1 < r_2 \leq k - 1$, και οι δύο γραμμές περιέχονται στον $\mathbf{G}_{\nu_{f-1}}$, οπότε η γραμμή $(r_2 - k + R_0)$ του \mathbf{G} πρέπει να προστεθεί στη γραμμή $(r_1 - k + R_0)$. Μετά την πρόσθεση, το τελευταίο μη μηδενικό στοιχείο της γραμμής r_1 του \mathbf{G}_{end} είναι το στοιχείο p_{r_2} του $\mathbf{G}_{m+1}(r_2)$, και το τελευταίο μη μηδενικό στοιχείο της γραμμής r_2 του \mathbf{G}_{end} είναι το στοιχείο n_{r_2} του $\mathbf{G}_m(r_2)$. Τέλος, για $0 \leq r_1 \leq k - R_0 - 1$ και $k - R_0 \leq r_2 \leq k - 1$, η γραμμή r_1 του \mathbf{G}_{end} ανήκει στον \mathbf{G}_{ν_f} , ενώ η γραμμή r_2 βρίσκεται στον $\mathbf{G}_{\nu_{f-1}}$. Εφόσον $r_2 = r_1 + 1$, αυτό μπορεί να συμβεί μόνο εάν $r_1 = k - R_0 - 1$ και $r_2 = k - R_0$. Επίσης, η r_1 είναι η τελευταία γραμμή

του \mathbf{G}_{ν_f} , ενώ η r_2 είναι η πρώτη γραμμή του \mathbf{G}_{ν_f-1} . Σ' αυτήν την περίπτωση, εάν πολλαπλασιάσουμε την πρώτη γραμμή του \mathbf{G} με D , και την προσθέσουμε στην τελευταία γραμμή, τότε το τελευταίο μη μηδενικό στοιχείο της γραμμής r_1 του \mathbf{G}_{end} θα είναι το στοιχείο p_{r_2} του $\mathbf{G}_{m+1}(r_2)$, και το τελευταίο μη μηδενικό στοιχείο της γραμμής r_2 του \mathbf{G}_{end} θα είναι το στοιχείο n_{r_2} του $\mathbf{G}_m(r_2)$. Η πράξη αυτή δεν επηρεάζει την L -ιδιότητα, αφού πολλαπλασιάζουμε με D πρώτα, κι έτσι ο \mathbf{G}_0 δεν αλλάζει. Αξίζει να σημειωθεί πως μετά τις συγκεκριμένες γραμμοπράξεις, ο \mathbf{G} παραμένει κανονικός.

Από τη σχέση (3.8), προκύπτει πως η γραμμή r του $\tilde{\mathbf{G}}$ δίνεται από,

$$\left[\mathbf{G}_0(r) \quad \mathbf{G}_1(r+1 \bmod k) \quad \dots \quad \mathbf{G}_m(r+m \bmod k) \quad \mathbf{G}_{m+1}(r+m+1 \bmod k) \right] \quad (3.39)$$

για $0 \leq r \leq k-1$. Τα μηδενικά (και μη ενεργά) στοιχεία πριν από τον $\mathbf{G}_0(r)$ και μετά τον $\mathbf{G}_{m+1}(r+m+1 \bmod k)$ παραλείπονται. Είναι προφανές πως για όλα τα τμήματα l , $0 \leq l \leq k-1$, του trellis module, οι πίνακες $\mathbf{G}_j(l)$, $0 \leq j < m$ περιέχουν μόνο ενεργά στοιχεία (λόγω της ιδιότητας P_1 , ο $\mathbf{G}_0(l)$ περιέχει μόνο μονάδες), εφόσον περιέχονται στο ενεργό διάστημα κάποιας γραμμής του $\tilde{\mathbf{G}}$.

Για κάθε τμήμα $l \in T_{m,m}$, $0 \leq l \leq k-1$, ο $\mathbf{G}_m(l)$ περιέχει μόνο ενεργά στοιχεία (λόγω της ιδιότητας P_2 ο $\mathbf{G}_m(l)$ περιέχει μόνο μονάδες). Επίσης, από την (3.39) συμπεραίνουμε πως εφόσον $\mathbf{G}_{m+1}(l) = \mathbf{0}$, τότε ο $\mathbf{G}_{m+1}(l)$ δεν περιέχεται στο ενεργό διάστημα κάποιας γραμμής του $\tilde{\mathbf{G}}$.

Για κάθε τμήμα $l \in T_{m,m+1}$, $0 \leq l < k-1$, ο $\mathbf{G}_m(l)$ περιέχει μόνο ενεργά στοιχεία, αφού στην ίδια γραμμή του $\tilde{\mathbf{G}}$ μετά τον $\mathbf{G}_m(l)$ ακολουθεί ο $\mathbf{G}_{m+1}(l+1)$ ο οποίος είναι μη μηδενικός. Επίσης, εφόσον $\mathbf{G}_{m+1}(l) = \mathbf{0}$, συμπεραίνουμε, όπως στην προηγούμενη περίπτωση, ότι ο \mathbf{G}_{m+1} δεν περιέχει ενεργά στοιχεία.

Για κάθε τμήμα $l \in T_{m+1,m+1}$, $1 \leq l < k-1$, ο $\mathbf{G}_m(l)$ περιέχει μόνο ενεργά στοιχεία, εφόσον στην ίδια γραμμή του $\tilde{\mathbf{G}}$ μετά τον $\mathbf{G}_m(l)$ ακολουθεί ο $\mathbf{G}_{m+1}(l+1)$, ο οποίος είναι μη μηδενικός. Επιπλέον, ο $\mathbf{G}_{m+1}(l)$ περιέχει μόνο ενεργά στοιχεία, αφού από την ιδιότητα P_3 περιέχει μόνο μονάδες.

Για κάθε τμήμα $l \in T_{m+1,m}$, $1 \leq l \leq k-1$, όταν ο \mathbf{G} είναι trellis κανονικός πίνακας, τα πρώτα p_l στοιχεία ενός εκ των πινάκων $\mathbf{G}_m(l)$ και $\mathbf{G}_{m+1}(l)$ και όλα τα n_l στοιχεία του άλλου πίνακα είναι ενεργά.

Συνοψίζοντας

- για κάθε τμήμα $l \notin T_{m+1,m}$ του trellis module, όλοι οι πίνακες $\mathbf{G}_j(l)$, για $0 \leq j \leq m_l$ περιέχουν μόνο ενεργά στοιχεία. Οι πίνακες $\mathbf{G}_j(l)$, για $j > m_l$ δεν περιέχουν ενεργά στοιχεία.
- για κάθε τμήμα $l \in T_{m+1,m}$ του trellis module, οι πίνακες $\mathbf{G}_j(l)$, για $0 \leq j \leq m-1$, περιέχουν μόνο ενεργά στοιχεία. Επίσης, τα πρώτα p_l στοιχεία ενός εκ των πινάκων $\mathbf{G}_m(l)$

και $\mathbf{G}_{m+1}(l)$ και όλα τα n_l στοιχεία του άλλου πίνακα είναι ενεργά. Οι πίνακες $\mathbf{G}_j(l)$, για $j > m + 1$ δεν περιέχουν ενεργά στοιχεία.

4. ΕΥΕΛΙΚΤΟΙ ΣΥΓΚΕΡΑΣΤΙΚΟΙ ΚΩΔΙΚΕΣ: ΜΕΤΑΒΛΗΤΟΣ ΡΥΘΜΟΣ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ

Στις σύγχρονες ασύρματες επικοινωνίες, οι απαιτήσεις από ένα σχήμα κωδικοποίησης είναι ιδιαίτερα αυξημένες. Καταρχάς, οι συνθήκες του καναλιού μεταβάλλονται με το χρόνο, κι έτσι απαιτούνται διαφορετικά επίπεδα προστασίας στην πληροφορία, και ταυτόχρονα σωστή διαχείριση των πόρων του συστήματος. Επίσης, στα σύγχρονα τηλεπικοινωνιακά δίκτυα, τα είδη της πληροφορίας που μεταδίδονται ποικίλουν (π.χ. φωνή, πακέτα δεδομένων, κτλ.). Κάθε είδος πληροφορίας έχει διαφορετική ανοχή στο θόρυβο (ή γενικότερα στις αλλοιώσεις που εισάγει το κανάλι), με αποτέλεσμα να χρειάζεται και διαφορετικό επίπεδο προστασίας. Μία λύση θα ήταν να χρησιμοποιηθούν διαφορετικοί κωδικοποιητές (και αντίστοιχα αποκωδικοποιητές) για διαφορετικές περιπτώσεις. Όμως, αν αναλογιστεί κανείς το γεγονός ότι οι πομποδέκτες είναι συνήθως μικρά τεμαχικά (π.χ. κινητά τηλέφωνα), υπάρχουν σοβαροί περιορισμοί στο μέγεθος της συσκευής. Έτσι λοιπόν, η ιδανική λύση είναι η κατασκευή ενός και μοναδικού κωδικοποιητή (και αποκωδικοποιητή αντίστοιχα), του οποίου κάποια παράμετρος (π.χ. ο ρυθμός κωδικοποίησης) μπορεί να μεταβάλλεται, έτσι ώστε να προσαρμόζεται η διορθωτική του ικανότητα (το επίπεδο προστασίας που παρέχεται στην πληροφορία) στις αντίστοιχες απαιτήσεις [86]. Όλα αυτά συμβαίνουν προφανώς, χρησιμοποιώντας το ίδιο κύκλωμα κωδικοποιητή και αποκωδικοποιητή στον πομπό και τον δέκτη αντίστοιχα.

Οι πιο δημοφιλείς κώδικες για την εφαρμογή αυτής της τεχνικής είναι οι *διάτρητοι συγκεραστικοί κώδικες μεταβλητού ρυθμού* (variable rate punctured convolutional codes VRPCC) [11, 12, 16, 17]. Το παραπάνω κωδικό σχήμα απαρτίζεται από έναν $(n, 1, m)$ *μπτρικό* συγκεραστικό κώδικα και ένα σύνολο πινάκων απαλοιφής κωδικών bit. Κάθε πίνακας απαλοιφής παράγει έναν κώδικα συγκεκριμένου ρυθμού. Επί της ουσίας πρόκειται για μία οικογένεια από PCC που παράγονται από τον ίδιο μπτρικό κώδικα. Όλοι οι παραγόμενοι κώδικες αποκωδικοποιούνται χρησιμοποιώντας το διάγραμμα trellis (άρα και τον αποκωδικοποιητή) του μπτρικού κώδικα. Οι συγκεραστικοί κώδικες μεταβλητού ρυθμού μπορούν να προσφέρουν πολλαπλά επίπεδα προστασίας μεταβάλλοντας το ρυθμό κωδικοποίησης. Έτσι, όταν δεν απαιτείται υψηλή προστασία από τον κώδικα, στέλνονται λιγότερα κωδικά bit στο κανάλι, ενώ στην αντίθετη περίπτωση στέλνονται περισσότερα. Επίσης, για κάθε εφαρμογή μπορεί να επιλέγεται ο κατάλληλος κώδικας από την οικογένεια, ανάλογα με τις απαιτήσεις. Είναι προφανές ότι με τη χρήση αυτής της τεχνικής γίνεται αποδοτικότερη εκμετάλλευση του διαθέσιμου εύρους ζώνης. Η ευελιξία που προσφέρεται από το συγκεκριμένο σχήμα κωδικοποίησης είναι πολύ χρήσιμη για τα σύγχρονα συστήματα επικοινωνιών. Για παράδειγμα, το πρότυπο IEEE 802.22 για γνωστικά ασύρματα περιφερειακά δίκτυα (cognitive wireless regional area networks (WRAN)) χρησιμοποιεί (σαν υποχρεωτικό σχήμα κωδικοποίησης) έναν $(2, 1, 6)$ μπτρικό συγκεραστικό κώδικα, μαζί με ένα σύνολο πινάκων απαλοιφής κωδικών bit, που οδηγούν σε κωδικούς ρυθμούς $2/3$,

3/4 και 5/6 [18].

Τελευταία έχουν αναπτυχθεί κι άλλες τεχνικές για την κατασκευή συγκεραστικών κωδικοποιητών μεταβλητού ρυθμού, όπως η τεχνική της απαλοιφής μονοπατιών του διαγράμματος trellis (path pruning) [20, 87, 21], όπου καθορισμένα bit χρησιμοποιούνται για την διαγραφή συγκεκριμένων μονοπατιών από το διάγραμμα trellis ενός (n, k, m) μητρικού συγκεραστικού κώδικα. Στο [22], η τεχνική της απαλοιφής μονοπατιών του διαγράμματος trellis και η μέθοδος απαλοιφής κωδικών bit συνδυάζονται προκειμένου να σχεδιαστούν οικογένειες συγκεραστικών κωδικών μεταβλητού ρυθμού, οι οποίες μπορούν να υποστηρίξουν περισσότερους κωδικούς ρυθμούς.

Στο παρόν κεφάλαιο συνδυάζουμε τις τεχνικές της απαλοιφής μονοπατιών του διαγράμματος trellis και της απαλοιφής κωδικών bit, προκειμένου να κατασκευάσουμε συγκεραστικούς κώδικες που εκτός από το ρυθμό, μπορούν να μεταβάλλουν και την υπολογιστική πολυπλοκότητα της διαδικασίας αποκωδικοποίησης [88]. Πιο συγκεκριμένα, από έναν $(n, 1, m)$ μητρικό συγκεραστικό κώδικα παράγουμε μία μεγάλη οικογένεια με κώδικες πολλαπλών ρυθμών. Για κάθε ρυθμό η οικογένεια περιέχει trellis modules με διάφορες τιμές της υπολογιστικής πολυπλοκότητας αποκωδικοποίησης. Όλα τα trellis modules προκύπτουν από το trellis module του μητρικού κώδικα, συνεπώς όλοι οι κώδικες της οικογένειας κωδικοποιούνται και αποκωδικοποιούνται χρησιμοποιώντας τον κωδικοποιητή και τον αποκωδικοποιητή του μητρικού κώδικα. Συγκριτικά με τους κλασσικούς συγκεραστικούς κώδικες μεταβλητού ρυθμού, οι οποίοι “ανταλλάσσουν” κέρδος κωδικοποίησης με εύρος ζώνης και αντίστροφα, οι συγκεκριμένες κατασκευές προσθέτουν ακόμα μία “διάσταση”, την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης, οδηγώντας σε σχήματα κωδικοποίησης που διαχειρίζονται με πιο αποδοτικό τρόπο τους πόρους του συστήματος. Επίσης, αυτά τα σχήματα κωδικοποίησης μπορούν να χρησιμοποιηθούν από συστήματα επικοινωνιών που υποστηρίζουν διάφορες εφαρμογές. Για κάθε εφαρμογή μπορεί να χρησιμοποιείται ο κατάλληλος κώδικας (όσον αφορά στο ρυθμό, στο κέρδος κωδικοποίησης, στην πολυπλοκότητα, κτλ.) από την οικογένεια.

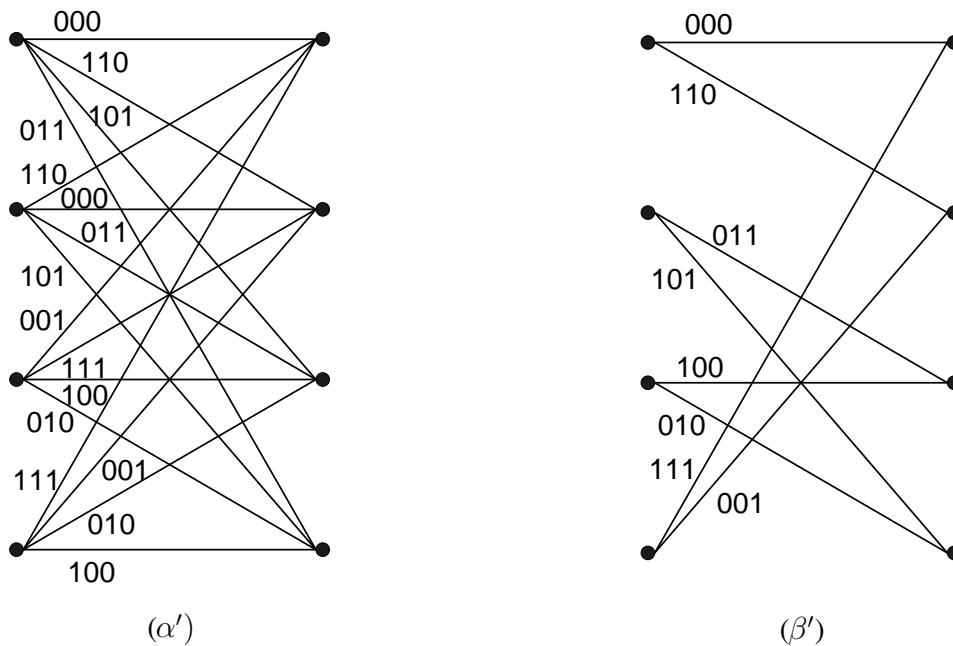
4.1 Η Τεχνική της Απαλοιφής Μονοπατιών του Διαγράμματος Trellis

Η γενική ιδέα της απαλοιφής μονοπατιών του διαγράμματος trellis παρουσιάζεται στα [20], [21]. Η διαγραφή μονοπατιών είναι αποτέλεσμα της εξαίρεσης συγκεκριμένων ακολουθιών bit πληροφορίας και επιτυγχάνεται εισάγοντας στην είσοδο του κωδικοποιητή, εκτός από τα “τυχαία” bit πληροφορίας, και κάποια καθορισμένα bit.

Παράδειγμα 4.1. Θεωρείστε για παράδειγμα τον $(3, 2, 2)$ συγκεραστικό κώδικα \mathcal{C} με πίνακα γεννήτορα

$$\mathbf{G}(D) = \begin{bmatrix} D + 1 & D & 1 \\ 1 & 1 & D \end{bmatrix}. \quad (4.1)$$

Το (συμβατικό) trellis module του \mathcal{C} φαίνεται στο Σχήμα 4.1(α'). Ο κωδικοποιητής που παράγει τον \mathcal{C} έχει δύο εισόδους, γι'αυτό και τα bit πληροφορίας χωρίζονται σε δύο ακολουθίες. Έστω ότι $u_t^{(1)}$ και $u_t^{(2)}$ είναι τα bit πληροφορίας που αντιστοιχούν στην πρώτη και τη δεύτερη ακολουθία αντίστοιχα, ενώ $\hat{u}_t^{(1)}$ και $\hat{u}_t^{(2)}$ είναι τα bit που εισέρχονται στην πρώτη και τη δεύτερη είσοδο του κωδικοποιητή αντίστοιχα, τη χρονική στιγμή t . Στην περίπτωση του \mathcal{C} ισχύει $u_t^{(1)} = \hat{u}_t^{(1)}$ και $u_t^{(2)} = \hat{u}_t^{(2)}$. Υποθέστε τώρα ότι ο κωδικοποιητής τροφοδοτείται από μία μόνο ενεργή ακολουθία πληροφορίας, δηλαδή μόνο ένα bit πληροφορίας εισάγεται στον κωδικοποιητή, τη χρονική στιγμή t . Πιο συγκεκριμένα, θεωρήστε ότι $\hat{u}_t^{(1)} = S_t^{(1)} + S_t^{(2)} + u_t^{(1)}$ και $\hat{u}_t^{(2)} = S_t^{(1)} + S_t^{(2)}$, όπου $S_t^{(1)}$ και $S_t^{(2)}$ είναι το περιεχόμενο του πρώτου και του δεύτερου καταχωρητή μήκους 1 αντίστοιχα, τη χρονική στιγμή t . Εφόσον τα bit πληροφορίας που εισάγονται ταυτόχρονα στον κωδικοποιητή κάθε χρονική στιγμή δεν είναι πλέον δύο αλλά ένα, μόνο δύο από τις τέσσερις μεταβάσεις (ακμές) που ξεκινούν από κάθε κατάσταση στο αρχικό trellis module μπορούν να επιλεγούν. Δηλαδή, διάφορα μονοπάτια του διαγράμματος trellis θα αποκοπούν. Έτσι, στην ουσία παράγεται ένας καινούργιος $(3, 1)$ κώδικας, ο \mathcal{C}_{pr} , διαγράφοντας κωδικές λέξεις από τον \mathcal{C} . Το trellis module του κώδικα \mathcal{C}_{pr} , που προκύπτει από το trellis module του \mathcal{C} διαγράφοντας τις κατάλληλες ακμές, φαίνεται στο Σχήμα 4.1(β').



Σχήμα 4.1. Τα trellis modules των κωδίκων (α') \mathcal{C} και (β') \mathcal{C}_{pr} , του Παραδείγματος 4.1.

Έστω (n, k) συγκεραστικός κώδικας \mathcal{C} , και έστω ότι ο κωδικοποιητής του τροφοδοτείται με $\hat{k} < k$ ακολουθίες πληροφορίας. Έστω $\mathbf{U}^{(i)}(D)$ το ανάπτυγμα της σειράς Laurent της ακολουθίας πληροφορίας $\{u_t^{(i)}\}$, όπου $1 \leq i \leq \hat{k}$. Θεωρήστε επίσης ότι $\hat{\mathbf{U}}^{(1)}(D), \hat{\mathbf{U}}^{(2)}(D), \dots, \hat{\mathbf{U}}^{(k)}(D)$ είναι τα αναπτύγματα των σειρών Laurent των ακολουθιών με τις οποίες τροφοδοτούνται τελικά οι k εισοδοί του κωδικοποιητή. Η διαδικασία της απαλοιφής μονοπατιών του διαγράμματος

trellis περιγράφεται από τη συνάρτηση $\Psi(\cdot)$

$$(\hat{\mathbf{U}}^{(1)}(D), \hat{\mathbf{U}}^{(2)}(D), \dots, \hat{\mathbf{U}}^{(k)}(D)) = \Psi(\mathbf{U}^{(1)}(D), \dots, (\mathbf{U}^{(k)}(D))). \quad (4.2)$$

Προκειμένου να εξασφαλιστεί η γραμμικότητα του κώδικα που προκύπτει από τη διαδικασία απαλοιφής μονοπατιών, η συνάρτηση $\Psi(\cdot)$ πρέπει να είναι γραμμική απεικόνιση από το $F_2^{\hat{k}}(D)$ (δηλαδή το χώρο των διανυσμάτων μήκους \hat{k} των οποίων οι συνιστώσες παίρνουν τιμές στο $F_2(D)$) στο $F_2^k(D)$. Έτσι, η σχέση γράφεται ως

$$[\hat{\mathbf{U}}^{(1)}(D), \hat{\mathbf{U}}^{(2)}(D), \dots, \hat{\mathbf{U}}^{(k)}(D)] = [\mathbf{U}^{(1)}(D), \dots, \mathbf{U}^{(k)}(D)]\Theta(D) \quad (4.3)$$

όπου ο $\Theta(D)$, ο οποίος καλείται *πίνακας εντοπισμού μονοπατιών* (path-locating matrix), είναι $\hat{k} \times k$ πίνακας στο $F_2(D)$, βαθμού (rank) \hat{k} . Χωρίς βλάβη της γενικότητας μπορούμε να θεωρήσουμε ότι ο $\Theta(D)$ είναι πολυωνυμικός πίνακας, δηλαδή ότι τα στοιχεία του ανήκουν στον πολυωνυμικό δακτύλιο $F_2[D]$. Οι κωδικές λέξεις του αρχικού (μπτρικού) κώδικα οι οποίες επιλέγονται για να αποτελέσουν τον καινούργιο κώδικα προκύπτουν από τη σχέση

$$[\mathbf{U}^{(1)}(D), \dots, (\mathbf{U}^{(k)}(D)]\Theta(D)\mathbf{G}(D) \quad (4.4)$$

όπου $\mathbf{G}(D)$ είναι ο πίνακας γεννήτορας του αρχικού κώδικα. Ο προκύπτων κώδικας είναι ένας \hat{k} -διάστατος υποκώδικας του αρχικού (μπτρικού) κώδικα, με πίνακα γεννήτορα

$$\mathbf{G}_{pr}(D) = \Theta(D)\mathbf{G}(D). \quad (4.5)$$

Έτσι, η απαλοιφή μονοπατιών στο Παράδειγμα 4.1 περιγράφεται από τον πίνακα εντοπισμού μονοπατιών $\Theta(D) = [1 + D \quad D]$, και ο πίνακας γεννήτορας του κώδικα που προκύπτει είναι ο

$$\mathbf{G}_{pr}(D) = [1 + D^2 \quad 1 \quad 1 + D + D^2].$$

Όπως προκύπτει λοιπόν από τα παραπάνω, η διαδικασία της απαλοιφής μονοπατιών του διαγράμματος trellis μπορεί να περιγραφεί από την σειριακή αλύσωση (serial concatenation) ενός πολυωνυμικού πίνακα $\Theta(D)$ με τον πίνακα γεννήτορα του μπτρικού κώδικα. Αξίζει να σημειωθεί εδώ πως ο προκύπτων κώδικας μπορεί να αποκωδικοποιηθεί χρησιμοποιώντας το διάγραμμα trellis του μπτρικού κώδικα μόνο αν ικανοποιείται ένας απλός περιορισμός: “Η πληροφορία που χρειάζεται για την αποκωδικοποίηση του προκύπτοντος κώδικα δεν πρέπει να υπερβαίνει την πληροφορία που μπορεί να προσφέρει ο αποκωδικοποιητής του μπτρικού κώδικα” [21]. Για παράδειγμα, εάν για τον πίνακα γεννήτορα του μπτρικού κώδικα ισχύει ότι $\nu_1 = 3$, και κατά τη διαδικασία απαλοιφής μονοπατιών χρησιμοποιείται το bit $\hat{u}_{t-4}^{(1)}$, τότε ο κώδικας που προκύπτει δεν μπορεί να αποκωδικοποιηθεί χρησιμοποιώντας τον αποκωδικοποιητή, και κατ’ επέκτασιν

το διάγραμμα trellis του μητρικού κώδικα.

Εφόσον κατά τη διαδικασία της απαλοιφής μονοπατιών του διαγράμματος trellis αποκόπτονται κωδικά μονοπάτια, και συνεπώς αφαιρούνται κωδικές λέξεις, ο κωδικός ρυθμός μειώνεται (το k μειώνεται σε \hat{k}). Το εύρος των ρυθμών που μπορούν να επιτευχθούν με τη συγκεκριμένη μέθοδο κυμαίνονται από $(k-1)/n$ σε $(1/n)$. Περισσότεροι ρυθμοί μπορούν να επιτευχθούν αν η απαλοιφή μονοπατιών εφαρμοστεί περιοδικά στο μητρικό διάγραμμα trellis [21], με περίοδο $T > 1$. Επί της ουσίας θεωρούμε ότι ο μητρικός κώδικας έχει παραμέτρους (Tn, Tk) και παράγεται από τον ισοδύναμο (με τον (n, k) κωδικοποιητή) $Tk \times Tn$ κωδικοποιητή $\mathbf{G}^{[T]}(D)$, ο οποίος προκύπτει από τη σχέση (2.53). Προφανώς, στην τελευταία σχέση θεωρούμε ότι $G_j(l) = G_j$ για κάθε j και l , καθώς ο αρχικός κώδικας δεν είναι χρονικά μεταβαλλόμενος. Κατ' αυτόν τον τρόπο, η τεχνική της απαλοιφής μονοπατιών μπορεί να εφαρμοστεί και σε $(n, 1)$ κώδικες.

4.2 Συνδυάζοντας τις Τεχνικές της Απαλοιφής κωδικών Bit και της Απαλοιφής Μονοπατιών του Διαγράμματος Trellis: Ανάλυση Πολυπλοκότητας

Η βασική ιδέα των ευέλικτων κωδικών είναι η χρήση του trellis module ενός $(n, 1, m)$ μητρικού συγκεραστικού κώδικα, για την κατασκευή ενός χρονικά μεταβαλλόμενου trellis module συγκεκριμένης υπολογιστικής πολυπλοκότητας αποκωδικοποίησης, που αντιστοιχεί σε συγκεκριαστικό κώδικα συγκεκριμένου ρυθμού. Η τεχνική της απαλοιφής μονοπατιών χρησιμοποιείται προκειμένου να αφαιρεθεί από το μητρικό διάγραμμα trellis ποσότητα μεταβάσεων (ακμών), γεγονός που οδηγεί σε ένα διάγραμμα trellis μικρότερης πολυπλοκότητας. Στη συνέχεια, εφαρμόζεται η τεχνική της απαλοιφής κωδικών bit για τη ρύθμιση του κωδικού ρυθμού. Πιο συγκεκριμένα, οι δύο προαναφερθείσες τεχνικές εφαρμόζονται όπως παρακάτω.

4.2.1 Απαλοιφή Μονοπατιών του Διαγράμματος Trellis

Η γενική ιδέα της τεχνικής απαλοιφής μονοπατιών παρουσιάστηκε και αναλύθηκε στα [20], [21]. Όπως αποδεικνύεται εκεί, η απαλοιφή μονοπατιών από το διάγραμμα trellis ενός (n, k) συγκεραστικού κώδικα \mathcal{C} μπορεί να επιτευχθεί συνδέοντας (σε σειρά) έναν κατάλληλο εντοπιστή μονοπατιών (path locator) και τον κωδικοποιητή του \mathcal{C} . Ο εντοπιστής μονοπατιών περιγράφεται από έναν $\hat{k} \times k$ ($\hat{k} < k$) πολυωνυμικό πίνακα $\Theta(D)$, που ονομάζεται *πίνακας εντοπισμού μονοπατιών*. Ο πίνακας γεννήτορας του (n, \hat{k}) νέου συγκεραστικού κώδικα \mathcal{C}_{pr} που παράγεται με αυτήν την διαδικασία προκύπτει πολλαπλασιάζοντας τον πίνακα $\Theta(D)$ με τον πίνακα γεννήτορα του αρχικού κώδικα.

Θεωρήστε έναν $(n, 1, m)$ μητρικό συγκεραστικό κώδικα, ο οποίος παράγεται από έναν πολυωνυμικό πίνακα γεννήτορα $\mathbf{G}(D)$. Υποθέστε ότι u_t είναι το bit πληροφορίας και \hat{u}_t το bit

στην είσοδο του μπτρικού κωδικοποιητή, τη χρονική στιγμή t . Η ειδική περίπτωση της απαλοιφής μονοπατιών που θα χρησιμοποιηθεί σ' αυτό το κεφάλαιο, υλοποιείται αντιστοιχώντας την πληροφορία u_t , καθώς και τα bit που βρίσκονται αποθηκευμένα στον καταχωρητή του κωδικοποιητή, στην τελική είσοδο \hat{u}_t του κωδικοποιητή.

Πιο συγκεκριμένα, κάθε T_{pr} χρονικές στιγμές το bit στην είσοδο του κωδικοποιητή δεν είναι ένα bit πληροφορίας, αλλά ένας γραμμικός συνδυασμός των bit της τρέχουσας κατάστασης

$$\mathbf{S}_t = \{\hat{u}_{t-1}, \dots, \hat{u}_{t-m}\}$$

του καταχωρητή, δηλαδή

$$\hat{u}_t = \begin{cases} u_{t_1(T_{pr}-1)+t_2} & , \text{ εάλν } t_2 \neq 0 \\ \sum_{i=1}^{\hat{d}} c_i \hat{u}_{t_1 T_{pr}-i} & , \text{ εάλν } t_2 = 0 \end{cases} \quad (4.6)$$

όπου $t_1 = \lfloor \frac{t}{T_{pr}} \rfloor$, $t_2 = t \bmod T_{pr}$, $t = 1, 2, \dots$, και \hat{d} είναι ο βαθμός του πολυωνόμου $c(X) = \sum_{i=1}^m c_i X^i$. Οι δυαδικοί συντελεστές διαλέγονται έτσι ώστε $c_i = 0$ για $0 = i \bmod T_{pr}$, δηλαδή τα bit εισόδου που δεν αποτελούν καινούρια πληροφορία προκύπτουν από το γραμμικό συνδυασμό των bit πληροφορίας (μόνο).

Ακολουθώντας την ανάλυση στο [21], ο κώδικας που προκύπτει από το συνδυασμό του μπτρικού κώδικα και της σχέσης (4.6), μπορεί να παραχθεί από τη σειριακή σύνδεση του $(T_{pr} - 1) \times T_{pr}$ πίνακα απαλοιφής μονοπατιών

$$\Theta(D) = \begin{bmatrix} 1 & 0 & \dots & 0 & c_{T_{pr}-1} + c_{2T_{pr}-1}D + c_{3T_{pr}-1}D^2 + \dots \\ 0 & 1 & \ddots & \vdots & c_{T_{pr}-2} + c_{2T_{pr}-2}D + c_{3T_{pr}-2}D^2 + \dots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & c_1 + c_{T_{pr}+1}D + c_{2T_{pr}+1}D^2 + \dots \end{bmatrix} \quad (4.7)$$

και του ισοδύναμου (με το μπτρικό) $T_{pr} \times nT_{pr}$ πίνακα γεννήτορα $\mathbf{G}^{[T_{pr}]}(D)$ [49]. Επί της ουσίας, ο $\mathbf{G}^{[T_{pr}]}(D)$, προκύπτει από τη σχέση (2.53) αν αντιμετωπίσουμε τον $(n, 1)$ μπτρικό κώδικα ως περιοδικό με περίοδο T_{pr} . Εξ' ορισμού, $c_i = 0$ για $i > m$. Επίσης, εφόσον στην (4.6) τα bit εισόδου που δεν αντιστοιχούν σε πληροφορία παράγονται μόνο από bit που βρίσκονται αποθηκευμένα στον καταχωρητή του κωδικοποιητή, τότε ο κώδικας που προκύπτει μπορεί να αποκωδικοποιηθεί χρησιμοποιώντας το trellis του μπτρικού κωδικοποιητή [21]. Επομένως, η τεχνική απαλοιφής μονοπατιών όπως χρησιμοποιείται σ' αυτό το κεφάλαιο αποτελεί υποπερίπτωση της μεθόδου απαλοιφής μονοπατιών *αποκωδικοποίησης σε μοναδικό διάγραμμα trellis* (single-trellis decoding) [21].

Η απαλοιφή μονοπατιών όπως εφαρμόζεται στο παρόν κεφάλαιο, και όπως περιγράφεται από τη σχέση (4.6), παράγει από το μπτρικό κώδικα έναν $(nT_{pr}, T_{pr} - 1)$ χρονικά μεταβαλλό-

μενο συγκεραστικό κώδικα. Το αντίστοιχο ημίαιερο διάγραμμα trellis, έπειτα από μία μικρή μεταβατική περίοδο γίνεται περιοδικό και αποτελείται από τις επαναλήψεις ενός trellis module \mathcal{M}_{pr} που απαρτίζεται από T_{pr} τμήματα trellis. Από τη σχέση (4.6) είναι προφανές πως τα bit που βρίσκονται αποθηκευμένα στον καταχωρητή του κωδικοποιητή, μπορεί να είναι είτε bit πληροφορίας, είτε bit που δεν αντιστοιχούν σε νέα πληροφορία κι έχουν προκύψει από το γραμμικό συνδυασμό παλαιότερων bit πληροφορίας.

Ορισμός 4.2. Ένα στοιχείο \hat{u}_{t-j} της κατάστασης \mathbf{S}_t καλείται *εξαρτημένο*, εάν δεν είναι bit πληροφορίας και αν μπορεί να εκφραστεί ως γραμμικός συνδυασμός των υπολοίπων στοιχείων της κατάστασης.

Λήμμα 4.3. Το στοιχείο \hat{u}_{t-j} της κατάστασης \mathbf{S}_t είναι εξαρτημένο αν και μόνο αν

1. $j = t \bmod T_{pr}$

2. $j + \hat{d} \leq m$.

Απόδειξη. Από τη σχέση (4.6), το \hat{u}_{t-j} δεν είναι bit πληροφορίας, αν και μόνο αν, $0 = t - j \bmod T_{pr}$. Υποθέστε ότι $\hat{d} > 0$, οπότε $c(X) \neq 0$. Εάν $j + \hat{d} \leq m$, τότε από την (4.6) το \hat{u}_{t-j} γράφεται ως γραμμικός συνδυασμός των bit πληροφορίας της κατάστασης

$$\hat{u}_{t-j} = \sum_{i=1}^{\hat{d}} c_i \hat{u}_{t-j-i}.$$

Επομένως, από τον Ορισμό 4.2 είναι εξαρτημένο.

Προκειμένου να ολοκληρωθεί η απόδειξη θα δείξουμε ότι όταν $j + \hat{d} > m$, τότε το \hat{u}_{t-j} δεν μπορεί να γραφεί ως γραμμικός συνδυασμός των υπολοίπων στοιχείων της κατάστασης. Έστω

$$\hat{u}_{t-j} = \sum_{i=1}^m \mu_i \hat{u}_{t-i} \quad (4.8)$$

και τουλάχιστον ένα $\mu_i \neq 0$, ενώ $\mu_j = 0$.

Εάν ισχύει ότι $\mu_i = 0$ για όλα τα bit της κατάστασης που δεν αντιστοιχούν σε πληροφορία, τότε στην (4.8) το \hat{u}_{t-j} εκφράζεται σαν άθροισμα από bit πληροφορίας. Αυτός ο γραμμικός συνδυασμός είναι μοναδικός, αφού όλα τα bit πληροφορίας είναι στην ουσία τυχαίες μεταβλητές. Ο προσθετός με το μικρότερο δείκτη που συμμετέχει στο άθροισμα της (4.8) είναι ο \hat{u}_{t-m} . Όμως, από την (4.6), το \hat{u}_{t-j} εκφράζεται σαν άθροισμα από bit πληροφορίας με το μικρότερο δυνατό δείκτη ίσο με $t - j - \hat{d}$. Εφόσον η αναπαράσταση του \hat{u}_{t-j} ως γραμμικός συνδυασμός bit πληροφορίας είναι μοναδικός, ισχύει $m \geq j + \hat{d}$, άτοπο.

Ας υποθέσουμε τώρα ότι τουλάχιστον ένα bit που δεν αντιστοιχεί σε νέα πληροφορία έχει μη μηδενικό συντελεστή στο γραμμικό συνδυασμό της σχέσης (4.8). Χρησιμοποιώντας τη σχέση (4.6), μπορούμε να αντικαταστήσουμε όλα τα bit που δεν αντιστοιχούν σε πληροφορία με

αθροίσματα από bit πληροφορίας. Δηλαδή, η (4.8) μπορεί να γραφεί σαν γραμμικός συνδυασμός bit πληροφορίας,

$$\hat{u}_{t-j} = \sum_{i=1}^{\hat{m}} \hat{\mu}_i \hat{u}_{t-i} \quad (4.9)$$

για κάποιο $\hat{m} \geq m$. Υποθέτουμε ότι $1 \leq q \leq m$ είναι ο μεγαλύτερος ακέραιος διάφορος του j , τέτοιος ώστε $\mu_q = 1$, όταν το \hat{u}_{t-q} δεν είναι bit πληροφορίας. Από την (4.6), το \hat{u}_{t-q} γράφεται ως

$$\hat{u}_{t-q} = \sum_{i=1}^{\hat{d}-1} c_i \hat{u}_{t-i-q} + \hat{u}_{t-\hat{d}-q}.$$

Εάν $\hat{d} + q > m$, τότε $\hat{m} = \hat{d} + q$. Χρησιμοποιώντας την ίδια επιχειρηματολογία όπως παραπάνω που αφορά στη μοναδικότητα της αναπαράστασης του \hat{u}_{t-j} σαν γραμμικός συνδυασμός από bit πληροφορίας, αυτό είναι εφικτό, μόνο αν $\hat{d} + q = \hat{d} + j$, δηλαδή $j = q$, άτοπο. Εάν $\hat{d} + q \leq m$, τότε $\hat{m} = m < \hat{d} + j$, τότε με παρόμοιο τρόπο οδηγούμαστε πάλι σε άτοπο.

Στην περίπτωση όπου $\hat{d} = 0$, οπότε και $c(X) = 0$, τότε από την (4.6) κάθε bit πληροφορίας \hat{u}_{t-j} , $j \leq m$, της κατάστασης \mathbf{S}_t είναι ίσο με το μηδέν, και συνεπώς είναι εξαρτημένο. \square

Είναι σημαντικό να παρατηρήσουμε πως από την απόδειξη του Λήμματος 4.3 προκύπτει ότι τελικά, κάθε εξαρτημένο bit της κατάστασης \mathbf{S}_t μπορεί να εκφραστεί σαν γραμμικός συνδυασμός bit πληροφορίας της \mathbf{S}_t .

Το διάγραμμα trellis που προκύπτει από τη διαδικασία απαλοιφής μονοπατιών είναι περιοδικό, και το αντίστοιχο trellis module απαρτίζεται από T_{pr} τμήματα trellis. Το τμήμα l , για $1 \leq l \leq T_{pr}$, περιέχει $2^{b_l^{pr}}$ ακμές (κλαδιά), όπου b_l^{pr} είναι η διάσταση του χώρου ακμών. Μία ακμή στο τμήμα l ενώνει μία κατάσταση σε βάθος l , με μία κατάσταση σε βάθος $l + 1$. Ο αριθμός των καταστάσεων σε βάθος l είναι ίσος με $2^{s_l^{pr}}$, όπου s_l^{pr} είναι η διάσταση του χώρου καταστάσεων.

Θα δείξουμε ότι τα s_l^{pr} και b_l^{pr} μπορούν να προσδιοριστούν πλήρως από την περίοδο T_{pr} , τη μνήμη m και από το βαθμό \hat{d} του πολυωνύμου $c(X)$.

Θεώρημα 4.4. Η διάσταση s_l^{pr} του χώρου καταστάσεων σε βάθος l του trellis module \mathcal{M}_{pr} δίνεται από τη σχέση

$$s_l^{pr} = \begin{cases} m - 1 - \left\lfloor \frac{m-\hat{d}}{T_{pr}} \right\rfloor, & \text{για } 1 \leq \delta \leq m - \hat{d} \bmod T_{pr} \\ m - \left\lfloor \frac{m-\hat{d}}{T_{pr}} \right\rfloor, & \text{για } (m - \hat{d} \bmod T_{pr}) + 1 \leq \delta \leq T_{pr} - 1 \text{ ή } \delta = 0 \end{cases} \quad (4.10)$$

όπου $\delta = l \bmod T_{pr}$.

Απόδειξη. Η διάσταση του χώρου καταστάσεων ισούται με τον αριθμό των γραμμικά ανεξάρτητων στοιχείων του διανύσματος \mathbf{S}_t . Εξ' ορισμού, τα μόνα στοιχεία που γράφονται ως γραμμικοί συνδυασμοί άλλων στοιχείων του \mathbf{S}_t είναι τα εξαρτημένα (συγκεκριμένα γράφονται

ως γραμμικοί συνδυασμοί bit πληροφορίας). Έστω D_t ο αριθμός των εξαρτημένων στοιχείων του \mathbf{S}_t , τη χρονική στιγμή t . Τότε, η διάσταση του χώρου καταστάσεων του \mathcal{M}_{pr} σε βάθος l είναι ίση με $m - D_t$, όπου $t \bmod T_{pr} = l \bmod T_{pr} = \delta$.

Από το Λήμμα 4.3, το bit \hat{u}_{t-j} , για $1 \leq j \leq m$ είναι εξαρτημένο, αν και μόνο αν $j_2 = \delta = t \bmod T_{pr}$ και $j_1 T_{pr} \leq m - \hat{d} - \delta$, όπου $j = j_1 T_{pr} + j_2$. Συνεπώς, το πλήθος D_t των εξαρτημένων bit του \mathbf{S}_t , δίνεται από τη σχέση

$$D_t = \begin{cases} \hat{\lambda} & , \text{ για } \delta = 0 \\ \hat{\lambda} + 1 & , \text{ για } \delta \neq 0 \end{cases} \quad (4.11)$$

όπου $\hat{\lambda}$ είναι ο μέγιστος ακέραιος που ικανοποιεί τη σχέση $\hat{\lambda} T_{pr} \leq m - \hat{d} - \delta$. Για $1 \leq \delta \leq m - \hat{d} \bmod T_{pr}$,

$$\hat{\lambda} = \left\lfloor \frac{m - \hat{d} - \delta}{T_{pr}} \right\rfloor = \left\lfloor \frac{m - \hat{d}}{T_{pr}} \right\rfloor. \quad (4.12)$$

Για $m - \hat{d} \bmod T_{pr} + 1 \leq \delta \leq T_{pr} - 1$,

$$\hat{\lambda} = \left\lfloor \frac{m - \hat{d} - \delta}{T_{pr}} \right\rfloor = \left\lfloor \frac{m - \hat{d}}{T_{pr}} \right\rfloor. \quad (4.13)$$

Για $m - \hat{d} \bmod T_{pr} + 1 \leq \delta \leq T_{pr} - 1$,

$$\hat{\lambda} = \left\lfloor \frac{m - \hat{d} - \delta}{T_{pr}} \right\rfloor = \left\lfloor \frac{m - \hat{d}}{T_{pr}} \right\rfloor - 1. \quad (4.14)$$

□

Σύμφωνα με τη μέθοδο απαλοιφής μονοπατιών που περιγράφεται σε αυτό το κεφάλαιο, τα bit εισόδου που σχετίζονται με τα πρώτα $T_{pr} - 1$ τμήματα trellis του trellis module \mathcal{M}_{pr} , είναι bit πληροφορίας. Επομένως, κάθε κατάσταση σε βάθος l , για $1 \leq l \leq T_{pr} - 1$, αποτελεί την αφετηρία δύο ακμών. Συνεπώς, η διάσταση του χώρου ακμών είναι ίση με τη διάσταση του χώρου καταστάσεων συν ένα. Κατά τη διάρκεια του τελευταίου τμήματος trellis, η είσοδος του κωδικοποιητή δεν είναι ένα bit πληροφορίας, αλλά ο γραμμικός συνδυασμός παλαιότερων bit πληροφορίας που βρίσκονται αποθηκευμένα στον καταχωρητή. Επομένως, μόνο μία ακμή ξεκινάει από κάθε κατάσταση σε βάθος T_{pr} . Η διάσταση του χώρου ακμών του τμήματος trellis T_{pr} είναι ίση με τη διάσταση του χώρου καταστάσεων σε βάθος T_{pr} . Συνεπώς ισχύει ότι

$$b_l^{pr} = \begin{cases} m - \left\lfloor \frac{m - \hat{d}}{T_{pr}} \right\rfloor & , \text{ για } 0 \leq \delta \leq m - \hat{d} \bmod T_{pr} \\ m + 1 - \left\lfloor \frac{m - \hat{d}}{T_{pr}} \right\rfloor & , \text{ για } (m - \hat{d} \bmod T_{pr}) + 1 \leq \delta \leq T_{pr} - 1 \end{cases} \quad (4.15)$$

όπου $\delta = l \bmod T_{pr}$. Ο συνολικός αριθμός ακμών $|B_{pr}|$ και καταστάσεων $|\Sigma_{pr}|$ του trellis module \mathcal{M}_{pr} δίνονται απο την

$$|B_{pr}| = \sum_{l=1}^{T_{pr}} 2^{b_l^{pr}} = (2T_{pr} - \hat{\alpha} - 1) \cdot 2^{m-\hat{\beta}} \quad (4.16)$$

και

$$|\Sigma_{pr}| = \sum_{l=1}^{T_{pr}} 2^{s_l^{pr}} = (2T_{pr} - \hat{\alpha}) \cdot 2^{m-1-\hat{\beta}} \quad (4.17)$$

όπου

$$\hat{\alpha} = m - \hat{d} \bmod T_{pr} \quad (4.18)$$

και

$$\hat{\beta} = \left\lfloor \frac{m - \hat{d}}{T_{pr}} \right\rfloor. \quad (4.19)$$

Ο αριθμός των συνενώσεων σε μία συγκεκριμένη κατάσταση σε βάθος $l + 1$ ισούται με τον αριθμό των ακμών που καταλήγουν σε αυτήν μείον ένα. Οπότε, ο αριθμός των συνενώσεων που περιέχονται στο τμήμα trellis l (δηλαδή ο αριθμός των συνενώσεων σε βάθος $l + 1$) ισούται με $2^{b_l^{pr}} - 2^{s_{l+1}^{pr}}$. Συνεπώς, από τις σχέσεις (4.16) και (4.17) προκύπτει πως ο συνολικός αριθμός E_{pr} των συνενώσεων που περιέχονται στο trellis module \mathcal{M}_{pr} δίνεται από τη σχέση

$$E_{pr} = \sum_{l=1}^{T_{pr}} (2^{b_l^{pr}} - 2^{s_{l+1}^{pr}}) = |B_{pr}| - |\Sigma_{pr}| = (T_{pr} - \frac{\hat{\alpha}}{2} - 1) \cdot 2^{m-\hat{\beta}}. \quad (4.20)$$

Αξίζει να σημειωθεί εδώ ότι $s_{T_{pr}+1}^{pr} = s_1^{pr}$.

4.2.2 Απαλοιφή Κωδικών Bit

Η δεύτερη και τελευταία φάση της κατασκευής των ευέλικτων κωδικών περιλαμβάνει την απαλοιφή κωδικών bit, προκειμένου να επιτευχθεί ο επιθυμητός κωδικός ρυθμός. Πιο συγκεκριμένα, το τελικό trellis module \mathcal{M}_{pu} κατασκευάζεται επαναλαμβάνοντας p φορές το trellis module \mathcal{M}_{pr} , και στη συνέχεια διαγράφοντας μία ποσότητα κωδικών bit βάσει συγκεκριμένου πίνακα απαλοιφής P . Το τελικό trellis module αποτελείται από $T_{pu} = pT_{pr}$ τμήματα trellis και το μέγεθος του P είναι $n \times T_{pu}$.

Ο πίνακας γεννήτορας $\mathbf{G}_{pu}(D)$ του κώδικα που προκύπτει από τη διαδικασία της απαλοιφής κωδικών bit δίνεται από τη σχέση [22]

$$\mathbf{G}_{pu}(D) = (\mathbf{\Theta}(D) \cdot \mathbf{G}^{[T_{pr}]}(D))^{[p]} \cdot \mathcal{P} \quad (4.21)$$

όπου $\mathbf{\Theta}(D) \cdot \mathbf{G}^{[T_{pr}]}(D)$ είναι ο πίνακας γεννήτορας του $(nT_{pr}, T_{pr} - 1)$ κώδικα \mathcal{C}_{pr} που προκύπτει από τη διαδικασία της απαλοιφής μονοπατιών, και $(\mathbf{\Theta}(D) \cdot \mathbf{G}^{[T_{pr}]}(D))^{[p]}$ ο ισοδύναμος $p(T_{pr} -$

1) $\times pnT_{pr}$ γεννήτορας πίνακας που παράγει τον \mathcal{C}_{pr} , όταν ο τελευταίος αντιμετωπίζεται ως περιοδικός με περίοδο p [22], [49]. Ο πίνακας \mathcal{P} προκύπτει από τον $nT_{pu} \times nT_{pu}$ μοναδιαίο πίνακα διαγράφοντας τη στήλη w , $\forall w \in \{n(j-1)+i \mid \forall p_{i,j} = 0\}$, όπου το $p_{i,j}$ είναι στοιχείο του πίνακα απαλοιφής P , για $1 \leq i \leq n$ και $1 \leq j \leq T_{pu}$.

Η διαδικασία της απαλοιφής κωδικών bit δεν επηρεάζει το συνολικό αριθμό ακμών, καταστάσεων και συνενώσεων, επομένως προκύπτει άμεσα πως ο συνολικός αριθμός καταστάσεων $|\Sigma_{pu}|$ και συνενώσεων E_{pu} που περιέχονται στο τελικό trellis module \mathcal{M}_{pu} είναι

$$|\Sigma_{pu}| = p|\Sigma_{pr}| \quad (4.22)$$

και

$$E_{pu} = pE_{pr}. \quad (4.23)$$

Η trellis πολυπλοκότητα του τελικού trellis module \mathcal{M}_{pu} εξαρτάται από τον πίνακα απαλοιφής κωδικών bit, και είναι ίση με

$$TC(\mathcal{M}_{pu}) = \sum_{j_1=0}^{p-1} \sum_{j_2=1}^{T_{pr}} n_{j_1 T_{pr} + j_2} 2^{b_{j_2}^{pr}}, \quad (4.24)$$

όπου $0 \leq n_j \leq n$ είναι ο αριθμός κωδικών bit του τμήματος j του \mathcal{M}_{pu} , για $1 \leq j \leq pT_{pr}$. Το τελικό trellis module \mathcal{M}_{pu} αντιστοιχεί σε έναν $(pT_{pr}n - N, p(T_{pr} - 1))$ συγκεραστικό κώδικα, όπου N είναι ο αριθμός των bit που απαλείφονται, δηλαδή $pT_{pr}n - N = \sum_{j=1}^{pT_{pr}} n_j$.

4.2.3 Περιορισμοί

Δοθέντος ενός $(n, 1, m)$ μητρικού κώδικα, για διάφορες επιλογές των T_{pr} , p , $c(X)$ και του πίνακα απαλοιφής κωδικών bit P , μπορούν να κατασκευαστούν κώδικες διαφόρων ρυθμών και υπολογιστικής πολυπλοκότητας αποκωδικοποίησης. Στην παρούσα διατριβή θα περιοριστούμε σε ένα υποσύνολο αυτών των κωδικών, σύμφωνα με το παρακάτω κριτήριο. Για συγκεκριμένο ρυθμό k'/n' , κατασκευάζουμε κώδικες υπολογιστικής πολυπλοκότητας αποκωδικοποίησης ίσης με αυτής των PCC μνήμης m' , για διάφορες τιμές του m' .

Η trellis πολυπλοκότητα του trellis module ενός (n', k', m') PCC [25] ισούται με

$$TC_{PCC} = n' \cdot 2^{m'+1}. \quad (4.25)$$

Επίσης, εκ κατασκευής το trellis module ενός PCC αποτελείται από k' τμήματα. Ο αριθμός των καταστάσεων είναι σταθερός σε κάθε βάθος, και ίσος με $2^{m'}$. Επίσης, σε κάθε κατάσταση καταλήγουν δύο ακμές, συνεπώς κάθε κατάσταση αντιστοιχεί σε μία συνένωση. Έτσι, ο συνολικός αριθμός συνενώσεων που περιέχεται στο trellis module ενός (n', k', m') PCC δίνεται από

τη σχέση

$$E_{PCC} = k' \cdot 2^{m'}. \quad (4.26)$$

Ο πρώτος περιορισμός αφορά στον αριθμό των συνενώσεων που περιέχονται στο trellis module \mathcal{M}_{pu} , ο οποίος πρέπει να είναι ίσος με τον αριθμό των συνενώσεων που περιέχονται στο trellis module ενός $(pT_{pr}n - N, p(T_{pr} - 1), m')$ PCC. Συνεπώς, από τις σχέσεις (4.20), (4.23) και (4.26) έχουμε ότι

$$\begin{aligned} p(T_{pr} - \frac{\hat{\alpha}}{2} - 1) \cdot 2^{m-\hat{\beta}} &= p(T_{pr} - 1) \cdot 2^{m'} \Leftrightarrow \\ 1 - \frac{\hat{\alpha}}{2(T_{pr} - 1)} &= 2^{m'-m+\hat{\beta}}. \end{aligned} \quad (4.27)$$

Εφόσον οι m' , m και $\hat{\beta}$ είναι ακέραιοι αριθμοί, το αριστερό μέλος της σχέσης (4.27) θα πρέπει να είναι της μορφής 2^ξ , όπου ο ξ είναι ακέραιος αριθμός. Επίσης, εξ' ορισμού $0 \leq \hat{\alpha} \leq T_{pr} - 1$, συνεπώς ισχύει ότι $\frac{1}{2} \leq 1 - \frac{\hat{\alpha}}{2(T_{pr}-1)} \leq 1$. Έτσι, το ξ μπορεί να πάρει μόνο τις τιμές 0 και -1, και η σχέση (4.27) ικανοποιείται μόνο σε δύο περιπτώσεις, δηλαδή όταν

$$\hat{\alpha} = 0 \text{ και } m' = m - \hat{\beta} \quad (4.28)$$

ή

$$\hat{\alpha} = T_{pr} - 1 \text{ και } m' = m - 1 - \hat{\beta}. \quad (4.29)$$

Ο δεύτερος περιορισμός αφορά στην trellis πολυπλοκότητα του trellis module \mathcal{M}_{pu} , η οποία πρέπει να είναι ίση με αυτή του trellis module ενός $(pT_{pr}n - N, p(T_{pr} - 1), m')$ PCC. Συνεπώς, από τις σχέσεις (4.24) και (4.25) ισχύει ότι

$$\sum_{j_1=0}^{p-1} \sum_{j_2=1}^{T_{pr}} n_{j_1 T_{pr} + j_2} 2^{b_{j_2}^{pr}} = (pT_{pr}n - N) \cdot 2^{m'+1}. \quad (4.30)$$

Για $\hat{\alpha} = 0$, από τη σχέση (4.15), η διάσταση του χώρου ακμών $T_{pr} - 1$ τμημάτων trellis του \mathcal{M}_{pr} είναι ίση με $m + 1 - \hat{\beta}$, ενώ η διάσταση του χώρου ακμών του τελευταίου τμήματος είναι ίση με $m - \hat{\beta}$. Επομένως, από τη σχέση (4.28), η σχέση (4.30) γράφεται ως

$$\sum_{j_1=0}^{p-1} \sum_{j_2=1}^{T_{pr}-1} n_{j_1 T_{pr} + j_2} 2^{m+1-\hat{\beta}} + \sum_{j_1=1}^p n_{j_1 T_{pr}} 2^{m-\hat{\beta}} = (pT_{pr}n - N) \cdot 2^{m-\hat{\beta}+1}. \quad (4.31)$$

Δηλαδή,

$$\begin{aligned} \sum_{j_1=0}^{p-1} \sum_{j_2=1}^{T_{pr}-1} n_{j_1 T_{pr} + j_2} + \frac{1}{2} \sum_{j_1=1}^p n_{j_1 T_{pr}} &= (p T_{pr} n - N) \\ &= \sum_{j_1=0}^{p-1} \sum_{j_2=1}^{T_{pr}-1} n_{j_1 T_{pr} + j_2} + \sum_{j_1=1}^p n_{j_1 T_{pr}}. \end{aligned}$$

Η παραπάνω συνθήκη ικανοποιείται μόνο όταν $\sum_{j_1=1}^p n_{j_1 T_{pr}} = 0$, δηλαδή όταν όλα τα κωδικά bit του τελευταίου τμήματος του \mathcal{M}_{pr} , ή ισοδύναμα όταν όλα τα κωδικά bit των τμημάτων σT_{pr} του \mathcal{M}_{pu} , για $1 \leq \sigma \leq p$ απαλείφονται. Επομένως, ο αριθμός των κωδικών bit μπορεί να κυμαίνεται από $pn(T_{pr} - 1)$ σε $p(T_{pr} - 1) + 1$, και ο ρυθμός k'/n' των κατασκευαζόμενων κωδικών, οι οποίοι έχουν ίδια υπολογιστική πολυπλοκότητα αποκωδικοποίησης με τους PCCs μνήμης $m' = m - \hat{\beta}$, μπορεί να είναι

$$\frac{p(T_{pr} - 1)}{pn(T_{pr} - 1)} \leq \frac{k'}{n'} \leq \frac{p(T_{pr} - 1)}{p(T_{pr} - 1) + 1}. \quad (4.32)$$

Παρομοίως, στη δεύτερη περίπτωση, για $\hat{\alpha} = T_{pr} - 1$, από τη σχέση (4.15) η διάσταση του χώρου ακμών όλων των τμημάτων του \mathcal{M}_{pr} (και συνεπώς του \mathcal{M}_{pu}) είναι ίση με $m - \hat{\beta}$. Επομένως, με βάση την (4.29), η (4.30) γράφεται

$$\sum_{j=1}^{p T_{pr}} n_j 2^{m-\hat{\beta}} = (p T_{pr} n - N) \cdot 2^{m-\hat{\beta}}. \quad (4.33)$$

Οπότε, ο αριθμός των κωδικών bit μπορεί να κυμαίνεται από $p T_{pr} n$ σε $p(T_{pr} - 1) + 1$, και ο ρυθμός k'/n' των κατασκευαζόμενων κωδικών, οι οποίοι έχουν ίδια υπολογιστική πολυπλοκότητα αποκωδικοποίησης με τους PCCs μνήμης $m' = m - \hat{\beta} - 1$, μεταβάλλεται στο διάστημα

$$\frac{p(T_{pr} - 1)}{pn T_{pr}} \leq \frac{k'}{n'} \leq \frac{p(T_{pr} - 1)}{p(T_{pr} - 1) + 1}. \quad (4.34)$$

4.3 Διαδικασία Κατασκευής – Αναζήτησης

Στην ενότητα 4.2 παρουσιάστηκε ο τρόπος με τον οποίο εφαρμόζουμε τις τεχνικές της απαλοιφής κωδικών bit και της απαλοιφής μονοπατιών του διαγράμματος trellis, ενώ αναλύθηκε και το "προφίλ πολυπλοκότητας" των παραγόμενων trellis modules. Βασικός στόχος είναι η χρήση αυτών των αποτελεσμάτων στην ανάπτυξη ευέλικτων συγκεραστικών κωδικών, δηλαδή στην κατασκευή μιας ολόκληρης οικογένειας κωδικών διαφόρων ρυθμών και τιμών της υπολογιστικής πολυπλοκότητας αποκωδικοποίησης από έναν $(n, 1, m)$ μητρικό συγκεραστικό κώδικα.

Θεωρείστε έναν $(n, 1, m)$ μπτρικό συγκεραστικό κώδικα και έναν κωδικό ρυθμό k'/n' , ο οποίος θέλουμε να περιέχεται στην οικογένεια. Όπως είδαμε και στην προηγούμενη ενότητα, για συγκεκριμένο ρυθμό κατασκευάζουμε κώδικες των οποίων η υπολογιστική πολυπλοκότητα αποκωδικοποίησης είναι ίση με αυτή των αντίστοιχων PCCs μνήμης m' , όπου $m' \leq m$ (αυτό προκύπτει από τις (4.28) και (4.29)). Η διαδικασία ξεκινάει με την κατασκευή ενός (n', k', m) PCC, χρησιμοποιώντας το μπτρικό κώδικα. Σ' αυτή την φάση δεν εφαρμόζουμε τη μέθοδο της απαλοιφής μονοπατιών, αφού κώδικες αυτής της πολυπλοκότητας μπορούν να παραχθούν απλά χρησιμοποιώντας τη μέθοδο απαλοιφής κωδικών bit. Δηλαδή, απαλείφοντας περιοδικά συγκεκριμένο αριθμό κωδικών bit από το μπτρικό κώδικα (από το διάγραμμα trellis αυτού), σύμφωνα με συγκεκριμένο $n \times k'$ πίνακα απαλοιφής, προκειμένου να επιτευχθεί ο επιθυμητός ρυθμός. Προφανώς, από όλους τους πιθανούς πίνακες απαλοιφής, επιλέγεται αυτός που παράγει τον καλύτερο (n', k', m) PCC (από πλευράς ελάχιστης απόστασης και φάσματος γενικότερα). Ο συγκεκριμένος κώδικας έχει τη μεγαλύτερη υπολογιστική πολυπλοκότητα αποκωδικοποίησης ανάμεσα στους κώδικες ρυθμού k'/n' που περιέχονται στην οικογένεια.

Το επόμενο βήμα είναι η κατασκευή των υπόλοιπων συγκεραστικών κωδικών k'/n' της οικογένειας, η υπολογιστική πολυπλοκότητα (αποκωδικοποίησης) των οποίων είναι ίση με την αντίστοιχη των PCC με μνήμη m' , για $m' < m$. Σ' αυτήν την περίπτωση χρησιμοποιείται η μέθοδος απαλοιφής μονοπατιών και η μέθοδος απαλοιφής κωδικών bit. Θεωρούμε ότι $k'/n' \geq 1/n$, συνεπώς ικανοποιούνται οι σχέσεις (4.32) και (4.34). Δοθέντος του ρυθμού k'/n' , αλλά και του m' , πρέπει να βρούμε όλες τις τριάδες (p, T_{pr}, \hat{d}) οι οποίες σε συνδυασμό με τον κατάλληλο πίνακα απαλοιφής οδηγούν σε trellis modules με την επιθυμητή υπολογιστική πολυπλοκότητα αποκωδικοποίησης. Στη συνέχεια εξετάζουμε την ύπαρξη τέτοιων τριάδων (p, T_{pr}, \hat{d}) , δηλαδή την ύπαρξη trellis modules των οποίων η υπολογιστική πολυπλοκότητα της αποκωδικοποίησης ισούται με την αντίστοιχη πολυπλοκότητα ενός (n', k', m') PCC.

Από τον τρόπο με τον οποίο εφαρμόζουμε τις τεχνικές της απαλοιφής μονοπατιών και της απαλοιφής κωδικών bit προκύπτει ότι

$$k' = p(T_{pr} - 1). \quad (4.35)$$

Συνεπώς, υπάρχει τέτοιο trellis module, αν υπάρχει ένας διαιρέτης $T_{pr} - 1 \geq 1$ του k' τέτοιος ώστε, για συγκεκριμένες τιμές των m και m' , να υπάρχει ένας ακέραιος $0 \leq \hat{d} \leq m$, που προκύπτει είτε από την (4.28)

$$\hat{d} = m'T_{pr} - m(T_{pr} - 1) \quad (4.36)$$

είτε από την (4.29)

$$\hat{d} = m'T_{pr} - m(T_{pr} - 1) + 1. \quad (4.37)$$

Επιπροσθέτως, εξ' ορισμού το \hat{d} δεν πρέπει να είναι πολλαπλάσιο του T_{pr} .

Από τη σχέση (4.36), προκύπτει ότι $0 \leq m'T_{pr} - m(T_{pr} - 1) \leq m$, δηλαδή

$$m \frac{T_{pr} - 1}{T_{pr}} \leq m' \leq m. \quad (4.38)$$

Επίσης, από την (4.37) προκύπτει ότι $0 \leq m'T_{pr} - m(T_{pr} - 1) + 1 \leq m$, δηλαδή

$$m \frac{T_{pr} - 1}{T_{pr}} - \frac{1}{T_{pr}} \leq m' \leq m - \frac{1}{T_{pr}}. \quad (4.39)$$

Εφόσον $m' < m$, τότε τα άνω φράγματα στις ανισότητες (4.38) και (4.39) ικανοποιούνται πάντα.

Για κάθε διαιρέτη $T_{pr} - 1$ του k' , ο οποίος ικανοποιεί την (4.38) ή την (4.39), και για τον οποίο το \hat{d} που προκύπτει δεν είναι πολλαπλάσιο του T_{pr} , υπάρχει μία “έγκυρη” τριάδα (p, T_{pr}, \hat{d}) η οποία (σε συνδυασμό με κατάλληλο πίνακα απαλοιφής κωδικών bit) οδηγεί σε ένα trellis module με την επιθυμητή υπολογιστική πολυπλοκότητα αποκωδικοποίησης. Ο συνολικός αριθμός τέτοιων τριάδων (p, T_{pr}, \hat{d}) είναι μικρότερος ή ίσος από το διπλάσιο του αριθμού των διαιρετών του k' .

Εφόσον $T_{pr} \geq 2$, το m' δεν μπορεί να πάρει οποιαδήποτε τιμή για δεδομένο m . Πιο συγκεκριμένα, από την (4.38),

$$\frac{m}{2} \leq m' \quad (4.40)$$

και από την (4.39)

$$\frac{m-1}{2} \leq m'. \quad (4.41)$$

Δηλαδή, η μικρότερη δυνατή τιμή του m' είναι η $\lceil \frac{m-1}{2} \rceil$.

Θα δείξουμε ότι υπάρχει τουλάχιστον μία αποδεκτή τριάδα (p, T_{pr}, \hat{d}) , δεδομένου ότι $m' \geq \lceil \frac{m-1}{2} \rceil$. Για τον προφανή διαιρέτη $T_{pr} - 1 = 1$ του k' , από τις σχέσεις (4.35) και (4.36) προκύπτει η τριάδα

$$(p, T_{pr}, \hat{d}) = (k', 2, 2m' - m) \quad (4.42)$$

και από τις σχέσεις (4.35) και (4.37) προκύπτει η τριάδα

$$(p, T_{pr}, \hat{d}) = (k', 2, 2m' - m + 1). \quad (4.43)$$

Συγκεκριμένα, για $m' = \lceil \frac{m-1}{2} \rceil$, εάν το m είναι άρτιος αριθμός, τότε ικανοποιείται και η σχέση (4.40) και η (4.41), οπότε και οι δύο τιμές του \hat{d} στην (4.42) και στην (4.43) είναι μη αρνητικές. Επίσης, όπως δεν μπορούν να είναι ταυτόχρονα και οι δύο πολλαπλάσια του $T_{pr} = 2$. Άρα,

τουλάχιστον μία εκ των (4.42) και (4.43) είναι αποδεκτή τριάδα. Εάν το m είναι περιττός αριθμός, τότε μόνο η (4.41) ικανοποιείται, δηλαδή μόνο στην (4.43) το \hat{d} είναι μη αρνητικό. Συγκεκριμένα, το \hat{d} στην (4.43) είναι ίσο με το μηδέν, συνεπώς η (4.43) είναι αποδεκτή. Τέλος, για $m' > \lceil \frac{m-1}{2} \rceil$, και οι δύο εκ των (4.40) και (4.41) ικανοποιούνται, ενώ οι τιμές του \hat{d} στην (4.42) και στην (4.43) δε μπορεί να είναι ταυτόχρονα πολλαπλάσια του $T_{pr} = 2$. Οπότε τουλάχιστον μία εκ των (4.42) και (4.43) είναι αποδεκτή τριάδα.

Για συγκεκριμένη τιμή του m' , κάθε μία από τις τριάδες (p, T_{pr}, \hat{d}) , μαζί με τον κατάλληλο πίνακα απαλοιφής P οδηγεί σε ένα trellis module pT_{pr} τμημάτων το οποίο αντιστοιχεί σε έναν κώδικα ρυθμού k'/n' . Η υπολογιστική πολυπλοκότητα αποκωδικοποίησης είναι ίση με αυτή του αντίστοιχου (n', k', m') PCC. Επίσης, πρέπει να τονίσουμε ότι η παράμετρος \hat{d} είναι ο βαθμός του πολυωνύμου $c(X)$. Συνεπώς, για δεδομένη τιμή του \hat{d} , θα πρέπει να λάβουμε υπόψιν μας όλα τα αποδεκτά πολυώνυμα $c(X)$ του συγκεκριμένου βαθμού. Για κάθε τριάδα (p, T_{pr}, \hat{d}) , επιλέγουμε το πολυώνυμο $c(X)$ και τον πίνακα απαλοιφής P που παράγουν τον καλύτερο κώδικα. Τέλος, επιλέγουμε τον καλύτερο κώδικα απ' όλες τις τριάδες.

Παράδειγμα 4.5. Έστω ένας $(2, 1, 3)$ μητρικός συγκεραστικός κώδικας με πίνακα γεννήτορα

$$\mathbf{G}(D) = [1 + D + D^3 \quad 1 + D + D^2 + D^3]. \quad (4.44)$$

Θα κατασκευάσουμε όλους τους κώδικες της οικογένειας με ρυθμό $3/4$. Σε πρώτη φάση κατασκευάζουμε έναν $(4, 3, 3)$ PCC (η διαδικασία απαλοιφής μονοπατιών δεν εφαρμόζεται), δηλαδή “συνδέουμε” τρία trellis modules του μητρικού κώδικα, και αναζητούμε τον πίνακα απαλοιφής κωδικών bit P που οδηγεί στον καλύτερο κώδικα. Στην προκειμένη περίπτωση, ο πίνακας απαλοιφής είναι ο

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

Το trellis module του κώδικα \mathcal{C}_1 που προκύπτει φαίνεται στο Σχ. 4.2, όπου ένα διαγραμμένο κωδικό bit συμβολίζεται με “x”. Η trellis πολυπλοκότητα του είναι ίση με 64, ενώ ο συνολικός αριθμός συνενώσεων είναι ίσος με 24. Η ελεύθερη απόσταση του \mathcal{C}_1 είναι 4.

Στη συνέχεια κατασκευάζουμε έναν συγκεραστικό κώδικα του οποίου η υπολογιστική πολυπλοκότητα αποκωδικοποίησης είναι ίση με την πολυπλοκότητα του αντίστοιχου $(4, 3, 2)$ PCC. Για $k' = 3$, $m' = 2$ και $m = 3$, το σύστημα εξισώσεων που αποτελείται από τις σχέσεις (4.28) και (4.35) έχει μία αποδεκτή λύση, δηλαδή την τριάδα $(p, T_{pr}, \hat{d}) = (3, 2, 1)$. Επίσης, και το σύστημα εξισώσεων που αποτελείται από τις σχέσεις (4.29) και (4.35) έχει μία αποδεκτή λύση, δηλαδή την τριάδα $(p, T_{pr}, \hat{d}) = (1, 4, 0)$. Ο καλύτερος κώδικας \mathcal{C}_2 προκύπτει από την τριάδα $(p, T_{pr}, \hat{d}) = (1, 4, 0)$ σε συνδυασμό με τον πίνακα απαλοιφής κωδικών bit

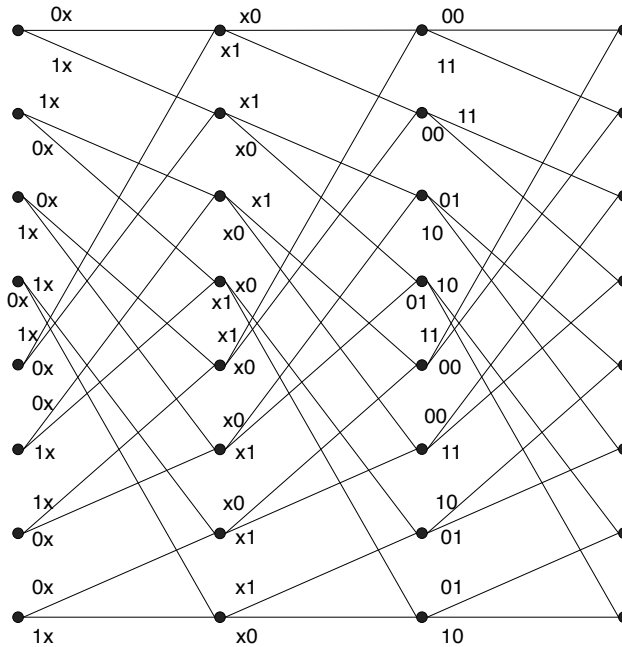
$$P = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Αξίζει να σημειωθεί πως εφόσον $\hat{d} = 0$, $c(X) = 0$, δηλαδή κάθε τρία bit πληροφορίας εισάγεται στο μπτρικό κωδικοποιητή ένα *πλασματικό* (dummy) μηδενικό bit (αυτό προκύπτει από την (4.6)). Το αντίστοιχο trellis module φαίνεται στο Σχ. 4.3. Η trellis πολυπλοκότητα είναι ίση με 32, και ο συνολικός αριθμός συνενώσεων είναι 12. Η ελεύθερη απόσταση του \mathcal{C}_2 είναι 3.

Τέλος κατασκευάζουμε έναν κώδικα ρυθμού 3/4 για $m' = 1$. Η μοναδική αποδεκτή λύση προκύπτει από το σύστημα εξισώσεων που αποτελείται από τις σχέσεις (4.29) και (4.35), κι αυτή είναι η τριάδα $(p, T_{pr}, \hat{d}) = (3, 2, 0)$. Η τελευταία σε συνδυασμό με τον πίνακα απαλοιφής κωδικών bit

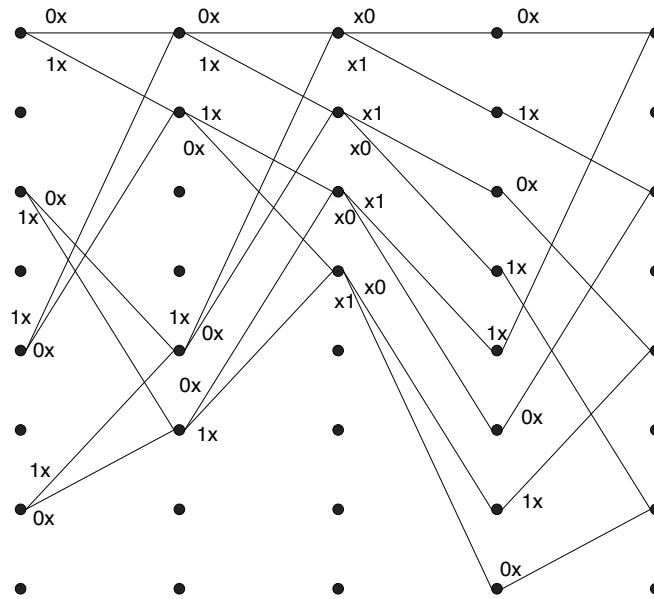
$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

παράγουν τον κώδικα \mathcal{C}_3 , του οποίου το trellis module απεικονίζεται στο Σχ. 4.4. Η trellis πολυπλοκότητα είναι ίση με 16 και ο συνολικός αριθμός συνενώσεων είναι 6. Η ελεύθερη απόσταση του \mathcal{C}_3 είναι 2.

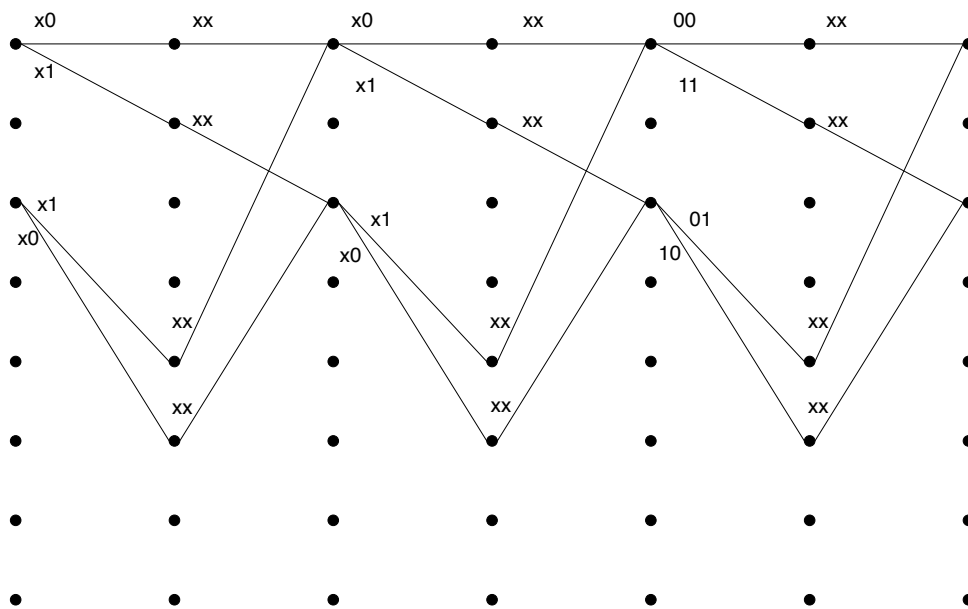


Σχήμα 4.2. Το trellis module του κώδικα \mathcal{C}_1 στο Παράδειγμα 4.5.

Η παράμετρος που επηρεάζει σημαντικά την πολυπλοκότητα της διαδικασίας αναζήτησης είναι το πλήθος των πινάκων απαλοιφής κωδικών bit που πρέπει να εξεταστούν, το οποίο αυξάνεται με το αριθμό n των κωδικών bit του μπτρικού κώδικα. Θεωρείστε έναν $(n, 1, m)$ μπτρικό κώδικα και μία τριάδα (p, T_{pr}, \hat{d}) , η οποία σε συνδυασμό με κατάλληλο πίνακα απαλοιφής κωδικών bit οδηγεί στην κατασκευή ενός (n', k', m') κώδικα. Υποθέστε επίσης ότι έχει επιλεγεί συγκεκριμένο πολυώνυμο $c(X)$ βαθμού \hat{d} για τη διαδικασία απαλοιφής μονοπατιών. Εάν το \hat{d} προκύπτει από τη σχέση (4.36) (δηλαδή ικανοποιείται η σχέση (4.28)), τότε όπως επισημάνθηκε στην υποενότητα 4.2.3, όλα τα κωδικά bit των τμημάτων σT_{pr} , $1 \leq \sigma \leq p$ του



Σχήμα 4.3. Το trellis module του κώδικα C_2 στο Παράδειγμα 4.5.



Σχήμα 4.4. Το trellis module του κώδικα C_3 στο Παράδειγμα 4.5.

trellis module \mathcal{M}_{pu} πρέπει να διαγραφούν, προκειμένου να επιτευχθεί η επιθυμητή υπολογιστική πολυπλοκότητα. Συνεπώς, ο συνολικός αριθμός των πινάκων απαλοιφής κωδικών bit που πρέπει να εξεταστούν (για συγκεκριμένο $c(X)$) είναι ίσος με

$$\binom{npT_{pr} - np}{n'} = \binom{np(T_{pr} - 1)}{n'} = \binom{nk'}{n'}. \quad (4.45)$$

Αξίζει να σημειωθεί πως αυτός είναι ο αριθμός των πινάκων απαλοιφής κωδικών bit που πρέπει να εξεταστεί προκειμένου να βρεθεί ο καλύτερος (n', k', m') “κλασσικός” PCC που προκύπτει από έναν $(n, 1, m')$ μπηρικό κώδικα. Η διαδικασία αυτή δεν είναι ιδιαίτερα απαιτητική (όσον αφορά στην πολυπλοκότητα) για λογικές τιμές του n , και έχει εφαρμοστεί πολλές φορές στη βιβλιογραφία [11], [12]. Εάν το \hat{d} προκύπτει από τη σχέση (4.37) (δηλαδή ικανοποιείται η σχέση (4.29)), τότε το πλήθος των πινάκων απαλοιφής κωδικών bit ανέρχεται σε

$$\binom{npT_{pr}}{n'}. \quad (4.46)$$

Ο συνολικός αριθμός λοιπόν των πινάκων απαλοιφής κωδικών bit αυξάνεται συγκριτικά με την προηγούμενη περίπτωση. Μπορεί να δειχθεί όμως ότι ο αριθμός αυτός είναι μικρότερος ή ίσος με τον αριθμό των πινάκων απαλοιφής κωδικών bit που πρέπει να εξεταστούν προκειμένου να βρεθεί ο καλύτερος (n', k', m') “κλασσικός” PCC που προκύπτει από έναν $(n_1, 1, m')$ μπηρικό κώδικα, όπου $n_1 = \left\lceil \frac{nT_{pr}}{T_{pr}-1} \right\rceil$. Παρόλ' αυτά, η ύπαρξη ισοδυναμιών μεταξύ διαφόρων πινάκων απαλοιφής κωδικών bit, μειώνει τον αρχικό αριθμό πινάκων που πρέπει να εξεταστούν. Για παράδειγμα, είναι γνωστό πως δύο trellis modules όπου το ένα αποτελεί κυκλική ολίσθηση (cyclic shift) του άλλου, είναι ισοδύναμα με την έννοια ότι παράγουν κώδικες πανομοιότυπων φασμάτων [30]. Επίσης, προκειμένου να επιταχυνθεί η διαδικασία αναζήτησης, η παραγωγή μερικών κωδικών λέξεων πρώτα (αντί του υπολογισμού της ελάχιστης απόστασης, ή κάποιων όρων των φασμάτων) είναι αρκετή για την απόρριψη πολλών κακών πινάκων απαλοιφής κωδικών bit, και κατ' επέκταση κωδικών. Τέλος, σε περιπτώσεις όπου η εξέταση όλων των πιθανών πινάκων απαλοιφής αποτελεί ιδιαίτερα χρονοβόρα διαδικασία, εκμεταλλευτήκαμε ένα ενδιαφέρον χαρακτηριστικό των συγκεκριμένων κωδικών. Όπως αποδεικνύεται και στο Παράρτημα 4.A, το trellis module \mathcal{M}_{pu} μπορεί να μετασχηματιστεί σε ένα *information bit-oriented* trellis module (δηλαδή κάθε τμήμα του trellis module αντιστοιχεί σε ένα bit πληροφορίας) k' τμημάτων, του οποίου η διάσταση του χώρου καταστάσεων παραμένει σταθερή σε κάθε βάθος και ίση με m' , όπως ακριβώς το trellis module των PCC δηλαδή, ή των κωδικών που παρουσιάζονται στα [28], [27]. Ακολουθώντας μία “κλασσική” σχεδιαστική τεχνική η οποία εφαρμόζεται σ' αυτές τις περιπτώσεις [11, 89, 12], εξετάσαμε μόνο πίνακες απαλοιφής σύμφωνα με τους οποίους παράγεται τουλάχιστον ένα κωδικό bit για κάθε bit πληροφορίας (και συνεπώς σε κάθε τμήμα του *information bit-oriented trellis module*).

Όσον αφορά στον αριθμό των διαφορετικών πολυωνύμων $c(X)$ που πρέπει να ληφθούν υπόψη για την κατασκευή ενός (n', k', m') κώδικα από έναν $(n, 1, m)$ μητρικό κώδικα, $2^{\hat{d}-1}$ (για $\hat{d} > 0$) πολυώνυμα πρέπει να εξεταστούν για κάθε τριάδα (p, T_{pr}, \hat{d}) . Παρόλ' αυτά, για πρακτικούς κωδικούς ρυθμούς οι διαιρέτες του k' είναι ελάχιστοι, ενώ κάποιιοι από αυτούς δεν οδηγούν σε αποδεκτή τριάδα. Επίσης, για πρακτικές τιμές του m , ο αριθμός των πολυωνύμων που αντιστοιχούν σε μία τριάδα (p, T_{pr}, \hat{d}) είναι μικρός, αφού $d \leq m$. Τέλος, εξ' ορισμού κάποια από τα πολυώνυμα απορρίπτονται, αφού $c_i = 0$ για $0 = i \pmod{T_{pr}}$. Έτσι, στις περισσότερες των περιπτώσεων τα πολυώνυμα που πρέπει να εξεταστούν είναι λίγα.

4.4 Οικογένειες Ευέλικτων Συγκεραστικών Κωδίκων

Χρησιμοποιώντας τη μέθοδο που περιγράψαμε στην προηγούμενη ενότητα κατασκευάσαμε οικογένειες ευέλικτων κωδίκων. Οι Πίνακες 4.1–4.3 περιέχουν τρεις οικογένειες, που παράγονται από έναν $(2, 1, 8)$, έναν $(3, 1, 8)$ κι έναν $(4, 1, 8)$ μητρικό συγκεραστικό κώδικα αντίστοιχα. Όλες οι οικογένειες περιέχουν κωδικές ρυθμών $(n' - 1)/n'$, για $2 \leq n' \leq 8$. Για την κατασκευή αυτών των κωδίκων λάβαμε υπόψη μας μόνο μητρικούς συγκεραστικούς κωδικές μέγιστης ελεύθερης απόστασης (maximum free distance convolutional codes). Αξίζει να σημειωθεί πως η διαδικασία απαλοιφής μονοπατιών αφαιρεί ένα σύνολο κωδικών λέξεων από το μητρικό κώδικα [21], ενώ η διαδικασία απαλοιφής κωδικών bit διαγράφει bit από τις εναπομείναντες κωδικές λέξεις. Επομένως, είναι λογικό να περιμένει κανείς πως ένας μητρικός κώδικας με κωδικές λέξεις μεγάλου βάρους (Hamming) έχει αυξημένες πιθανότητες να παράγει καλούς κωδικές, δηλαδή οι κωδικές μέγιστης ελεύθερης απόστασης είναι καλοί υποψήφιοι σαν μητρικοί κωδικές των συγκεκριμένων κατασκευών. Οι ευέλικτοι κωδικές στους Πίνακες 4.1 και 4.2 βρέθηκαν λαμβάνοντας υπόψη όλους τους $(2, 1, 8)$ και $(3, 1, 8)$ κωδικές μέγιστης ελεύθερης απόστασης αντίστοιχα και παράγοντας τις αντίστοιχες κωδικές τους οικογένειες, όπως παρουσιάστηκε στην ενότητα 4.3. Ο $(4, 1, 8)$ μητρικός συγκεραστικός κώδικας που παράγει την κωδική οικογένεια στον Πίνακα 4.3 βρέθηκε όπως περιγράφεται παρακάτω. Συγκεντρώσαμε όλους τους $(3, 1, 8)$ συγκεραστικούς κωδικές που παράγουν καλές κωδικές οικογένειες. Στη συνέχεια προσθήσαμε ένα επιπλέον πολυώνυμο στους πολυωνυμικούς πίνακες γεννήτορες που τους παράγουν (για κάθε πίνακα λάβαμε υπόψη μας όλα τα δυνατά επιπλέον πολυώνυμα). Μ' αυτόν τον τρόπο καταλήξαμε με διάφορους $(4, 1, 8)$ συγκεραστικούς κωδικές, από τους οποίους κρατήσαμε μόνο τους κωδικές μέγιστης ελεύθερης απόστασης. Για κάθε έναν απ' αυτούς παράχθηκε η αντίστοιχη κωδική οικογένεια, όπως περιγράψαμε και στην ενότητα 4.3. Η συγκεκριμένη διαδικασία εξασφαλίζει ότι ο χώρος αναζήτησης αποτελείται από $(4, 1, 8)$ συγκεραστικούς κωδικές που παράγουν κωδικές οικογένειες τουλάχιστον τόσο καλές όσο αυτές που παράγονται από τους καλύτερους $(3, 1, 8)$ ευέλικτους μητρικούς συγκεραστικούς κωδικοποιητές.

Οι Πίνακες 4.1–4.3 περιέχουν όλους τους συναφείς κωδικές, καθώς και πληροφορίες για την trellis πολυπλοκότητα τους TC , τον αριθμό των συνενώσεων E_{pu} , τον αριθμό των καταστά-

Πίνακας 4.1. Οικογένεια κωδίκων που παράγεται από το μητρικό συγκεραστικό κωδικοποιητή

$$\mathbf{G}(D) = [1 + D^4 + D^5 + D^6 + D^8 \quad 1 + D + D^3 + D^5 + D^6 + D^7 + D^8]$$

k'/n'	m'	TC	E_{pu}	$ \Sigma_{pu} $	T_{pr}	p	$c(X)$	P	d_f	φάσματα $\frac{A_{d_f}, A_{d_f+1}, \dots}{I_{d_f}, I_{d_f+1}, \dots}$
1/2	8	1024	256	256	-	-	-	3	12	11,0,50,0 33,0,281,0
1/2	5	128	32	96	2	1	X^3	11	8	3,0,16,0 6,0,75,0
1/2	4	64	16	32	2	1	0	30	7	3,3,3,24 7,10,13,110
2/3	8	768	256	256	-	-	-	23	7*	3,9,50,190 11,46,324,1594
2/3	7	384	128	192	3	1	$X + X^2 + X^4 + X^5$	130	7*	6,28,90,328 25,178,780,3311
2/3	6	192	64	96	3	1	$X + X^2$	310	6	7,0,98,0 19,0,554,0
2/3	5	96	32	64	3	1	0	230	5*	4,14,54,179 20,70,392,1626
2/3	4	48	16	48	2	2	X	0203	4*	1,11,25,68 1,48,148,494
3/4	8	682.7	256	256	-	-	-	223	6*	10,77,303,1599 52,659,3265,21442
3/4	7	341.3	128	213.3	4	1	X^5	3201	6	38,0,1061,0 198,0,11619,0
3/4	6	170.7	64	85	4	1	0	3120	5*	11,44,183,964 54,291,1661,10727
3/4	4	42.7	16	48	2	3	X	022003	4	12,0,330,0 49,0,2749,0
4/5	8	640	256	256	-	-	-	2123	5*	7,49,351,2259 31,469,4205,34011
4/5	7	320	128	160	5	1	$X^2 + X^3$	12130	5*	11,113,665,4220 66,1183,8740,69094
4/5	6	160	64	96	3	2	$X + X^2$	220320	4*	2,35,202,1268 2,238,1832,15721
4/5	5	80	32	64	3	2	0	210103	4	23,88,652,3889 184,814,9101,62840
4/5	4	40	16	32	2	4	0	20201030	4	53,0,1822,0 312,0,20264,0
5/6	8	614	256	256	-	-	-	12213	5	19,187,1499,11809 168,2469,25174,242850
5/6	7	307	128	153.6	6	1	$X + X^2$	312210	4*	1,43,283,2167 3,323,3218,32930
5/6	5	76.8	32	96	2	5	X^3	1010011011	4	48,0,2973,0 310,0,40180,0
5/6	4	38.4	16	48	2	5	X	2020200222	3*	5,46,334,2443 11,351,3633,36151
6/7	8	597.3	256	256	-	-	-	122123	4*	2,48,427,4153 9,447,5954,76660
6/7	7	298.7	128	170.6	7	1	$X + X^2$	2121320	4*	3,98,859,7958 23,992,12440,153749
6/7	6	149.3	64	85.3	4	2	0	12202320	4	14,165,1464,13592 87,1625,21586,262086
6/7	5	74.7	32	64	3	3	0	210110301	4	159,0,12432,0 1591,0,223106,0
6/7	4	37.3	16	48	2	6	X	022020020203	3	13,113,955,7901 56,945,11700,127871
7/8	8	585.1	256	256	-	-	-	2212123	4	6,132,1289,13986 70,1842,24096,337514
7/8	7	292.6	128	146.2	8	1	0	12212310	4	16,230,2377,25643 98,2959,43186,592654
7/8	5	73.1	32	96	2	7	X^3	10100110100320	3*	7,74,830,8766 19,636,11260,165429
7/8	4	36.6	16	32	2	7	0	20101010202030	3	21,192,1848,17979 126,1966,27127,341931

* Αυτός ο κώδικας έχει μικρότερη d_f από τον αντίστοιχο καλύτερο κώδικα στη βιβλιογραφία του ίδιου ρυθμού και της ίδιας υπολογιστικής πολυπλοκότητας αποκωδικοποίησης.

Πίνακας 4.2. Οικογένεια κωδίκων που παράγεται από το μητρικό συγκεραστικό κωδικοποιητή $G(D) = [1+D^3+D^4+D^6+D^8 \quad 1+D+D^2+D^3+D^4+D^6+D^7+D^8 \quad 1+D^2+D^3+D^4+D^8]$

k'/n'	m'	TC	E_{pu}	$ \Sigma_{pu} $	T_{pr}	p	$c(X)$	P	d_f	φάσματα $\frac{A_{d_f}, A_{d_f+1}, \dots}{I_{d_f}, I_{d_f+1}, \dots}$
1/2	8	1024	256	256	-	-	-	6	11*	$\frac{1,7,13,29}{3,32,59,194}$
1/2	5	128	32	96	2	1	$X + X^3$	21	8	$\frac{2,0,20,0}{5,0,98,0}$
1/2	4	64	16	32	2	1	0	60	6*	$\frac{1,1,3,7}{2,3,10,27}$
2/3	8	768	256	256	-	-	-	13	8	$\frac{28,0,334,0}{143,0,2828,0}$
2/3	7	384	128	192	3	1	$X + X^4 + X^5$	320	7*	$\frac{10,30,113,451}{50,206,1004,4698}$
2/3	6	192	64	96	3	1	X^2	620	6	$\frac{3,15,42,169}{10,69,295,1390}$
2/3	5	96	32	64	3	1	0	206	6	$\frac{34,0,335,0}{170,0,2946,0}$
2/3	4	48	16	32	2	2	0	4030	5	$\frac{5,23,71,238}{15,104,457,1948}$
3/4	8	682.7	256	256	-	-	-	443	6*	$\frac{6,49,241,1233}{33,409,2595,16195}$
3/4	7	341.3	128	213.3	4	1	$X + X^2 + X^5$	4221	6	$\frac{15,82,384,2035}{88,643,3993,26035}$
3/4	6	170.7	64	106.7	4	1	X	1412	5*	$\frac{5,35,185,883}{15,218,1530,9415}$
3/4	5	85.3	32	96	2	3	$X + X^3$	200112	4*	$\frac{2,18,82,414}{2,85,564,3770}$
3/4	4	42.7	16	32	2	3	0	402050	4	$\frac{5,33,137,668}{18,184,1133,7002}$
4/5	8	640	256	256	-	-	-	2246	6	$\frac{80,0,3664,0}{701,0,54397,0}$
4/5	7	320	128	192	5	1	X^4	62420	6	$\frac{148,0,6171,0}{1253,0,89954,0}$
4/5	6	160	64	96	3	2	X^2	140130	4*	$\frac{1,23,150,952}{3,123,1330,11374}$
4/5	5	80	32	64	3	2	0	202502	4	$\frac{6,57,352,2187}{33,423,3702,29874}$
4/5	4	40	16	48	2	4	X	04040106	4	$\frac{30,130,842,5045}{160,1028,9336,69197}$
5/6	8	614	256	256	-	-	-	24416	5	$\frac{10,135,1041,8353}{55,1488,14854,152575}$
5/6	7	307	128	179.2	6	1	X^3	423402	5	$\frac{33,260,1956,15891}{256,2988,29835,302305}$
5/6	5	76.8	32	96	2	5	$X + X^3$	0101200142	4	$\frac{25,194,1389,10892}{124,1734,17757,177643}$
5/6	4	38.4	16	48	2	5	X	0440010141	4	$\frac{113,0,5759,0}{959,0,91620,0}$
6/7	8	597.3	256	256	-	-	-	221225	4*	$\frac{1,63,526,4870}{2,692,8169,97598}$
6/7	7	298.7	128	149.3	7	1	X	3121420	4*	$\frac{6,93,850,7919}{27,931,12314,148929}$
6/7	6	149.3	64	106.7	4	2	X	22104406	4	$\frac{13,207,1750,15942}{56,1944,24463,291724}$
6/7	5	74.7	32	64	3	3	0	202120242	4	$\frac{110,0,8896,0}{949,0,154240,0}$
6/7	4	37.3	16	48	2	6	X	044001040450	3	$\frac{6,84,678,5963}{16,709,8222,100125}$
7/8	8	585.1	256	256	-	-	-	2211116	4	$\frac{3,119,1194,12876}{31,1537,22143,304650}$
7/8	7	292.6	128	164.5	8	1	X	26212140	4	$\frac{8,175,1801,19107}{47,2148,30770,424382}$
7/8	5	73.1	32	96	2	7	$X + X^3$	01202001200130	3*	$\frac{4,105,989,10032}{18,1014,14661,198189}$
7/8	4	36.6	16	48	2	7	X	40100440040144	3	$\frac{12,154,1546,14954}{46,1562,22806,297211}$

* Αυτός ο κώδικας έχει μικρότερη d_f από τον αντίστοιχο καλύτερο κώδικα στη βιβλιογραφία του ίδιου ρυθμού και της ίδιας υπολογιστικής πολυπλοκότητας αποκωδικοποίησης.

Πίνακας 4.3. Οικογένεια κωδίκων που παράγεται από το μητρικό συγκεραστικό κωδικοποιητή $\mathbf{G}(D) = [1 + D^3 + D^4 + D^6 + D^8 \quad 1 + D + D^2 + D^3 + D^4 + D^6 + D^7 + D^8 \quad 1 + D^2 + D^3 + D^4 + D^8 \quad 1 + D + D^2 + D^3 + D^5 + D^7 + D^8]$

k'/n'	m'	TC	E_{pu}	$ \Sigma_{pu} $	T_{pr}	p	$c(X)$	P	d_f	φράσματα $\frac{A_{d_f}, A_{d_f+1} \dots}{I_{d_f}, I_{d_f+1} \dots}$	καλύτεροι κώδικες $d_f, \frac{A_{d_f}, A_{d_f+1}}{I_{d_f}, I_{d_f+1}}$
1/2	8	1024	256	256	-	-	-	3	12	$\frac{11,0,50,0}{33,0,281,0}$	$12^b, \frac{11,0}{33,0}$
1/2	6	256	64	192	2	1	$X^3 + X^5$	11	9*	$\frac{3,7,9,21}{9,38,49,132}$	$10^b, \frac{11,0}{36,0}$
1/2	5	128	32	96	2	1	X^3	14	8	$\frac{1,8,7,12}{2,36,32,62}$	$8^b, \frac{1,8}{2,36}$
1/2	4	64	16	32	2	1	0	90	7	$\frac{2,3,4,16}{4,12,20,72}$	$7^b, \frac{2,3}{4,12}$
2/3	8	768	256	256	-	-	-	26	8	$\frac{28,0,334,0}{143,0,2828,0}$	$8^{b,r}, \frac{9,58}{38,416}$
2/3	7	384	128	192	3	1	$X + X^2 + X^5$	1A0	7*	$\frac{10,39,107,376}{38,255,918,3722}$	$8^t, \frac{60,0}{314,0}$
2/3	6	192	64	96	3	1	X^2	190	6	$\frac{3,14,58,191}{9,67,394,1604}$	$6^b, \frac{1,17}{1,81}$
2/3	5	96	32	64	3	1	0	901	6	$\frac{16,0,210,0}{71,0,1702,0}$	$6^{t,r}, \frac{13,0}{43,0}$
2/3	4	48	16	32	2	2	0	8060	5	$\frac{5,23,71,238}{15,104,457,1948}$	$5^{t,r}, \frac{4,17}{8,69}$
3/4	8	682.7	256	256	-	-	-	886	6*	$\frac{6,49,241,1233}{33,409,2595,16195}$	$7^{c,r}, \frac{44,199}{-}$ $6^b, \frac{1,44}{1,312}$
3/4	7	341.3	128	213.3	4	1	$X + X^2 + X^5$	8442	6	$\frac{15,82,384,2035}{88,643,3993,26035}$	$6^b, \frac{10,86}{35,649}$
3/4	6	170.7	64	106.7	4	1	X	2824	5*	$\frac{5,35,185,883}{15,218,1530,9415}$	$6^t, \frac{45,109}{259,865}$
3/4	5	85.3	32	96	2	3	X^3	011041	5	$\frac{14,73,325,1580}{54,508,3173,18900}$	$5^t, \frac{13,67}{48,449}$ $5^r, \frac{13,64}{-}$
3/4	4	42.7	16	32	2	3	0	8040A0	4	$\frac{5,33,137,668}{18,184,1133,7002}$	$4^t, \frac{3,44}{6,296}$
4/5	8	640	256	256	-	-	-	448C	6	$\frac{80,0,3664,0}{701,0,54397,0}$	$6^b, \frac{55,254}{364,2355}$
4/5	7	320	128	192	5	1	X^4	C4840	6	$\frac{148,0,6171,0}{1253,0,89954,0}$	$6^{b,r}, \frac{134,0}{1015,0}$
4/5	6	160	64	96	3	2	X^2	280260	4*	$\frac{1,23,150,952}{3,123,1330,11374}$	$5^{t,r}, \frac{20,144}{89,1144}$
4/5	5	80	32	64	3	2	0	810424	4	$\frac{7,62,376,2379}{26,405,3956,32495}$	$4^t, \frac{3,55}{4,329}$
4/5	4	40	16	48	2	4	X	0808020C	4	$\frac{30,130,842,5045}{160,1028,9336,69197}$	$4^t, \frac{31,0}{118,0}$
5/6	8	614	256	256	-	-	-	4882C	5	$\frac{10,135,1041,8353}{55,1488,14854,152575}$	$5^b, \frac{11,148}{66,1667}$
5/6	7	307	128	179.2	6	1	X^3	846804	5	$\frac{33,260,1956,15891}{256,2988,29835,302305}$	$5^b, \frac{29,268}{202,2860}$
5/6	5	76.8	32	96	2	5	X^3	101010104	4	$\frac{19,171,1251,9573}{100,1592,17441,166331}$	$4^t, \frac{13,149}{51,1215}$
5/6	4	38.4	16	32	2	5	0	8080108030	4	$\frac{111,0,5688,0}{762,0,77562,0}$	$4^t, \frac{111,0}{742,0}$
6/7	8	597.3	256	256	-	-	-	882816	5*	$\frac{56,472,4285,41619}{628,7021,82665,985662}$	$6^{c,r}, \frac{700,0}{-}$ $5^b, \frac{39,376}{314,4886}$
6/7	7	298.7	128	149.3	7	1	X	2882C10	4*	$\frac{3,96,875,7990}{12,1042,13104,155939}$	$5^{c,r}, \frac{73,729}{-}$ $4^b, \frac{4,119}{10,1341}$
6/7	6	149.3	64	85.3	4	2	0	28202890	4	$\frac{13,174,1559,14226}{53,1579,21920,265318}$	$4^b, \frac{7,158}{36,1426}$
6/7	5	74.7	32	64	3	3	0	101810A01	4	$\frac{44,316,3332,28180}{281,3439,52978,570805}$	$4^b, \frac{39,350}{223,3954}$
6/7	4	37.3	16	48	2	6	X	0880020808A0	3	$\frac{6,84,678,5963}{16,709,8222,100125}$	$3^b, \frac{6,77}{11,592}$
7/8	8	585.1	256	256	-	-	-	1211843	4	$\frac{4,109,1005,10733}{18,1283,17281,239638}$	$4^b, \frac{4,115}{11,1531}$
7/8	7	292.6	128	146.2	8	1	0	18448C80	4	$\frac{8,176,1777,19081}{43,2071,29935,419358}$	$4^b, \frac{13,228}{94,2681}$
7/8	5	73.1	32	96	2	7	X^3	10040101040441	4	$\frac{229,0,23631,0}{2439,0,482885,0}$	$4^b, \frac{144,0}{1104,0}$
7/8	4	36.6	16	32	2	7	0	808020801010A0	3	$\frac{13,148,1487,14565}{46,1416,21588,286959}$	$3^t, \frac{12,154}{36,1505}$

^b Κώδικας που δημοσιεύεται στο [15]. ^t Κώδικας που δημοσιεύεται στο [29]. ^c Κώδικας που δημοσιεύεται στο [27]. ^r Κώδικας που δημοσιεύεται στο [28].

* Αυτός ο κώδικας έχει μικρότερη d_f από τον αντίστοιχο καλύτερο κώδικα στη βιβλιογραφία του ίδιου ρυθμού και της ίδιας υπολογιστικής πολυπλοκότητας αποκωδικοποίησης.

σεων $|\Sigma_{pu}|$ (όλα κανονικοποιημένα με το k'), το ρυθμό τους, το m' , το T_{pr} , το p , το πολυώνυμο $c(X)$, τον πίνακα απαλοιφής P , την ελεύθερη απόσταση d_f , και τους τέσσερις πρώτους όρους των φασμάτων τους. Οι πίνακες απαλοιφής στον Πίνακα 4.1 έχουν γραφεί όπως εξηγείται παρακάτω. Η δυαδική αναπαράσταση ενός συμβόλου είναι η αντίστοιχη στήλη του πίνακα απαλοιφής. Το πιο αριστερό σύμβολο αντιστοιχεί στην πιο αριστερή στήλη του πίνακα απαλοιφής. Το ίδιο ισχύει και για τους πίνακες απαλοιφής που περιέχονται στους πίνακες 4.2 και 4.3, με τη διαφορά ότι έχουν γραφεί σε οκταδική και δεκαεξαδική μορφή αντίστοιχα. Επίσης, η τελευταία στήλη του Πίνακα 4.3 περιέχει την ελεύθερη απόσταση και τους δύο πρώτους όρους των φασμάτων των καλύτερων κωδίκων με τον ίδιο ρυθμό και υπολογιστική πολυπλοκότητα αποκωδικοποίησης που δημοσιεύονται στα [15, 29, 27, 28]. Για τους κώδικες που παρουσιάζονται στα [27] και [28] δεν δίνεται συγκεκριμένος πίνακας γεννήτορας, άλλα ο πίνακας ισοτιμίας τους, οπότε παραθέτουμε μόνο το φάσμα βαρών των κωδικών λέξεων.

Παρόλο που θεωρούμε ως μέτρο της πολυπλοκότητας την υπολογιστική πολυπλοκότητα του αλγόριθμου του Viterbi, ανάλογα με την υλοποίηση του αποκωδικοποιητή, ο συνολικός αριθμός των καταστάσεων του trellis module μπορεί να επηρεάζει τη συνολική πολυπλοκότητα αποκωδικοποίησης. Γι'αυτό το λόγο, ο αριθμός των καταστάσεων (κανονικοποιημένος με το k') του trellis module κάθε κώδικα περιέχεται στους Πίνακες 4.1–4.3. Εάν ένας κώδικας συγκεκριμένου ρυθμού και τιμής του m' δεν περιλαμβάνεται, τότε ή το trellis module του έχει περισσότερες καταστάσεις από το αντίστοιχο του κώδικα με ίδιο ρυθμό και μεγαλύτερη τιμή του m' , ή ο καλύτερος κώδικας που βρέθηκε είναι χειρότερος από αυτόν του ίδιου ρυθμού και μικρότερου m' .

Στο [11], αναπτύσσονται οικογένειες PCC μεταβλητού ρυθμού από $(2, 1)$ μητρικούς συγκεραστικούς κώδικες, για διάφορες τιμές της μνήμης. Παρομοίως, οικογένειες PCC μεταβλητού ρυθμού από $(4, 1)$ μητρικούς συγκεραστικούς κώδικες παρουσιάζονται στο [12]. Επίσης, στα [15, 29, 27, 28] δημοσιεύονται οι καλύτεροι γνωστοί συγκεραστικοί κώδικες οι οποίοι έχουν υπολογιστική πολυπλοκότητα αποκωδικοποίησης ίση με αυτή των PCC (για την ακρίβεια οι κώδικες στο [15] είναι PCC). Εφόσον όλοι οι κώδικες που παρουσιάζονται σ' αυτό το κεφάλαιο έχουν υπολογιστική πολυπλοκότητα αποκωδικοποίησης ίση με αυτή των αντίστοιχων (n', k', m') PCC, θα τους συγκρίνουμε με τους κώδικες στα παραπάνω άρθρα. 16 από τους 30 κώδικες του Πίνακα 4.1, 21 κώδικες από τους 31 του Πίνακα 4.2 και 25 από τους 32 κώδικες του Πίνακα 4.3 έχουν ελεύθερη απόσταση ίση με αυτή των καλύτερων κωδίκων που δημοσιεύονται στα [15, 29, 27, 28]. Επίσης, πολλοί από τους κώδικες στον Πίνακα 4.3 έχουν συγκρίσιμα φάσματα (για την ίδια ελάχιστη απόσταση) με τους καλύτερους γνωστούς κώδικες στα προαναφερθέντα άρθρα (σε δύο περιπτώσεις μάλιστα έχουν καλύτερα φάσματα). Ακόμη, πολλοί κώδικες στον Πίνακα 4.1 έχουν παρόμοια φάσματα με τους κώδικες στο [11], κάποιιοι έχουν καλύτερα, ενώ πολύ λίγοι έχουν χειρότερα. Παρόμοια συμπεράσματα προκύπτουν και από τη σύγκριση των κωδίκων στον Πίνακα 4.3 με τους κώδικες στο [12]. Συνοψίζοντας, οι προ-

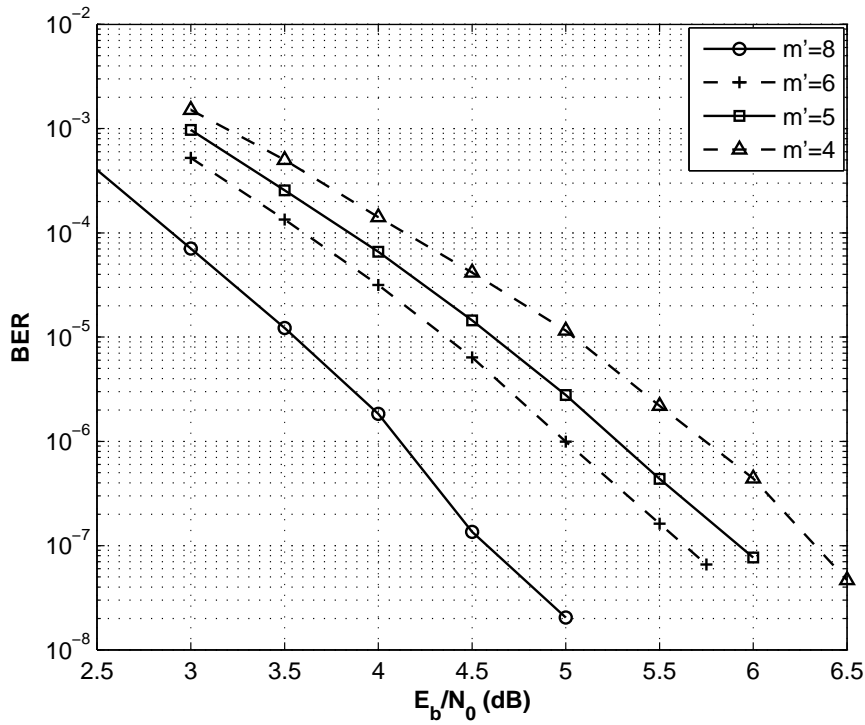
τεινόμενες κωδικές οικογένειες έχουν το ίδιο καλή απόδοση με τις αντίστοιχες οικογένειες PCC μεταβλητού ρυθμού, και επιπροσθέτως παρέχουν τη δυνατότητα μεταβολής της υπολογιστικής πολυπλοκότητας της αποκωδικοποίησης. Αξίζει επίσης να σημειωθεί πως ενώ στο [11] και στο [12] διαλέγονται δύο και τέσσερα καινούρια πολυώνυμα αντίστοιχα για κάθε τιμή της μνήμης, μόνο δύο πολυώνυμα παράγουν όλους τους κώδικες στον Πίνακα 4.1 και μόνο τέσσερα στον πίνακα 4.3. Παρατηρούμε λοιπόν ότι υπάρχουν καλοί ευέλικτοι συγκεραστικοί κωδικοποιητές που μπορούν να παράγουν καλές κωδικές οικογένειες με τη χρήση των τεχνικών της απαλοιφής μονοπατιών του διαγράμματος trellis και της απαλοιφής των κωδικών bit.

4.5 Προσομοιώσεις

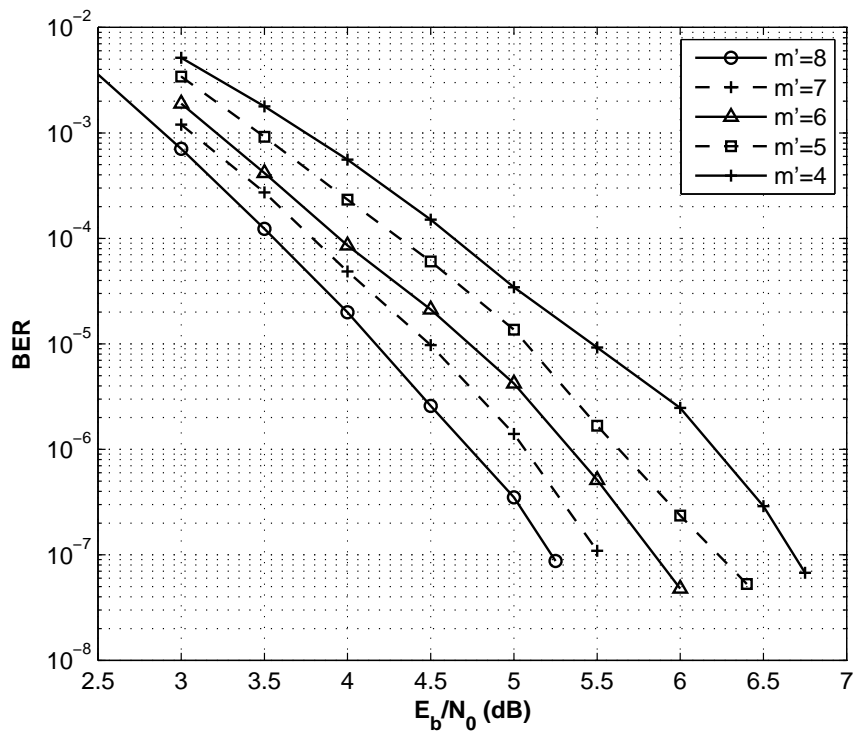
Στα πλαίσια της μελέτης των ευέλικτων συγκεραστικών κωδικών έχουμε προσομοιώσει την απόδοση (BER) των κωδικών του Πίνακα 4.3, για το AWGN κανάλι, χρησιμοποιώντας διαμόρφωση BPSK. Τα αποτελέσματα εμφανίζονται στα Σχήματα 4.5–4.8.

Όπως φαίνεται στο Σχ. 4.5(α'), μπορεί να επιτευχθεί πιθανότητα σφάλματος 10^{-7} για διάφορες τιμές του E_b/N_0 , οι οποίες κυμαίνονται από 4.6dB μέχρι 6.3dB, μεταβάλλοντας μόνο την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης, και διατηρώντας το ρυθμό σταθερό και ίσο με $1/2$. Αντίστοιχη συμπεριφορά έχουν και οι κώδικες των υπόλοιπων ρυθμών στα Σχ. 4.6, 4.7 και 4.8. Επίσης, παρόμοιες παρατηρήσεις μπορούν να γίνουν εάν διατηρήσουμε το m' σταθερό και μεταβάλλουμε μόνο τον κωδικό ρυθμό.

Σε αντίθεση με τους κλασσικούς συγκεραστικούς κώδικες μεταβλητού ρυθμού [13, 16, 17] (ένας μπτρικός κώδικας δηλαδή και ένα σύνολο πινάκων απαλοιφής κωδικών bit), με τους οποίους “ανταλλάσσεται” κέρδος κωδικοποίησης για εύρος ζώνης και αντιστρόφως, οι προτεινόμενες κατασκευές προσθέτουν μία ακόμη “διάσταση”, την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης, οδηγώντας έτσι σε σχήματα κωδικοποίησης που διαχειρίζονται αποδοτικότερα τους πόρους του συστήματος. Για παράδειγμα, θεωρήστε την οικογένεια κωδικών στον Πίνακα 4.3. Υποθέστε ότι ο λόγος σήματος προς θόρυβο (E_b/N_0) είναι 7.1dB, και πως χρησιμοποιείται ο κώδικας ρυθμού $3/4$ και $m' = 4$ ($TC = 42.7$, $E_{pu} = 16$, ανά bit πληροφορίας), επιτυγχάνοντας έτσι πιθανότητα σφάλματος 10^{-7} (Σχ. 4.6(α')). Υποθέστε επίσης πως ο E_b/N_0 μειώνεται κατά 0.8dB. Υπάρχουν πολλές επιλογές που μπορούν να γίνουν κατά την επόμενη μετάδοση (κωδικής ακολουθίας), έτσι ώστε να αντισταθμιστεί η μείωση του E_b/N_0 και να παραμείνει σταθερή η πιθανότητα σφάλματος. Μία επιλογή είναι η χρήση του κώδικα ρυθμού $1/2$ και $m' = 4$ ($TC = 64$, $E_{pu} = 16$, ανά bit πληροφορίας) (Σχ. 4.5(α')), να μειώσουμε δηλαδή το ρυθμό και να διατηρήσουμε την υπολογιστική πολυπλοκότητα σε παρόμοια επίπεδα. Μία άλλη επιλογή είναι η χρήση του κώδικα με ρυθμό $3/4$ και $m' = 6$ (Σχ. 4.6(α')) ($TC = 170.7$, $E_{pu} = 64$, ανά bit πληροφορίας), να διατηρήσουμε δηλαδή σταθερό το ρυθμό και να αυξήσουμε την υπολογιστική πολυπλοκότητα. Τέλος, θα μπορούσε να χρησιμοποιηθεί ο κώδικας ρυθμού

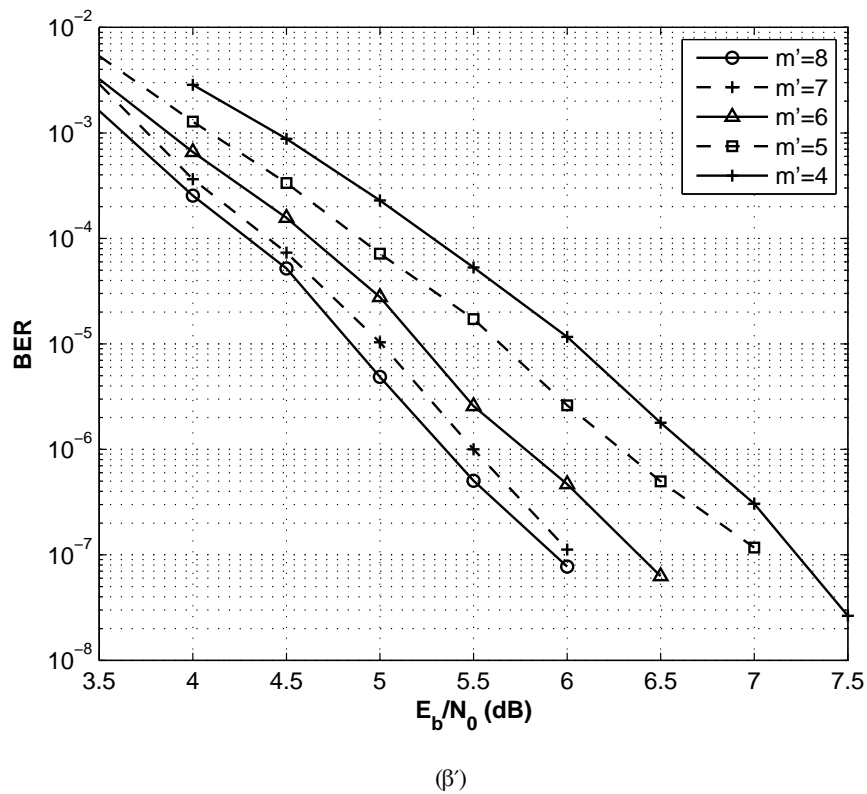
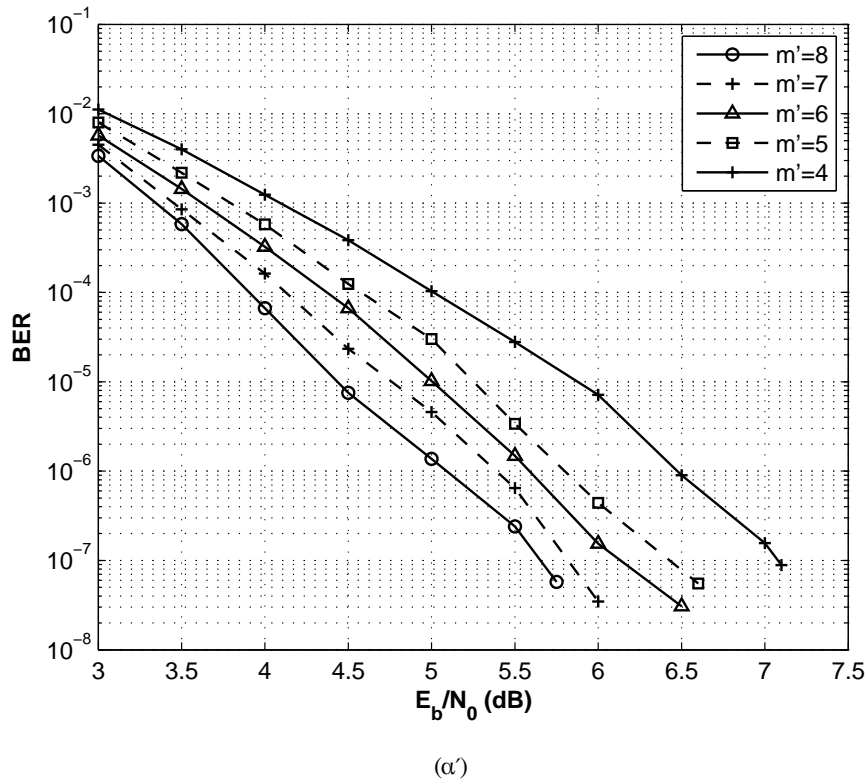


(α')

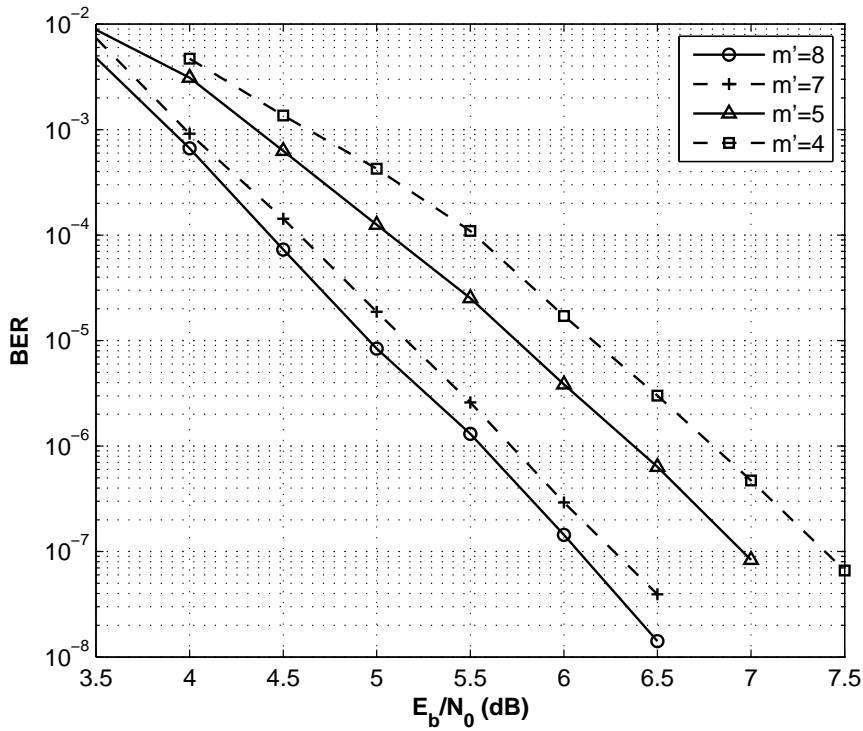


(β')

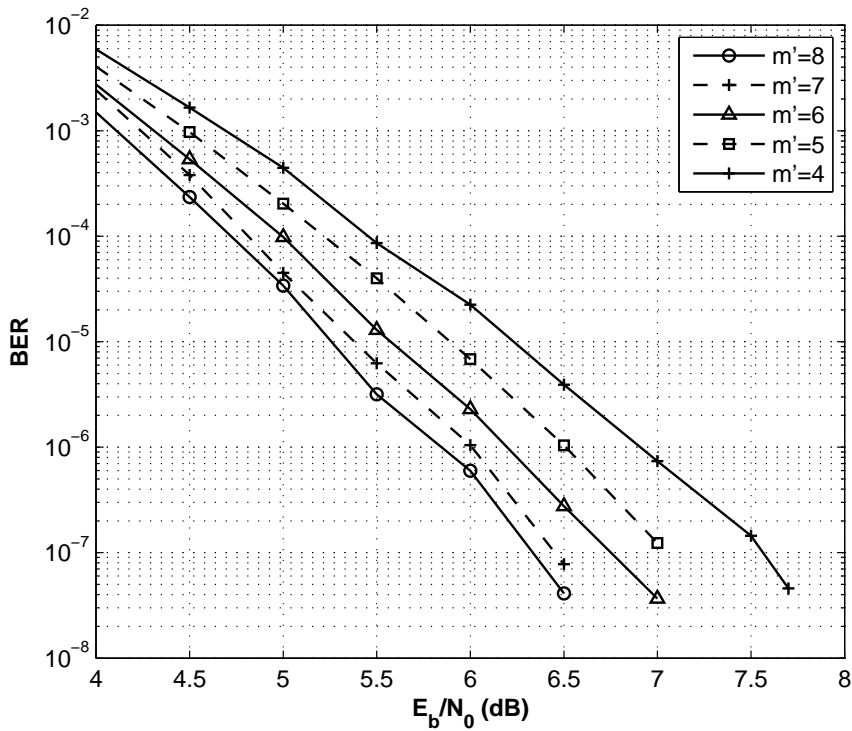
Σχήμα 4.5. Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμούς (α') $1/2$ και (β') $2/3$.



Σχήμα 4.6. Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμούς (α') $3/4$ και (β') $4/5$.

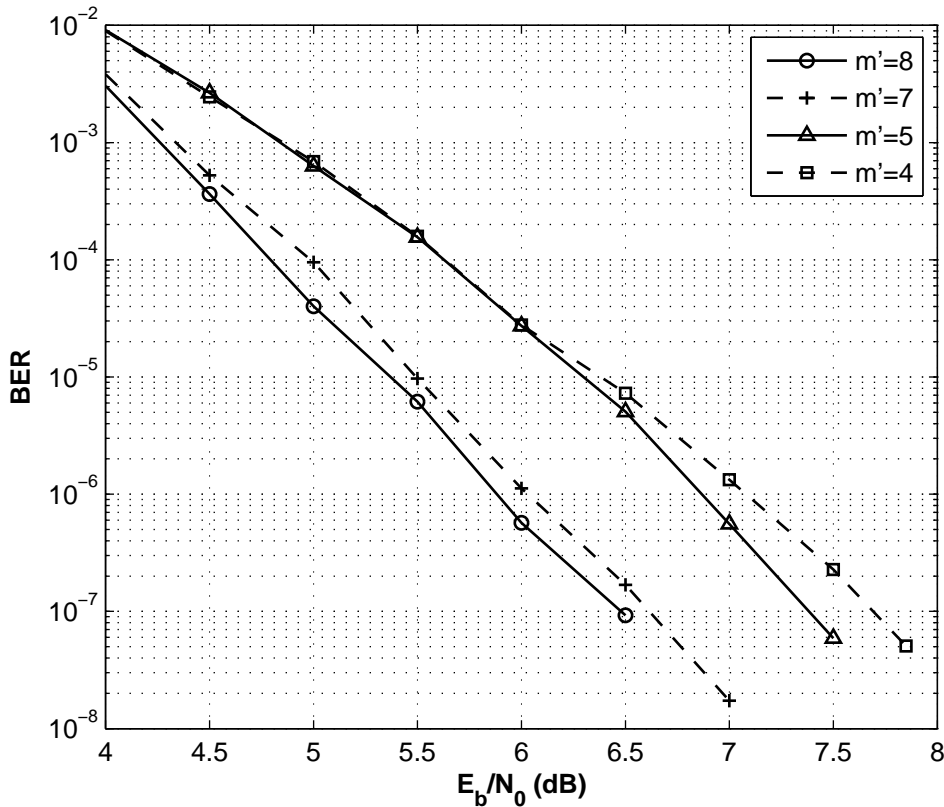


(α')



(β')

Σχήμα 4.7. Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμούς (α') 5/6 και (β') 6/7.



Σχήμα 4.8. Προσομοιώσεις του BER διαφόρων κωδίκων του Πίνακα 4.3, για διάφορες τιμές του m' και ρυθμό 7/8.

2/3 και $m' = 5$ ($TC = 96$, $E_{pu} = 32$, ανά bit πληροφορίας) (Σχ. 4.5(β')), οδηγώντας σε μικρή μείωση του ρυθμού και μικρή αύξηση της πολυπλοκότητας.

Σαν ένα άλλο παράδειγμα, υποθέστε ότι $E_b/N_0 = 5.25\text{dB}$, και πως χρησιμοποιείται ο κώδικας ρυθμού 2/3 και $m' = 8$ ($TC = 768$, $E_{pu} = 256$, ανά bit πληροφορίας), επιτυγχάνοντας έτσι πιθανότητα σφάλματος 10^{-7} (Σχ. 4.5(β')). Υποθέστε πως ο E_b/N_0 αυξάνεται κατά 1.5dB. Σ' αυτήν την περίπτωση δεν χρειάζεται τόσο ισχυρός κώδικας. Συνεπώς, κατά την επόμενη μετάδοση μπορεί να αυξηθεί ο ρυθμός, ή να μειωθεί η πολυπλοκότητα. Για παράδειγμα, μπορεί να χρησιμοποιηθεί ο κώδικας ρυθμού 2/3 και $m' = 4$ ($TC = 48$, $E_{pu} = 16$, ανά bit πληροφορίας) (Σχ. 4.5(β')), οδηγώντας σε εξαιρετική μείωση της υπολογιστικής πολυπλοκότητας. Εναλλακτικά, μπορεί να χρησιμοποιηθεί ο κώδικας ρυθμού 6/7 και $m' = 6$ ($TC = 149.3$, $E_{pu} = 64$, ανά bit πληροφορίας) (Σχ. 4.7(β')), οδηγώντας σε μεγάλη αύξηση του ρυθμού και μεγάλη μείωση της υπολογιστικής πολυπλοκότητας.

Ο Πίνακας 4.4 αποτελεί προϊόν λεπτομερέστερης επεξεργασίας των αποτελεσμάτων της προσομοίωσης. Για κάθε τιμή του E_b/N_0 περιέχει (για κάθε κωδικό ρυθμό) τον κώδικα με τη μικρότερη υπολογιστική πολυπλοκότητα αποκωδικοποίησης, ο οποίος επιτυγχάνει $P_b(E) \leq \phi$, για διάφορες τιμές του ϕ . Στην περίπτωση που δύο κώδικες έχουν παρεμφερή τιμή πολυπλο-

κότητας και διαφορετικό ρυθμό, περιέχεται μόνο ο κώδικας με το μεγαλύτερο ρυθμό. Επί της ουσίας, ο Πίνακας 4.4 δείχνει τον πιο αποδοτικό τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί η οικογένεια συγκεραστικών κωδίκων που περιέχεται στον Πίνακα 4.3.

Οι επιπλέον δυνατότητες που παρέχουν οι προτεινόμενοι κώδικες σ' ένα σύστημα επικοινωνιών είναι εμφανείς και στον πίνακα 4.4. Για παράδειγμα, εάν $E_b/N_0 = 5\text{dB}$ και επιθυμούμε πιθανότητα σφάλματος $P_b(E) \leq 10^{-5}$, μπορούν να χρησιμοποιηθούν τέσσερις διαφορετικοί κώδικες από την οικογένεια στον πίνακα 4.3, ανάλογα με τον αν θέλουμε μειωμένη υπολογιστική πολυπλοκότητα ή αυξημένο ρυθμό.

Πίνακας 4.4. Λεπτομερής επεξεργασία των προσομοιώσεων της κωδικής οικογένειας στον Πίνακα 4.3

E_b/N_0 (dB)	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8
$P_b \leq 10^{-3}$	(2, 1, 8)	(2, 1, 5) (3, 2, 8)	(2, 1, 4) (3, 2, 5)	(3, 2, 4) (4, 3, 5) (5, 4, 6) (6, 5, 7)	(5, 4, 4) (7, 6, 5) (8, 7, 7)	(8, 7, 4)	(8, 7, 4)	(8, 7, 4)	(8, 7, 4)	(8, 7, 4)	(8, 7, 4)	(8, 7, 4)
$P_b \leq 10^{-4}$		(2, 1, 8)	(2, 1, 8)	(2, 1, 5) (3, 2, 6) (4, 3, 8)	(2, 1, 4) (3, 2, 5) (4, 3, 6) (5, 4, 7) (6, 5, 8)	(4, 3, 4) (5, 4, 5) (7, 6, 6) (8, 7, 7)	(7, 6, 4) (8, 7, 7)	(8, 7, 4)				
$P_b \leq 10^{-5}$				(2, 1, 8)	(2, 1, 6) (3, 2, 7) (4, 3, 8)	(2, 1, 5) (3, 2, 6) (5, 4, 7) (6, 5, 8)	(3, 2, 4) (4, 3, 5) (5, 4, 6) (8, 7, 7)	(4, 3, 4) (7, 6, 5) (8, 7, 7)	(8, 7, 4)	(8, 7, 4)	(8, 7, 4)	(8, 7, 4)
$P_b \leq 10^{-6}$					(2, 1, 8)	(2, 1, 6) (3, 2, 8)	(2, 1, 4) (3, 2, 6) (5, 4, 7)	(2, 1, 4) (4, 3, 5) (5, 4, 6) (7, 6, 7) (8, 7, 8)	(4, 3, 4) (7, 6, 5) (8, 7, 7)	(7, 6, 4) (8, 7, 5)	(8, 7, 4)	(8, 7, 4)
$P_b \leq 10^{-7}$						(2, 1, 8)	(3, 2, 7)	(2, 1, 5) (3, 2, 6) (4, 3, 7) (5, 4, 8)	(2, 1, 4) (4, 3, 5) (5, 4, 6) (7, 6, 7) (8, 7, 8)	(3, 2, 4) (6, 5, 5) (7, 6, 6) (8, 7, 7)	(6, 5, 4) (8, 7, 5)	(8, 7, 4)

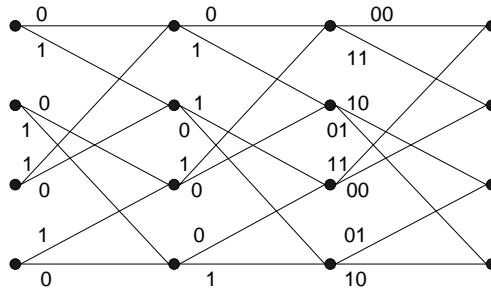
4.A Χαρακτηριστικά των υπό Εξέταση Κωδίκων

Τα trellis modules που κατασκευάζονται με τη μέθοδο που παρουσιάστηκε στην ενότητα 4.3 σχηματίζουν δύο κατηγορίες. Η πρώτη αποτελείται από τα trellis modules για τα οποία ο βαθμός \hat{d} του αντίστοιχου πολυώνυμου $c(X)$ προκύπτει από τη σχέση (4.36) (δηλαδή ικανοποιείται η σχέση (4.28)), ενώ η δεύτερη κατηγορία αποτελείται από τα trellis modules για τα οποία ο βαθμός \hat{d} του αντίστοιχου πολυώνυμου $c(X)$ προκύπτει από τη σχέση (4.37) (δηλαδή ικανοποιείται η σχέση (4.29)). Όσον αφορά στην πρώτη κατηγορία, από τη σχέση (4.10) προκύπτει ότι η διάσταση του χώρου καταστάσεων σε κάθε βάθος του trellis module \mathcal{M}_{pu} είναι σταθερή και ίση με m' . Επίσης, από τις σχέσεις (4.10) και (4.15) προκύπτει πως δύο ακμές ξεκινούν από κάθε κατάσταση του \mathcal{M}_{pu} σε βάθος $j \neq \sigma T_{pr}$, $1 \leq \sigma \leq p$, ενώ μόνο μία ακμή ξεκινάει από κάθε κατάσταση σε βάθος $l = \sigma T_{pr}$, $1 \leq \sigma \leq p$. Παρομοίως, για τα trellis modules της δεύτερης κατηγορίας προκύπτει πως η διάσταση του χώρου καταστάσεων σε βάθος $l \neq \sigma T_{pr}$, $1 \leq \sigma \leq p$ είναι ίση με m' , ενώ δύο ακμές ξεκινούν από κάθε κατάσταση. Η διάσταση του χώρου καταστάσεων σε βάθος $l = \sigma T_{pr}$, $1 \leq \sigma \leq p$ είναι ίση με $m' + 1$, ενώ μόνο μία ακμή ξεκινάει από κάθε κατάσταση. Και στις δύο περιπτώσεις, αν συνδυάσουμε τα τμήματα trellis $(\sigma T_{pr} - 1)$ και σT_{pr} του \mathcal{M}_{pu} σε ένα τμήμα trellis, για κάθε $1 \leq \sigma \leq p$, το αποτέλεσμα είναι ένα *information bit-oriented* trellis module (κάθε τμήμα trellis δηλαδή αντιστοιχεί σε ένα bit πληροφορίας) k' τμημάτων, του οποίου η διάσταση του χώρου καταστάσεων παραμένει σταθερή σε κάθε βάθος και ίση με m' . Ακριβώς δηλαδή όπως τα trellis modules των PCC και των κωδίκων που παρουσιάζονται στα [27] και [28]. Για την ακρίβεια, οι κώδικες που κατασκευάζονται σ' αυτό το κεφάλαιο ανήκουν στην κατηγορία των κωδίκων που παρουσιάζονται στο [28] (όπως άλλωστε και οι PCC, αλλά και οι κώδικες στο [27]).

Θεωρείστε για παράδειγμα το μητρικό συγκεραστικό κώδικα του παραδείγματος 4.5 και τον παραγόμενο κώδικα \mathcal{C}_2 . Το trellis module του \mathcal{C}_2 φαίνεται στο Σχ. 4.3 και ανήκει στη δεύτερη κατηγορία. Εάν διαγράψουμε τις “αχρησιμοποίητες” καταστάσεις (τις καταστάσεις από τις οποίες ούτε αναχωρεί αλλά ούτε και φθάνει κάποια ακμή) και συνδυάσουμε τα τμήματα trellis 3 και 4 σε ένα ενιαίο τμήμα trellis (εάν δηλαδή διαγράψουμε όλες τις καταστάσεις σε βάθος 4 και αντικαταστήσουμε κάθε αλληλουχία μίας ακμής από το τμήμα 3 και μίας ακμής από το τμήμα 4 με μία ενιαία ακμή [27]), τότε το trellis module του Σχ. 4.3 μετασχηματίζεται σε αυτό του Σχ. 4.9. Θεωρείστε σαν ένα άλλο παράδειγμα τον κώδικα \mathcal{C}_4 που προκύπτει από την τριάδα $(p, T_{pr}, \hat{d}) = (2, 3, 3)$ (σ' αυτήν την περίπτωση $k' = 4$ και $m' = 2$), σε συνδυασμό με τον πίνακα απαλοιφής κωδικών bit

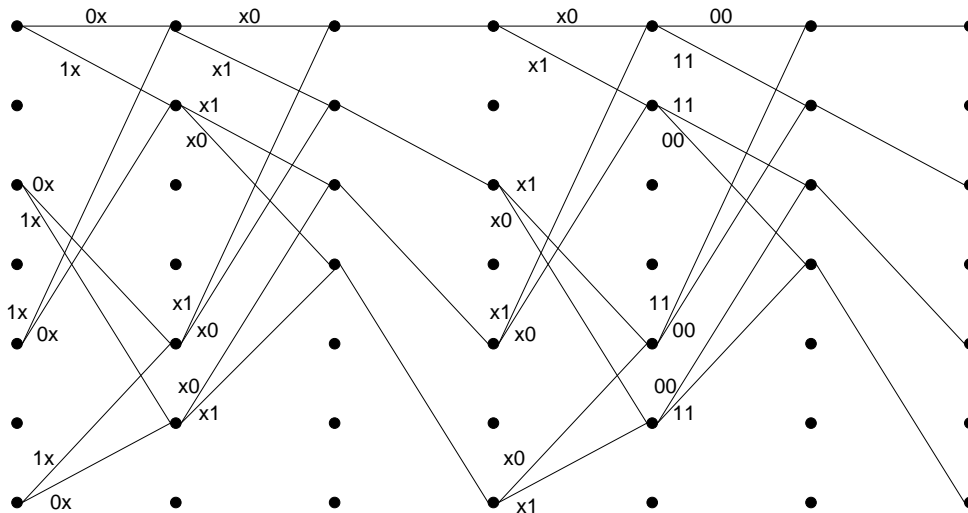
$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Το αντίστοιχο trellis module απεικονίζεται στο Σχ. 4.10 και ανήκει στην πρώτη κατηγορία.



Σχήμα 4.9. Information bit oriented trellis module που αναπαριστά τον κώδικα C_2 .

Ακολουθώντας παρόμοια διαδικασία με παραπάνω, αν διαγράψουμε τις αχρησιμοποίητες καταστάσεις και συνδυάσουμε τα τμήματα trellis 2, 3 και 5, 6 σε δύο ενιαία τμήματα, τότε το trellis module του Σχ. 4.10 μετασχηματίζεται σε αυτό του Σχ. 4.11. Το τελευταίο είναι ένα information bit-oriented trellis module του οποίου η διάσταση του χώρου καταστάσεων είναι σταθερή και ίση με 2, και αντιστοιχεί σ' έναν $(5, 4)$ συγκεραστικό κώδικα. Τα trellis modules στα Σχ. 4.9

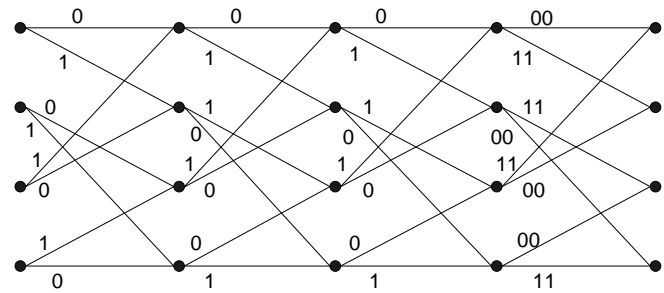


Σχήμα 4.10. Trellis module που αναπαριστά τον κώδικα C_4

και 4.11 έχουν σταθερή δομή (ο τρόπος με τον οποίο συνδέονται οι καταστάσεις μεταξύ τους δεν αλλάζει από τμήμα trellis σε τμήμα trellis), γι'αυτό σύμφωνα με το [28] (και ύστερα από κατάλληλη αρίθμηση των καταστάσεων) αντιστοιχούν σε PCC. Πράγματι, ο κώδικας C_2 (και το trellis module στο Σχ. 4.9) μπορεί να παραχθεί από τον $(2, 1, 2)$ μητρικό κώδικα με πίνακα γεννήτορα $\mathbf{G}(D) = [1 + D + D^2 \quad 1 + D^2]$, σε συνδυασμό με τον πίνακα απαλοιφής κωδικών bit

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

Αντίστοιχα, ο κώδικας C_4 (και το trellis module στο Σχ. 4.11) μπορεί να παραχθεί από τον $(3, 1, 2)$ μητρικό κώδικα με πίνακα γεννήτορα $\mathbf{G}(D) = [1 + D^2 \quad 1 + D + D^2 \quad 1 + D + D^2]$, σε



Σχήμα 4.11. Information bit-oriented trellis module που αναπαριστά τον κώδικα C_4 .

συνδυασμό με τον πίνακα απαλοιφής κωδικών bit

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

Κάθε κώδικας λοιπόν που κατασκευάζεται με τη μέθοδο που παρουσιάστηκε στην ενότητα 4.3 μπορεί να περιγραφεί από ένα information bit-oriented trellis module k' τμημάτων, του οποίου η διάσταση του χώρου καταστάσεων είναι σταθερή και ίση με m' . Ισοδύναμα, μπορεί να παραχθεί από έναν k' -περιοδικό χρονικά μεταβαλλόμενο συγκεραστικό κωδικοποιητή μνήμης m' του οποίου η είσοδος κάθε χρονική στιγμή αποτελείται από ένα bit, ενώ η έξοδος αποτελείται από n_i κωδικά bit, $1 \leq i \leq k'$.

5. ΑΝΑΔΡΟΜΙΚΟΙ ΕΥΕΛΙΚΤΟΙ ΣΥΓΚΕΡΑΣΤΙΚΟΙ ΚΩΔΙΚΟΠΟΙΗΤΕΣ ΓΙΑ ΤΟΥΡΜΠΟ ΚΩΔΙΚΕΣ

Στο προηγούμενο κεφάλαιο, οι τεχνικές της απαλοιφής μονοπατιών του διαγράμματος trellis και της απαλοιφής κωδικών bit οδήγησαν στην κατασκευή ευέλικτων συγκεραστικών κωδικοποιητών, οι οποίοι μπορούν να μεταβάλλουν εκτός από τον κωδικό τους ρυθμό και την υπολογιστική πολυπλοκότητα αποκωδικοποίησης. Εδώ, επεκτείνοντας αυτή την ανάλυση αναπτύσσουμε ευέλικτους συγκεραστικούς κωδικοποιητές κατάλληλους για τούρμπο κώδικες, οι οποίοι οδηγούν σε ευέλικτα τούρμπο σχήματα κωδικοποίησης [90]. Θα εξετάσουμε την εφαρμογή των αποτελεσμάτων του κεφαλαίου 4 στην περίπτωση των αναδρομικών $(n, 1, m)$ μπτρικών κωδικοποιητών, καθώς και τις δυνατότητες της χρήσης αυτής της μεθόδου στους τούρμπο κώδικες (turbo codes). Αποτελέσματα προσομοιώσεων δείχνουν ότι με τη συγκεκριμένη μέθοδο, και για διάφορες τιμές του Eb/N_0 , μείωση της υπολογιστικής πολυπλοκότητας της αποκωδικοποίησης μπορεί να οδηγήσει σε αύξηση της απόδοσης.

5.1 Εισαγωγή στους Κώδικες Τούρμπο

Ο Shannon το 1948, στο θεμελιώδες άρθρο του [1], προσδιόρισε τα όρια ενός συστήματος επικοινωνιών. Απέδειξε ότι υπάρχουν κώδικες διόρθωσης σφαλμάτων οι οποίοι μπορούν να παρέχουν οσοδήποτε υψηλή αξιοπιστία κατά τη μετάδοση, δοθέντος ότι οι ρυθμοί της πληροφορίας είναι μικρότεροι της χωρητικότητας του καναλιού. Παρά τις προσπάθειες της επιστημονικής κοινότητας, μέχρι το 1993 οι καλύτεροι κώδικες απέιχαν τουλάχιστον 2db απ' το όριο του Shannon. Το 1993, οι Berrou, Glavieux και Thitimajshima επιτυγχάνουν ένα από τα μεγαλύτερα επιτεύγματα στην ιστορία της κωδικοποίησης καναλιού, την ανακάλυψη των κωδικών τούρμπο [34]. Οι τελευταίοι αποτελούν επανάσταση, αφού η απόδοση τους απέχει 0.5dB από το όριο του Shannon. Αυτό επιτυγχάνεται με κόστος μία "λογική" και αποδεκτή πολυπλοκότητα αποκωδικοποίησης, χρησιμοποιώντας καινοτόμες ιδέες, όπως π.χ. αυτήν της επαναληπτικής αποκωδικοποίησης. Εδώ να τονίσουμε πως η κατασκευή ενός τόσο ισχυρού κώδικα δεν είναι δύσκολη. Εάν διαλέξουμε τυχαία έναν μπλοκ κώδικα με κωδικές λέξεις εξαιρετικά μεγάλου μήκους, ή έναν συγκεραστικό κωδικοποιητή με τεράστια μνήμη, το πιο πιθανό είναι πως η απόδοση του θα πλησιάζει το όριο του Shannon. Όμως, θα ήταν αδύνατη η αποκωδικοποίηση του, λαμβάνοντας υπόψη τις δυνατότητες των σημερινών συστημάτων. Έτσι, συνήθως λεγότανε πως: *όλοι οι κώδικες είναι καλοί, εκτός από αυτούς που μπορούμε να αποκωδικοποιήσουμε!* [91]. Γι' αυτό άλλωστε, ο Berrou είχε δηλώσει πως πρώτα εφηύραν τον τούρμπο αποκωδικοποιητή, και έπειτα σχεδίασαν έναν κατάλληλο κωδικοποιητή.

5.1.1 Ο Τούρμπο Κωδικοποιητής

Ένας τούρμπο κωδικοποιητής (στη μορφή που πρωτοπαρουσιάστηκε) αποτελείται από δύο $(2, 1, m)$ αναδρομικούς συστηματικούς συγκεραστικούς κωδικοποιητές, οι οποίοι συνενώνονται παράλληλα, μέσω ενός συμπλέκτη (interleaver) μήκους N . Το μπλοκ διάγραμμα ενός τούρμπο κωδικοποιητή φαίνεται στο Σχ. 5.1(α'). Η διαδικασία της κωδικοποίησης πραγματοποιείται ως εξής: μία ακολουθία εισόδου \mathbf{u} με μήκος N εισάγεται στον πρώτο συγκεραστικό κωδικοποιητή, ο οποίος παράγει τις ακολουθίες $\mathbf{v}^{(0)}$ (η οποία είναι η ίδια η \mathbf{u}) και $\mathbf{v}^{(1)}$. Στη συνέχεια, η \mathbf{u} εισάγεται στον συμπλέκτη όπου και παράγεται μία μετάθεση της, η \mathbf{u}' , με την οποία τροφοδοτείται ο δεύτερος κωδικοποιητής. Η συστηματική έξοδος του τελευταίου δεν χρησιμοποιείται, κι έτσι παράγεται μόνο η ακολουθία $\mathbf{v}^{(2)}$. Τουλάχιστον ένας από τους δύο συγκεραστικούς κωδικοποιητές πρέπει να τερματιστεί (δηλαδή να επιστρέψει στην κατάσταση 0), οπότε στην πιο απλή περίπτωση για παράδειγμα η \mathbf{u} αποτελείται από $N - m$ bit πληροφορίας και m bit τα οποία θα τερματίσουν τον πρώτο κωδικοποιητή. Έτσι, ο κωδικός ρυθμός είναι ίσος με $((N - m)/3N) \simeq 1/3$ (Το N είναι πολύ μεγαλύτερο του m). Συνήθως οι δύο περιεχόμενοι κωδικοποιητές είναι πανομοιότυποι, και τότε ο τούρμπο κώδικας λέγεται *συμμετρικός*. Στο Σχ. 5.1(β) απεικονίζεται ένας τούρμπο κώδικας που χρησιμοποιεί τον συγκεραστικό κωδικοποιητή με πίνακα γεννήτορα τον

$$\mathbf{G}(D) = \begin{bmatrix} 1 & \frac{1 + D + D^3}{1 + D^2 + D^3} \end{bmatrix}. \quad (5.1)$$

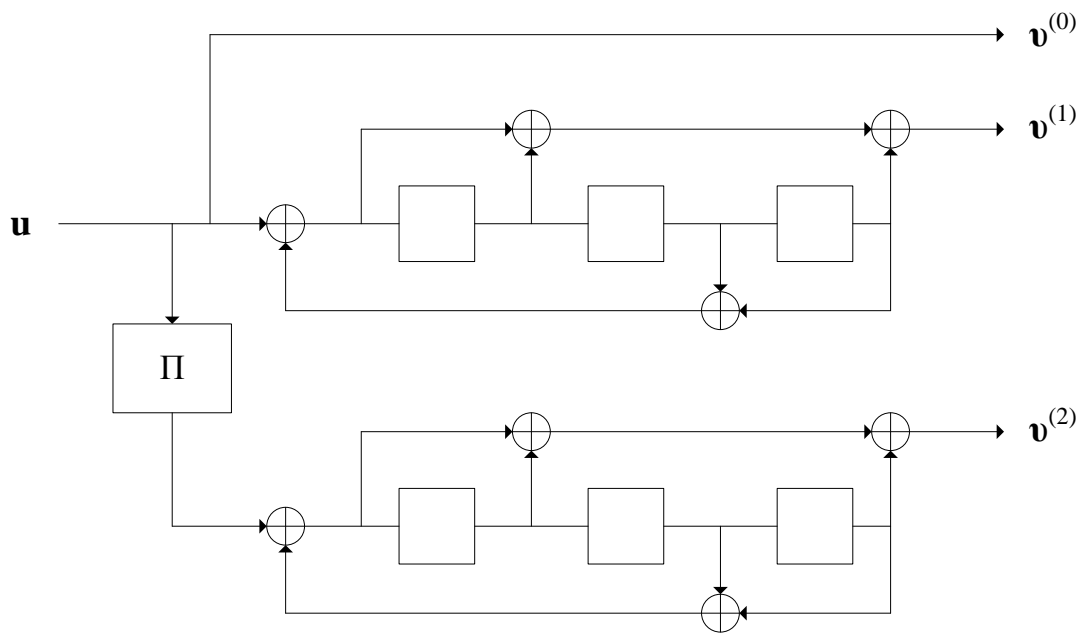
Σαν περιεχόμενος κωδικοποιητής μπορεί να χρησιμοποιηθεί οποιοσδήποτε (n, k, m) συστηματικός αναδρομικός κωδικοποιητής, οδηγώντας σε έναν κώδικα τούρμπο ρυθμού $k/(2n - k)$. Η χρήση μη αναδρομικού περιεχόμενου συγκεραστικού κωδικοποιητή είναι απαγορευτική, καθώς σ' αυτήν την περίπτωση δεν υπάρχει το λεγόμενο κέρδος *συμπλέκτη* (interleaver gain). Το βάρος μιας κωδικής λέξης που παράγεται από έναν αναδρομικό κωδικοποιητή, εξαρτάται (εκτός από το βάρος της αντίστοιχης λέξης πληροφορίας) και από τη θέση που έχουν οι μονάδες στη λέξη πληροφορίας. Έτσι, μια λέξη πληροφορίας η οποία οδηγεί τον πρώτο κωδικοποιητή σε κωδική λέξη χαμηλού βάρους, μπορεί εφόσον μεσολαβήσει ο συμπλέκτης να οδηγήσει τον δεύτερο κωδικοποιητή σε κωδική λέξη υψηλού βάρους. Κατ' αυτόν τον τρόπο, η πλειοψηφία των λέξεων του τούρμπο κώδικα εμφανίζει υψηλά βάρη (spectral thinning). Αντίθετα, το βάρος μιας κωδικής λέξης ενός μη αναδρομικού κώδικα εξαρτάται κυρίως από το βάρος της λέξης πληροφορίας (εδώ η θέση των μονάδων δεν παίζει σημαντικό ρόλο). Συνεπώς, σ' αυτήν την περίπτωση η χρήση του συμπλέκτη δεν προκαλεί το φαινόμενο που περιγράψαμε παραπάνω.

5.1.2 Ο Τούρμπο Αποκωδικοποιητής

Η ιδέα να αποκωδικοποιήσουμε τον κώδικα τούρμπο (ο οποίος είναι γραμμικός μπλοκ κώδικας), εφαρμόζοντας έναν αλγόριθμο μεγίστης πιθανοφάνειας όπως ο αλγόριθμος του Viterbi



(α')



(β')

Σχήμα 5.1. Ο τούρμπο κωδικοποιητής: (α') μπλοκ διάγραμμα και (β') ένα συγκεκριμένο παράδειγμα.

στο διάγραμμα trellis που περιγράφει τον κώδικα δεν είναι πραγματοποιήσιμη, καθώς το πλήθος των καταστάσεων σε κάποιο βάθος του τελευταίου μπορεί να φθάσει τιμές της τάξης του 2^{N+2m} [91]. Η καινοτόμος λύση που δόθηκε στο [34] ήταν η εξής: χρησιμοποιούνται οι αποκωδικοποιητές των περιεχόμενων συγκεραστικών κωδίκων, καθένας εκ των οποίων εκτιμάει την ακολουθία πληροφορίας βασιζόμενος στο τμήμα της κωδικής λέξης που του αντιστοιχεί (στη συστηματική ακολουθία δηλαδή και στην αντίστοιχη ακολουθία ισοτιμίας). Επιπροσθέτως, κάθε αποκωδικοποιητής εκμεταλλεύεται την πρόοδο του άλλου, ανακυκλώνοντας μία μετρική αξιοπιστίας που αφορά τα bit πληροφορίας, η οποία διαρκώς βελτιώνεται. Κατ' αυτόν τον τρόπο βέβαια, η αποκωδικοποίηση δεν είναι βέλτιστη, δηλαδή δεν αποτελεί αποκωδικοποίηση μεγίστης πιθανοφάνειας. Όπως όμως διαπιστώθηκε (εμπειρικά), αφενός μεν οδηγεί σε πολύ χαμηλή πιθανότητα σφάλματος, αφετέρου δε προσεγγίζει την αποκωδικοποίηση μεγίστης πιθανοφάνειας σε ικανοποιητικό βαθμό, ειδικά όσο αυξάνεται ο σηματοθορυβικός λόγος [92, 93, 94].

Θεωρείστε ότι για κάθε συγκεραστικό κωδικοποιητή χρησιμοποιείται ένας a posteriori SISO (Soft Input – Soft Output) αποκωδικοποιητής (BCJR, Log-Map, Max-Log-Map). Έστω $\mathbf{r}^{(0)}$, $\mathbf{r}^{(1)}$ και $\mathbf{r}^{(2)}$ οι ληφθείσες ακολουθίες που αντιστοιχούν στις κωδικές ακολουθίες $\mathbf{v}^{(0)}$, $\mathbf{v}^{(1)}$ και $\mathbf{v}^{(2)}$, ενώ με $\mathbf{r}_{\neq t}^{(0)}$ συμβολίζουμε την ακολουθία που προκύπτει εάν από την $\mathbf{r}^{(0)}$ αφαιρέσουμε το σύμβολο $r_t^{(0)}$. Ο SISO αποκωδικοποιητής 1, υπολογίζει για κάθε bit πληροφορίας u_t το λογάριθμο του λόγου πιθανοφάνειας (Log Likelihood Ratio-LLR)

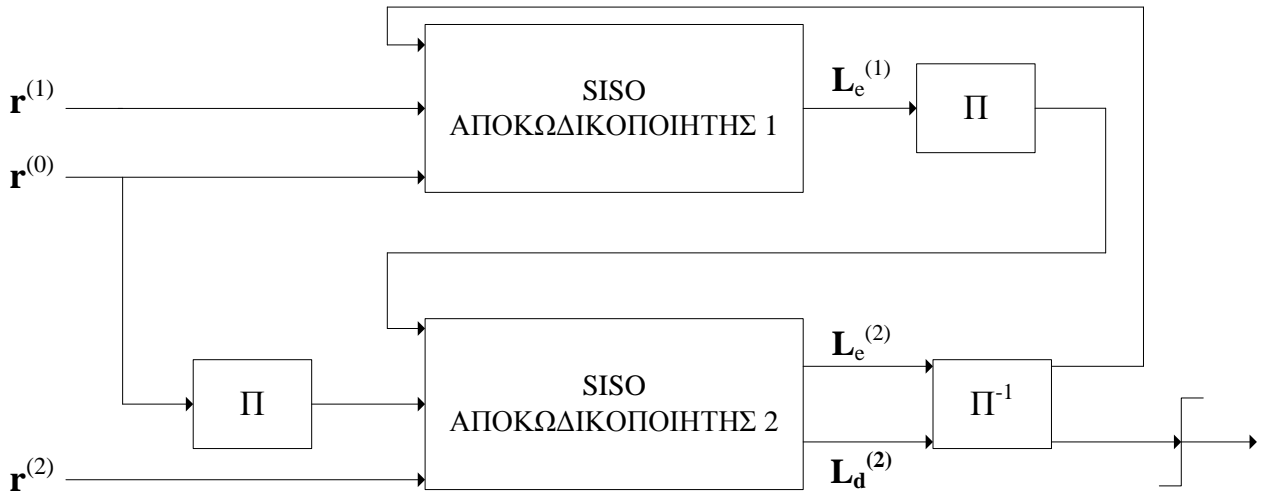
$$L(u_t|\mathbf{r}^{(0)}, \mathbf{r}^{(1)}) = \ln \frac{P(u_t = 1|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})}{P(u_t = 0|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})}. \quad (5.2)$$

Διαχωρίζοντας την εξάρτηση από το $r_t^{(0)}$, και ύστερα από πράξεις, η σχέση 5.2 γίνεται [48]

$$\begin{aligned} L(u_t|\mathbf{r}^{(0)}, \mathbf{r}^{(1)}) &= \ln \frac{P(r_t^{(0)}|u_t = 1)P(\mathbf{r}_{\neq t}^{(0)}, \mathbf{r}^{(1)}|u_t = 1)P(u_t = 1)}{P(r_t^{(0)}|u_t = 0)P(\mathbf{r}_{\neq t}^{(0)}, \mathbf{r}^{(1)}|u_t = 0)P(u_t = 0)} \\ &= L(r_t^{(0)}|u_t) + L(\mathbf{r}_{\neq t}^{(0)}, \mathbf{r}^{(1)}|u_t) + L(u_t). \end{aligned} \quad (5.3)$$

Όπως βλέπουμε, ο $L(u_t|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})$ αποτελείται από τρεις προσθετέους, τον $L(r_t^{(0)}|u_t)$ που εξαρτάται από το $r_t^{(0)}$, τον $L(\mathbf{r}_{\neq t}^{(0)}, \mathbf{r}^{(1)}|u_t)$, ο οποίος αποτελεί έμμεση πληροφορία για το u_t από την ακολουθία ισοτιμίας του πρώτου κωδικοποιητή και από την ακολουθία $\mathbf{r}_{\neq t}^{(0)}$, και την a priori πληροφορία $L(u_t)$. Χάρην συντομίας θα συμβολίζουμε τους $L(u_t|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})$, $L(\mathbf{r}_{\neq t}^{(0)}, \mathbf{r}^{(1)}|u_t)$ και $L(u_t)$, με $L_d^{(1)}(u_t)$, $L_e^{(1)}(u_t)$ και $L_a^{(1)}(u_t)$ αντίστοιχα.

Κατά την έναρξη της αποκωδικοποίησης, η a priori πληροφορία $L_a^{(1)}(u_t)$ στον SISO αποκωδικοποιητή 1 είναι ίση με το μηδέν. Αφού υπολογιστούν οι $L_d^{(1)}(u_t)$, στη συνέχεια χρησιμοποιώντας τη σχέση (5.3) υπολογίζονται οι $L_e^{(1)}(u_t)$. Οι τελευταίοι, αποτελούν την πληροφορία που μεταφέρεται από τον SISO αποκωδικοποιητή 1 στον SISO αποκωδικοποιητή 2, ο οποίος θα την



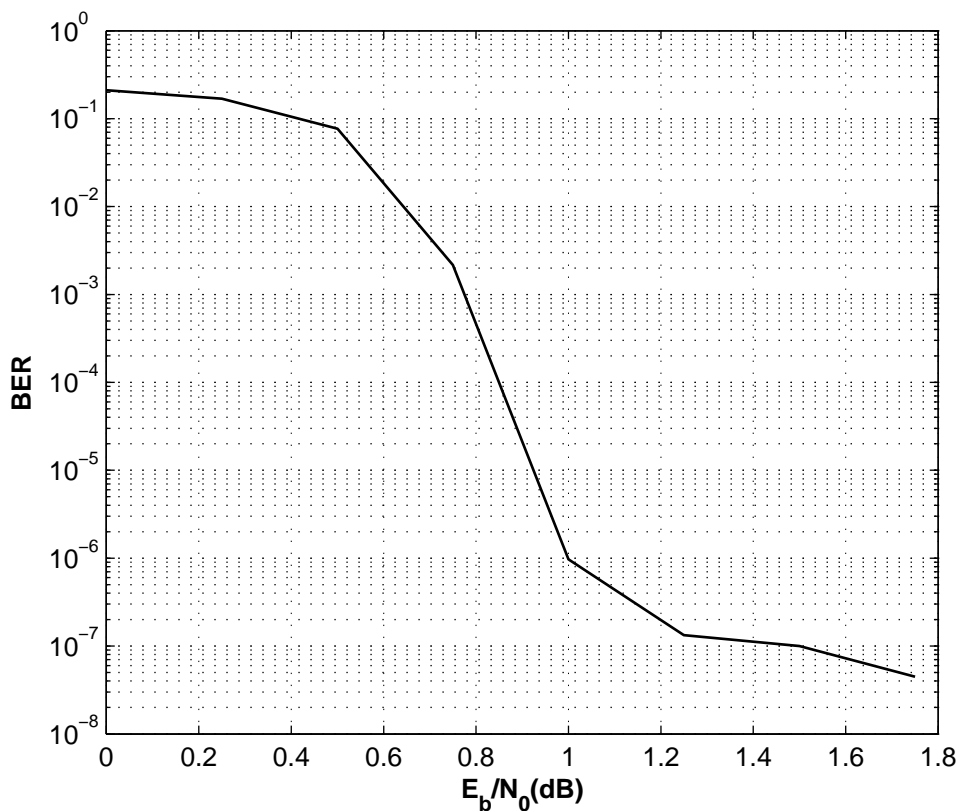
Σχήμα 5.2. Μπλοκ διάγραμμα του τούρμπο αποκωδικοποιητή.

χρησιμοποιήσει ως a priori πληροφορία. Ο $L_e^{(1)}(u_t)$ ονομάζεται *εξωτερική πληροφορία* (extrinsic information). Η παραπάνω διαδικασία επαναλαμβάνεται και στον SISO αποκωδικοποιητή 2, ο οποίος με τη σειρά του παράγει την εξωτερική πληροφορία $L_e^{(2)}(u_t)$. Ό,τι περιγράψαμε μέχρι στιγμής αποτελεί μία, και συγκεκριμένα την πρώτη, επανάληψη του τούρμπο αποκωδικοποιητή. Η δεύτερη επανάληψη είναι πανομοιότυπη με την πρώτη, με τη διαφορά ο αποκωδικοποιητής 1 πλέον χρησιμοποιεί ως a priori πληροφορία τους $L_e^{(2)}(u_t)$ που υπολογίστηκαν από τον αποκωδικοποιητή 2 κατά τη διάρκεια της προηγούμενης επανάληψης. Η μεταφορά πληροφοριών από τον πρώτο κωδικοποιητή στον δεύτερο γίνεται διαμέσου του συμπλέκτη Π που χρησιμοποιήθηκε στον τούρμπο κωδικοποιητή, ενώ για την μεταφορά πληροφοριών κατά την αντίθετη κατεύθυνση χρησιμοποιείται ο αντίστροφος συμπλέκτης Π^{-1} . Έπειτα από έναν επαρκή αριθμό επαναλήψεων, ο δεύτερος αποκωδικοποιητής εκτιμάει την τιμή των bit πληροφορίας από το πρόσημο των $L_d^{(2)}(u_t)$. Το μπλοκ διάγραμμα του τούρμπο αποκωδικοποιητή απεικονίζεται στο Σχ.5.2. Οι αλγόριθμοι που χρησιμοποιούνται για τον υπολογισμό των $L_d^{(1)}(u_t)$ και $L_d^{(2)}(u_t)$ είναι ο BCJR και οι προσεγγίσεις του (Log-MAP, Max-Log-Map), οι οποίοι χρησιμοποιούν το διάγραμμα trellis των περιεχόμενων συγκεραστικών κωδίκων.

Είναι σημαντικό να παρατηρήσουμε πως η εξωτερική πληροφορία που μεταφέρει ο ένας αποκωδικοποιητής στον άλλον δεν είναι τυχαία. Ο αποκωδικοποιητής 1 παρέχει στον 2 πληροφορία για τα u_t , που προκύπτει από την ληφθείσα ακολουθία $\mathbf{r}_{\neq t}^{(0)}$, αλλά και από την ακολουθία $\mathbf{r}^{(1)}$ (στην οποία δεν έχει πρόσβαση ο αποκωδικοποιητής 2), και αντιστρόφως.

5.1.3 Σχεδίαση Τούρμπο Κωδίκων

Στο Σχ. 5.3 βλέπουμε την καμπύλη απόδοσης ενός τούρμπο κώδικα, για το AWGN κανάλι, χρησιμοποιώντας διαμόρφωση BPSK. Μπορούμε να διακρίνουμε τρεις περιοχές. Η πρώτη (≤ 0.5 dB) ονομάζεται περιοχή *μη σύγκλισης* (non convergence region), διότι ο επαναληπτικός αποκωδικοποιητής δε συγκλίνει σε χαμηλή πιθανότητα σφάλματος. Η δεύτερη (0.5dB–1 dB) είναι η περιοχή *καταρράκτη* (waterfall region), όπου ο ρυθμός μείωσης της πιθανότητας σφάλματος είναι πολύ μεγάλος, και η καμπύλη απόδοσης μοιάζει με καταρράκτη. Η τρίτη (≥ 1 dB) είναι η περιοχή *δαπέδου σφάλματος* (error floor), όπου η πιθανότητα σφάλματος μειώνεται με πολύ μικρότερο ρυθμό. Η περιοχή καταρράκτη εξαρτάται κυρίως από το μήκος του συμπλέκτη, ενώ



Σχήμα 5.3. Η μορφή της καμπύλης απόδοσης ενός τούρμπο κώδικα.

η περιοχή δαπέδου σφάλματος εξαρτάται από τα βάρη των κωδικών λέξεων, και κατ' επέκτασιν από τα βάρη των κωδικών λέξεων των περιεχόμενων συγκεραστικών κωδίκων.

Εύκολα γίνεται αντιληπτό ότι η σχεδίαση – βελτιστοποίηση ενός τούρμπο κώδικα, αντιμετωπίζοντας τα δομικά του στοιχεία ως ένα ενιαίο σύνολο είναι αδύνατη. Αφενός, το πλήθος των συνδυασμών όλων των δυνατών πολυπλεκτών και συγκεραστικών κωδικοποιητών είναι τεράστιο, κι αφετέρου οι διαδικασίες εύρεσης μεγεθών όπως η ελάχιστη απόσταση για τόσο πολύπλοκα συστήματα είναι εξαιρετικά χρονοβόρες [92, 35, 36]. Παρακάτω θα δούμε δύο τε-

χνικές οι οποίες χρησιμοποιούνται προκειμένου να προσδιοριστούν τα χαρακτηριστικά του περιεχόμενου συγκεραστικού κωδικοποιητή, τα οποία επηρεάζουν την απόδοση του τούρμπο κώδικα. Έτσι, η κατασκευή ενός τούρμπο κώδικα χωρίζεται σε δύο φάσεις, στην επιλογή του συγκεραστικού κωδικοποιητή και στη συνέχεια στην επιλογή του συμπλέκτη [35].

Η πρώτη τεχνική [35, 36] βασίζεται σε φράγματα της απόδοσης της αποκωδικοποίησης μεγίστης πιθανοφάνειας, και οδηγεί σε σχεδιαστικούς κανόνες για τους περιεχόμενους κωδικοποιητές, κυρίως για μεσαίες και χαμηλές τιμές της πιθανότητας σφάλματος (κυρίως δηλαδή στην περιοχή δαπέδου σφάλματος). Επί της ουσίας, εισάγοντας την έννοια του *ομοιόμορφου συμπλέκτη* (uniform interleaver), η οποία βασίζεται στην πιθανολογική ανάλυση του συνόλου όλων των πολυπλεκτών, υπολογίζεται η μέση κατανομή βαρών των κωδικών λέξεων των τούρμπο κωδικών που περιέχουν συγκεκριμένο συγκεραστικό κωδικοποιητή, χρησιμοποιώντας την κατανομή βαρών των λέξεων που αντιστοιχεί στον κωδικοποιητή. Ο ομοιόμορφος συμπλέκτης είναι ένα πιθανολογικό στοιχείο που αντιστοιχίζει μία συγκεκριμένη ακολουθία εισόδου μήκους N και βάρους w σε όλες τις διακριτές $\binom{N}{w}$ μεταθέσεις της με την ίδια πιθανότητα

$$\frac{1}{\binom{N}{w}}. \quad (5.4)$$

Ορίζοντας την *υπό συνθήκη συνάρτηση απαρίθμησης βάρους* (conditional weight enumerator function-CWEF) του τούρμπο κώδικα ως

$$A^{C_T}(w, Z) = \sum_j A_{w,j}^{C_T} Z^j \quad (5.5)$$

όπου Z είναι μία ψευδομεταβλητή και $A_{w,j}^{C_T}$ είναι το πλήθος των κωδικών λέξεων των οποίων το τμήμα των bit ισότητας έχει βάρος j κι οι οποίες παράγονται από λέξεις πληροφορίας βάρους w , ένα άνω φράγμα της πιθανότητας σφάλματος bit του τούρμπο κώδικα για το AWGN κανάλι και για αποκωδικοποίηση μεγίστης πιθανοφάνειας δίνεται από τη σχέση

$$P_b(E) \leq \sum_w \frac{w}{N} W^w A^{C_T}(w, Z) |_{W=Z=e^{-RE_b/N_0}}. \quad (5.6)$$

Με R συμβολίζεται ο κωδικός ρυθμός. Για αναδρομικό περιεχόμενο κωδικοποιητή και για αρκετά μεγάλο μήκος συμπλέκτη N , από τη σχέση (5.6), και ύστερα από πράξεις (χρησιμοποιείται η $A^{C_T}(w, Z)$ και ο ομοιόμορφος συμπλέκτης) [35, 36] προκύπτει (για μεγάλο μήκος του συμπλέκτη) το φράγμα

$$P_b(E) \leq \sum_{i=1}^{\lfloor \frac{N}{2} \rfloor} i \binom{2i}{i} N^{-1} W^{2i} [A(2, Z, 1)]^{2i} |_{W=Z=e^{-RE_b/N_0}}. \quad (5.7)$$

όπου n

$$A(2, Z, 1) = \sum_j A_{2+j,2} Z^j \quad (5.8)$$

απαριθμεί τις ακολουθίες του περιεχόμενου συγκεραστικού κώδικα που παράγονται από λέξεις πληροφορίας με βάρος δύο, οι οποίες αντιστοιχούν σε μονοπάτια που αναχωρούν από την κατάσταση μηδέν και επιστρέφουν σε αυτή μόνο μια φορά. $A_{2+j,2}$ είναι το πλήθος των κωδικών λέξεων που παράγονται από λέξεις βάρους 2, και των οποίων το βάρος του τμήματος ισοτιμίας είναι ίσο με j (προφανώς το βάρος όλης της κωδικής λέξης είναι ίσο με $2 + j$). Όπως βλέπουμε, η απόδοση του τούρμπο κώδικα επηρεάζεται από το μήκος του συμπλέκτη. Συγκεκριμένα, όσο αυξάνεται το N μειώνεται η πιθανότητα σφάλματος, κι αυτό το φαινόμενο το αποκαλούμε *κέρδος του συμπλέκτη* (interleaver gain). Με αντίστοιχο τρόπο μπορεί ναδειχθεί ότι το τελευταίο απουσιάζει, εάν χρησιμοποιήσουμε μη αναδρομικό περιεχόμενο συγκεραστικό κωδικοποιητή.

Ο σημαντικότερος όρος της $A(2, Z, 1)$ που επηρεάζει την πιθανότητα σφάλματος, είναι ο πρώτος. Δηλαδή, ο όρος που αντιστοιχεί στην ακολουθία ισοτιμίας ελαχίστου βάρους z_{min} , που προκύπτει από μια λέξη βάρους 2. Έτσι λοιπόν, σχεδιάζοντας έναν τούρμπο κώδικα για μέτριες και χαμηλές τιμές της πιθανότητας σφάλματος, θα πρέπει να επιλέξουμε έναν αναδρομικό περιεχόμενο συγκεραστικό κωδικοποιητή, με όσο το δυνατόν μεγαλύτερη τιμή της $d_2 = 2 + z_{min}$, και όσο το δυνατόν λιγότερα μονοπάτια που παράγονται από λέξεις βάρους 2 κι έχουν βάρος d_2 .

Γενικότερα θα συμβολίζουμε με d_w την ελάχιστη απόσταση μιας κωδικής λέξης που προκύπτει από λέξη πληροφορίας βάρους w , και με $A_{d_w,w}$ το σύνολο των κωδικών λέξεων που παράγονται από λέξεις βάρους w και έχουν βάρος d_w . Τα d_w και $A_{d_w,w}$, όπως κι η απόδοση του τούρμπο κώδικα, εξαρτώνται από τη χρήση συγκεκριμένου συγκεραστικού κωδικοποιητή. Δύο διαφορετικοί κωδικοποιητές που παράγουν τον ίδιο συγκεραστικό κώδικα μπορεί να έχουν τελείως διαφορετική απόδοση όταν χρησιμοποιηθούν σ' έναν τούρμπο κώδικα.

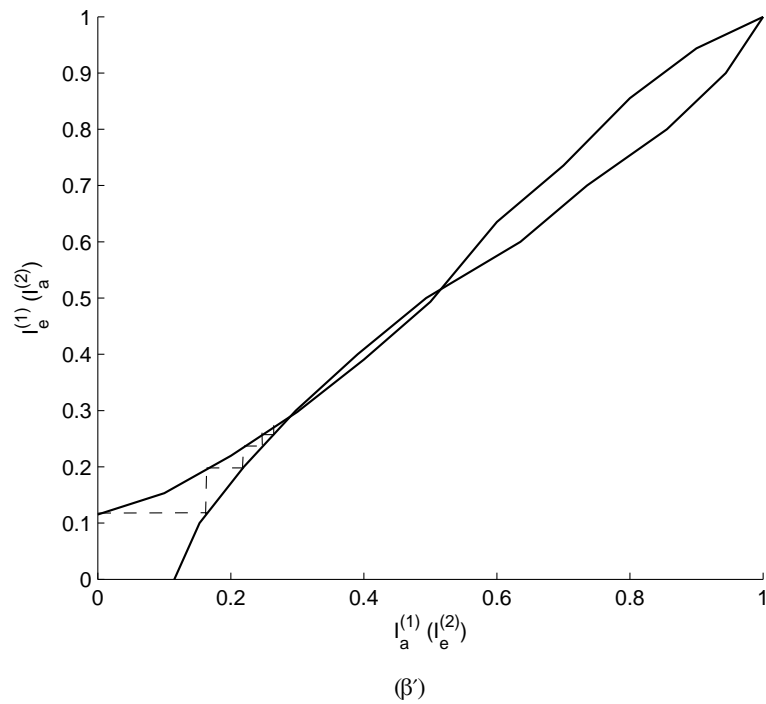
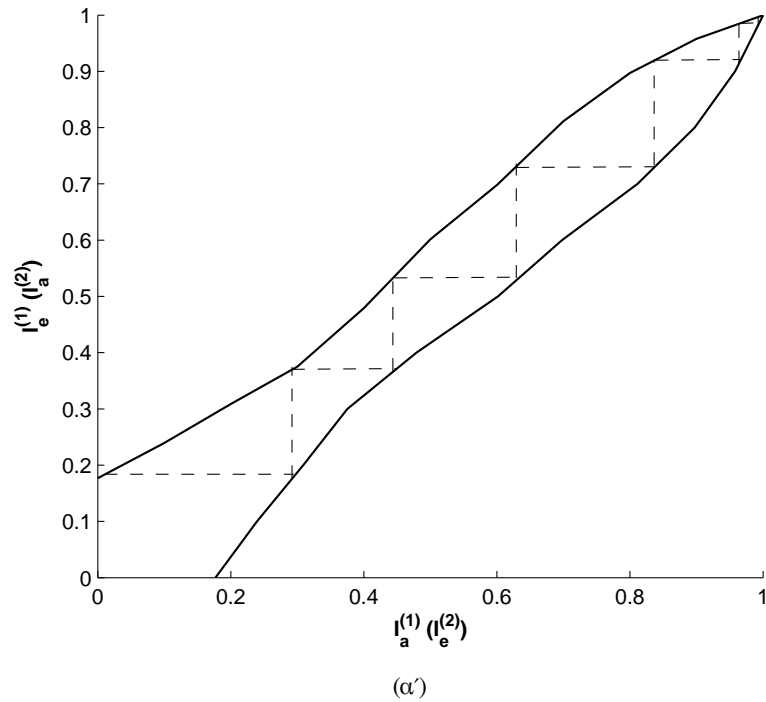
Όταν ο συμπλέκτης δεν είναι αρκετά μεγάλος, η πιθανότητα σφάλματος εξαρτάται και από άλλες λέξεις πληροφορίας χαμηλού βάρους. Έτσι, η επικρατέστερη τεχνική βελτιστοποίησης ενός περιεχόμενου συγκεραστικού κωδικοποιητή περιλαμβάνει πρώτα τη βελτιστοποίηση του ζευγαριού $(d_2, A_{d_2,2})$ όπως αναφέραμε παραπάνω, και στη συνέχεια τη βελτιστοποίηση των ζευγαριών $(d_w, A_{d_w,w})$, για μερικές τιμές του w ακόμα [38, 95, 37, 96, 44, 71].

Η δεύτερη τεχνική σχεδίασης συγκεραστικών κωδικοποιητών κατάλληλων για κώδικες τούρμπο, απαιτεί την προσομοίωση της συμπεριφοράς του συγκεραστικού αποκωδικοποιητή κατά τη διάρκεια της επαναληπτικής αποκωδικοποίησης, χρησιμοποιώντας ένα είδος στατιστικής ανάλυσης που ονομάζεται *ανάλυση μεταφοράς εξωτερικής πληροφορίας* (extrinsic information transfer (EXIT) analysis) [97, 98]. Η βασική φιλοσοφία αυτής της μεθόδου είναι να αντιμετωπιστεί ο συγκεραστικός αποκωδικοποιητής σαν στατιστικός επεξεργαστής που μετασχηματίζει μία ποσότητα εισόδου, δηλαδή την εξωτερική πληροφορία L_e του άλλου κωδικοποιητή για τα

bit πληροφορίας, σε μία ποσότητα εξόδου, δηλαδή τις L_e που υπολόγισε αυτός. Τις L_e στην είσοδο του αποκωδικοποιητή θα τις συμβολίζουμε με L_a , αφού όπως είδαμε παίζουν το ρόλο της a priori πληροφορίας. Το μέτρο της ποιότητας των L_e είναι η αμοιβαία πληροφορία $I(u; L_e)$ μεταξύ των τελευταίων και των bit πληροφορίας. Μετρήσεις των L_e δείχνουν ότι ακολουθούν προσεγγιστικά τη Γκαουσιανή κατανομή (για συμπλέκτη μεγάλου μήκους, και θεωρώντας το AWGN κανάλι). Εισάγοντας στον αποκωδικοποιητή (για συγκεκριμένο σηματοθορυβικό λόγο) a priori πληροφορία L_a που έχει συγκεκριμένη αμοιβαία πληροφορία $I_a(u; L_a)$ με τα bit εισόδου u , μετρείται η αμοιβαία πληροφορία $I_e(u; L_e)$ μεταξύ των L_e στην έξοδο του αποκωδικοποιητή και των bit πληροφορίας. Η τελευταία υπολογίζεται χρησιμοποιώντας μια εμπειρική κατανομή των L_e , όπως μετρήθηκαν στην έξοδο του SISO αποκωδικοποιητή. Έτσι κατασκευάζουμε την EXIT (εμπειρική) συνάρτηση εξόδου

$$I_e = T(I_a, E_b/N_0). \quad (5.9)$$

Θεωρώντας τον τουρμπο αποκωδικοποιητή που περιγράψαμε παραπάνω, ο οποίος απαρτίζεται από δύο πανομοιότυπους συγκεραστικούς αποκωδικοποιητές, όπου η εξωτερική πληροφορία L_e στην έξοδο του ενός χρησιμοποιείται ως a priori πληροφορία L_a στην είσοδο του άλλου, σχεδιάζουμε την (5.9) για τον πρώτο αποκωδικοποιητή, και στο ίδιο διάγραμμα την τοποθετούμε συμμετρικά ως προς την ευθεία $x = y$, προκειμένου να περιγράψουμε και τη συμπεριφορά του δεύτερου αποκωδικοποιητή. Όταν οι δύο καμπύλες τέμνονται στο σημείο $I_e^{(1)} = I_e^{(2)} = 1$ αυτό σημαίνει ότι η επαναληπτική αποκωδικοποίηση συγκλίνει σε χαμηλή πιθανότητα σφάλματος. Αντίθετα, όταν οι δύο καμπύλες τέμνονται νωρίτερα, τότε η αμοιβαία πληροφορία των L_e και των bit πληροφορίας δεν θα πλησιάσει ποτέ τη μονάδα, και ο επαναληπτικός αποκωδικοποιητής δεν θα συγκλίνει σε χαμηλή πιθανότητα σφάλματος. Η παραπάνω τεχνική χρησιμοποιείται κυρίως για να βρεθεί η τιμή του σηματοθορυβικού λόγου, πάνω από την οποία οι δύο καμπύλες δεν τέμνονται (pinch off limit [97]). Δηλαδή η τιμή του σηματοθορυβικού λόγου στην οποία ξεκινάει η περιοχί καταρράκτη στην καμπύλη απόδοσης του τουρμπο κώδικα, όταν το μήκος του συμπλέκτη είναι πολύ μεγάλο. Έτσι, μ' αυτήν την τεχνική μπορούμε να επιλέξουμε τον συγκεκριμένο κώδικα ο οποίος θα οδηγήσει σε βελτιστοποίηση του τουρμπο κώδικα για χαμηλές τιμές του σηματοθορυβικού λόγου. Στο Σχ. 5.4 βλέπουμε το διάγραμμα EXIT του αντίστοιχου αποκωδικοποιητή του συγκεραστικού κωδικοποιητή που περιγράφεται στην (5.1) (θεωρούμε ότι χρησιμοποιείται ο Max-Log-MAP αλγόριθμος αποκωδικοποίησης). Για $E_b/N_0 = 0.5$ ο τουρμπο αποκωδικοποιητής θα συγκλίνει σε χαμηλή πιθανότητα σφάλματος, ενώ για $E_b/N_0 = 0.5$ η πιθανότητα σφάλματος θα είναι μεγάλη. Η διακεκομμένη γραμμή αναπαριστά την "πορεία" της επαναληπτικής αποκωδικοποίησης (decoding trajectory), και την ανταλλαγή εσωτερικής πληροφορίας μεταξύ των δύο SISO αποκωδικοποιητών.



Σχήμα 5.4. Διαγράμματα EXIT που αντιστοιχούν στον συγκεραστικό κωδικοποιητή της (5.1) για (α') $E_b/N_0 = 0.5$ και (β') $E_b/N_0 = 0$

5.2 Αναδρομικοί Ευέλικτοι Συγκεραστικοί Κωδικοποιητές

Όπως ακριβώς και στην περίπτωση των συγκεραστικών κωδίκων, η τεχνική της απαλοιφής κωδικών bit μπορεί να μεταβάλει το ρυθμό ενός τούρμπο κώδικα, ενώ η χρήση ενός συνόλου πινάκων απαλοιφής οδηγεί σ' έναν διάτρητο τούρμπο κώδικα μεταβλητού ρυθμού [39, 40, 41, 99]. Δυστυχώς όμως, η από κοινού βελτιστοποίηση του πίνακα απαλοιφής και του “μπτρικού” τούρμπο κώδικα είναι εξαιρετικά δύσκολη, καθώς υπάρχουν πολλοί συνδυασμοί, ενώ για κάθε πιθανό υποψήφιο θα πρέπει να υπολογιστούν μετρικές (όπως π.χ. η ελάχιστη απόσταση του κώδικα που προκύπτει), χρησιμοποιώντας ιδιαίτερα πολύπλοκους αλγόριθμους [42]. Μία διαφορετική προσέγγιση είναι η κατασκευή διάτρητων συγκεραστικών κωδικοποιητών με τη μέθοδο που παρουσιάστηκε στα [35, 36, 37, 38], κατάλληλων να χρησιμοποιηθούν σε τούρμπο κώδικες. Με λίγα λόγια, προκειμένου να σχεδιάσει κανείς τούρμπο κώδικες μεταβλητού ρυθμού, είναι προτιμότερο να σχεδιάσει συγκεραστικούς κώδικες μεταβλητού ρυθμού βελτιστοποιημένους για τούρμπο κωδικοποίηση. Αυτή η μέθοδος χρησιμοποιείται στο [43, 44], όπου παράγεται μια οικογένεια αναδρομικών διάτρητων συγκεραστικών κωδικοποιητών μεταβλητού ρυθμού, από έναν μπτρικό αναδρομικό κωδικοποιητή χαμηλού ρυθμού.

Σ' αυτήν την ενότητα εξετάζουμε την κατασκευή των κωδικών οικογενειών που παρουσιάστηκαν στο κεφάλαιο 4, με τη χρήση αναδρομικών μπτρικών κωδικοποιητών. Σκοπός μας είναι να κατασκευάσουμε ευέλικτους τούρμπο κώδικες, χρησιμοποιώντας τους παραπάνω ως περιεχόμενους κωδικοποιητές. Ξεκινώντας με έναν $(n, 1, m)$ αναδρομικό συστηματικό μπτρικό συγκεραστικό κωδικοποιητή, θέλουμε να αναπτύξουμε ένα σύνολο από trellis modules, διαφόρων προφίλ πολυπλοκότητας (δηλαδή τιμών της trellis πολυπλοκότητας και αριθμών συνενώσεων), τα οποία αντιστοιχούν σε αναδρομικούς κωδικοποιητές διαφόρων ρυθμών. Τονίζουμε εδώ ότι όταν οι μεταβάσεις (ακμές) του trellis module \mathcal{M} ενός κώδικα \mathcal{C} αντιστοιχούν σε συγκεκριμένες τιμές των bit εισόδου, τότε το \mathcal{M} περιγράφει έναν συγκεκριμένο κωδικοποιητή του \mathcal{C} .

Όπως είδαμε άλλωστε, η αποκωδικοποίηση ενός τούρμπο κώδικα γίνεται με τη χρήση του αποκωδικοποιητή του περιεχόμενου συγκεραστικού κώδικα και του αλγόριθμου BCJR (ή των παραλλαγών του). Η υπολογιστική πολυπλοκότητα του αλγόριθμου BCJR, όπως και κάθε αλγόριθμου που χρησιμοποιεί το διάγραμμα trellis, εξαρτάται ισχυρά από το εκάστοτε trellis module, και συγκεκριμένα από ποσότητες όπως η trellis πολυπλοκότητα και ο αριθμός των συνενώσεων. Συγκεκριμένα, αν θεωρήσουμε ένα από τα trellis modules που κατασκευάζονται στο κεφάλαιο 4, το οποίο αντιστοιχεί σ' έναν κώδικα ρυθμού k'/n' και έχει το προφίλ πολυπλοκότητας του trellis module ενός PCC με μνήμη m' , τότε λαμβάνοντας υπόψιν τα ιδιαίτερα χαρακτηριστικά του (διάσταση του χώρου καταστάσεων σε κάθε βάθος, διάσταση του χώρου ακμών σε κάθε τμήμα, κτλ.), καθώς και την ανάλυση που έγινε στην ενότητα 2.3, ο αλγόριθμος Max-Log-MAP θα πραγματοποιήσει: $\sum_{i=1}^q (n_{l_i} - 1)2^{m'+1}$ προσθέσεις ανά k' bit πληροφορίας για να υπολογιστούν οι μετρικές ακμής γ , όπου l_i , $1 \leq i \leq q$ είναι τα (q στο πλήθος) τμήματα trellis για τα οποία ισχύει $n_{l_i} > 0$. Στη συνέχεια, πραγματοποιούνται $q2^{m'+1}$ προσθέσεις και $k'2^{m'}$

συγκρίσεις για τον υπολογισμό των προς τα εμπρός μετρικών, ενώ οι ίδιες πράξεις απαιτούνται και για τον υπολογισμό των προς τα πίσω μετρικών. Τέλος, για τον υπολογισμό των k' LLR χρειάζονται $k'(2^{m'+2} + 1)$ προσθέσεις και $k'(2^{m'+1} - 2)$ συγκρίσεις. Συνολίζοντας, ο αλγόριθμος Max-Log-MAP θα πραγματοποιήσει $(n' + q + 2k')2^{m'+1} + k'$ προσθέσεις και $4k'2^{m'} - k'$ συγκρίσεις κατά τη διάρκεια ενός trellis module, δηλαδή ανά k' bit πληροφορίας. Όπως παρατηρούμε, η υπολογιστική πολυπλοκότητα του αλγόριθμου Max-Log-MAP αυξάνεται εκθετικά με την m' .

Θεωρούμε έναν (μύ αναδρομικό) $(n, 1, m)$ μητρικό συγκεραστικό κωδικοποιητή, ο οποίος περιγράφεται από τον πίνακα γεννήτορα

$$\mathbf{G}(D) = [g^{(0)}(D) \quad g^{(1)}(D) \dots g^{(n-1)}(D)]. \quad (5.10)$$

Εφαρμόζουμε την τεχνική της απαλοιφής μονοπατιών του διαγράμματος trellis, σύμφωνα με τη σχέση (4.6), χρησιμοποιώντας συγκεκριμένο πολυώνυμο $c(X) = \sum_{i=1}^m c_i X^i$ και περίοδο T_{pr} . Όπως είδαμε και στο κεφάλαιο 4, το αποτέλεσμα είναι ένας νέος κώδικας \mathcal{C}_{pr} , καθώς και το trellis module \mathcal{M}_{pr} αυτού, το οποίο αποτελείται από T_{pr} τμήματα trellis.

Έστω επίσης ο $(n, 1, m)$ αναδρομικός συστηματικός μητρικός κωδικοποιητής

$$\mathbf{G}_{sys}(D) = \left[1 \quad \frac{g^{(1)}(D)}{g^{(0)}(D)} \dots \frac{g^{(n-1)}(D)}{g^{(0)}(D)} \right]. \quad (5.11)$$

Οι δύο μητρικοί κωδικοποιητές είναι ισοδύναμοι και παράγουν τον ίδιο κώδικα, ενώ τα (συμβατικά) trellis modules τους είναι σχεδόν ίδια. Δηλαδή, αποτελούνται από τις ίδιες μεταβάσεις (ακμές), ενώ μία μετάβαση παράγει τις ίδιες τιμές των bit εξόδου και στα δύο trellis modules [17]. Η μόνη διαφορά βρίσκεται στην αντιστοίχιση εισόδου – εξόδου, δηλαδή στην τιμή που θα πρέπει να έχει το bit πληροφορίας προκειμένου να πραγματοποιηθεί μια συγκεκριμένη μετάβαση από μία συγκεκριμένη κατάσταση. Συνεπώς, είναι δυνατόν να παράγουμε τον κώδικα \mathcal{C}_{pr} χρησιμοποιώντας τον αναδρομικό μητρικό κωδικοποιητή. Συγκεκριμένα, μπορούμε να κατασκευάσουμε το trellis module M'_{pr} , το οποίο αποτελείται από τις ίδιες ακμές που αποτελούν το \mathcal{M}_{pr} . Όπως και παραπάνω, υποθέτουμε ότι κάθε T_{pr} χρονικές στιγμές, η είσοδος του (αναδρομικού) κωδικοποιητή δεν είναι ένα καινούργιο bit πληροφορίας, αλλά ένα bit μ' που έχει προκύψει από το γραμμικό συνδυασμό των bit που βρίσκονται αποθηκευμένα στον καταχωρητή. Θα πρέπει να υπολογίσουμε την τιμή του μ' που θα μας οδηγήσει στην κατασκευή του M'_{pr} . Έστω $\mathbf{S}_{T_{pr}} = [S_{T_{pr},1} \quad S_{T_{pr},2} \dots S_{T_{pr},m}]$ μια κατάσταση του \mathcal{M}_{pr} σε βάθος T_{pr} . Με την εφαρμογή της διαδικασίας απαλοιφής μονοπατιών, από την $\mathbf{S}_{T_{pr}}$ είναι εφικτή μόνο η μετάβαση που οδηγεί στην κατάσταση $\mathbf{S}_{T_{pr}+1} = [\mu \quad S_{T_{pr},1} \dots S_{T_{pr},m-1}]$ (ο κωδικοποιητής δεν είναι αναδρομικός) σε βάθος $T_{pr} + 1$. Το μ είναι το “bit απαλοιφής”, το οποίο παράγεται από

τη σχέση (4.6), χρησιμοποιώντας το πολυώνυμο $c(X)$, δηλαδή

$$\mu = \sum_{i=1}^m c_i S_{T_{pr}, i}. \quad (5.12)$$

Θεωρείστε τώρα την ίδια κατάσταση $S_{T_{pr}}$ σε βάθος T_{pr} , στο trellis module \mathcal{M}'_{pr} . Λόγω της διαδικασίας απαλοιφής μονοπατιών, είναι δυνατή μόνο η μετάβαση που οδηγεί στην κατάσταση $S_{T_{pr}+1} = [w \ S_{T_{pr},1} \dots S_{T_{pr},m-1}]$ σε βάθος $T_{pr} + 1$, όπου

$$w = \mu' + \sum_{i=1}^m g_i^{(0)} S_{T_{pr}, i} \quad (5.13)$$

και τα $g_i^{(0)}$ συμβολίζουν τους συντελεστές του πολυωνύμου ανάδρασης $g^{(0)}(D)$ (προφανώς $g_0^{(0)} = 1$). Αφού τα \mathcal{M}'_{pr} και \mathcal{M}_{pr} αποτελούνται από τις ίδιες ακμές (μεταβάσεις), τότε $\mu = w$, και από τις σχέσεις (5.12) και (5.13)

$$\mu' = \sum_{i=1}^m (c_i + g_i^{(0)}) s_{T_{pr}}^i = \sum_{i=1}^m c'_i s_{T_{pr}}^i \quad (5.14)$$

όπου το νέο πολυώνυμο $c'(X) = \sum_{i=1}^m c'_i X^i$ χρησιμοποιείται για την παραγωγή του κατάλληλου bit απαλοιφής για τον αναδρομικό μητρικό κωδικοποιητή. Αξίζει να τονίσουμε σ' αυτό το σημείο ότι η επιλογή των μεταβάσεων και κατ' επέκτασιν των μονοπατιών που θα "επιζήσουν" από το μητρικό διάγραμμα trellis, εξαρτάται αποκλειστικά από τις τιμές που παίρνει η είσοδος (το bit) απαλοιφής (δηλαδή από το τελευταίο τμήμα trellis του \mathcal{M}'_{pr} ή του \mathcal{M}_{pr}). Τα πρώτα $(T_{pr} - 1)$ τμήματα και των δύο trellis modules αντιστοιχούν σε καινούργια bit πληροφορίας.

Εάν θεωρήσουμε μία ακολουθία από p "αντίγραφα" του \mathcal{M}_{pr} και μία αντίστοιχη από p "αντίγραφα" του \mathcal{M}'_{pr} , και χρησιμοποιήσουμε και στις δύο περιπτώσεις τον ίδιο πίνακα απαλοιφής κωδικών bit P προκειμένου να αυξήσουμε τον κωδικό ρυθμό (όπως ακριβώς είδαμε στο κεφάλαιο 4), θα κατασκευάσουμε τα trellis modules \mathcal{M}_{pu} και \mathcal{M}'_{pu} αντίστοιχα. Τα τελευταία, παράγουν τον ίδιο $(pT_{pr}n - N, p(T_{pr} - 1))$ συγκεραστικό κώδικα \mathcal{C}_{pu} , όπου N είναι ο αριθμός των κωδικών bit που απαλείφονται. Παρόλ' αυτά, αντιστοιχούν (πιθανότατα) σε διαφορετικούς κωδικοποιητές. Το σημαντικότερο συμπέρασμα της παραπάνω ανάλυσης είναι ότι τα \mathcal{M}_{pu} και \mathcal{M}'_{pu} έχουν το ίδιο προφίλ πολυπλοκότητας, αφού απαρτίζονται από τις ίδιες ακμές, ενώ μια συγκεκριμένη ακμή αντιστοιχεί στον ίδιο αριθμό κωδικών bit και στα δύο trellis modules. Συνεπώς, δοθέντος ενός $(n, 1, m)$ αναδρομικού μητρικού κωδικοποιητή και συγκεκριμένων παραμέτρων απαλοιφής μονοπατιών του διαγράμματος trellis και απαλοιφής κωδικών bit (δηλαδή $c'(X)$, T_{pr} , p , P), μπορούμε εύκολα να υπολογίσουμε οποιοδήποτε μέγεθος που σχετίζεται με την πολυπλοκότητα του trellis module \mathcal{M}'_{pu} , υπολογίζοντας πρώτα το πολυώνυμο $c(X)$ από τη σχέση (5.14), δηλαδή το πολυώνυμο που παράγει το \mathcal{M}_{pr} από τον ισοδύναμο $(n, 1, m)$

μη αναδρομικό κωδικοποιητή. Στη συνέχεια, χρησιμοποιώντας τα αποτελέσματα της ανάλυσης στην ενότητα 4.2 μπορούμε να υπολογίσουμε το προφίλ πολυπλοκότητας του \mathcal{M}_{pu} , το οποίο ταυτίζεται με το αντίστοιχο του \mathcal{M}'_{pu} .

Παρόλο που το \mathcal{M}'_{pu} προκύπτει από έναν αναδρομικό συστηματικό κωδικοποιητή και το αντίστοιχο trellis module αυτού, δεν αντιστοιχεί απαραίτητα σε έναν αναδρομικό συστηματικό κωδικοποιητή. Η συστηματικότητα εξασφαλίζεται με την επιβολή περιορισμών στον πίνακα απαλοιφής κωδικών bit P . Συγκεκριμένα, ο P δεν πρέπει να απαλείφει τα συστηματικά κωδικά bit του μητρικού trellis, στα τμήματα trellis του \mathcal{M}'_{pu} που αντιστοιχούν σε bit πληροφορίας. Επίσης, το \mathcal{M}'_{pu} περιγράφει έναν αναδρομικό κωδικοποιητή, αν και μόνο αν για κάθε ακολουθία πληροφορίας με βάρος Hamming ίσο με τη μονάδα παράγεται κωδική λέξη απείρου βάρους Hamming.

Όπως ακριβώς και στο κεφάλαιο 4, δοθέντος ενός $(n, 1, m)$ αναδρομικού συγκεραστικού κωδικοποιητή και ενός συγκεκριμένου κωδικού ρυθμού k'/n' , κατασκευάζουμε trellis modules (που αντιστοιχούν σε αναδρομικούς συστηματικούς κωδικοποιητές) των οποίων n trellis πολυπλοκότητα και ο αριθμός των συνενώσεων είναι ίσα με τα αντίστοιχα μεγέθη του trellis module των PCC με μήνιμ $m' \leq m$. Για $m' = m$ η τεχνική απαλοιφής μονοπατιών του διαγράμματος trellis δεν εφαρμόζεται, αφού η επιθυμητή πολυπλοκότητα μπορεί να επιτευχθεί χρησιμοποιώντας μόνο την τεχνική απαλοιφής κωδικών bit (κατασκευάζοντας δηλαδή έναν κλασσικό PCC, χρησιμοποιώντας το μητρικό trellis module). Για την κατασκευή των υπόλοιπων κωδικών χρησιμοποιούνται και οι δύο προαναφερθείσες τεχνικές. Συγκεκριμένα, όπως και στην ενότητα 4.3 δοθέντων των k'/n' και m' , βρίσκουμε όλες τις τριάδες (p, T_{pr}, \hat{d}) οι οποίες, σε συνδυασμό με τους κατάλληλους πίνακες απαλοιφής, παράγουν από το μη αναδρομικό ισοδύναμο μητρικό κωδικοποιητή trellis modules με την επιθυμητή πολυπλοκότητα. Υπενθυμίζουμε, ότι το \hat{d} είναι ο βαθμός του πολυώνυμου $c(X)$, συνεπώς για συγκεκριμένο \hat{d} πρέπει να ληφθούν υπόψιν όλα τα αντίστοιχα πολυώνυμα $c(X)$. Για κάθε πολυώνυμο $c(X)$ χρησιμοποιούμε τη σχέση (5.14) προκειμένου να υπολογίσουμε το αντίστοιχο πολυώνυμο $c'(X)$, για τον αναδρομικό μητρικό κωδικοποιητή. Λαμβάνοντας υπόψιν μας όλα τα πιθανά πολυώνυμα $c'(X)$ και τους πίνακες απαλοιφής κωδικών bit P , επιλέγουμε αυτά που οδηγούν στο βέλτιστο αναδρομικό συστηματικό κωδικοποιητή.

5.3 Αποτελέσματα Αναζήτησης και Προσομιώσεις

Σ'αυτήν την ενότητα παρουσιάζουμε μια οικογένεια αναδρομικών συστηματικών κωδικοποιητών, οι οποίοι παράγονται από έναν $(2, 1, 5)$ αναδρομικό συστηματικό μητρικό κωδικοποιητή. Όλοι οι κωδικοποιητές είναι ρυθμού $(n' - 1)/n'$, για $2 \leq n' \leq 6$. Εφόσον οι συγκεκριμένοι κωδικοποιητές σχεδιάζονται με σκοπό να χρησιμοποιηθούν ως περιεχόμενοι σε τούρμπο κώδικες, η βελτιστοποίησή τους (για μέτριο και μεγάλο SNR) περιλαμβάνει πρωτίστως τη μεγιστοποίηση της απόστασης d_2 και την ελαχιστοποίηση του $A_{d_2,2}$, και στη συνέχεια την αντίστοιχη βελτι-

στοποίηση των d_3 και $A_{d_3,3}$ [17, 35, 36, 38]. Ο Πίνακας 5.1 περιέχει όλους τους κώδικες (κωδι-

Πίνακας 5.1. Οικογένεια κωδικοποιητών που παράγεται από τον αναδρομικό συστηματικό μητρικό κωδικοποιητή $\mathbf{G}(D) = \left[1 \quad \frac{1+D+D^4+D^5}{1+D^2+D^3+D^4+D^5} \right]$

k'/n'	m'	T_{pr}	p	$c(X)$	P	$(d_2, A_{d_2,2}), (d_3, A_{d_3,3})$	pinch-off limit (dB)
1/2	5	-	-	-	3	(20, 1), (9, 2)	0.3
1/2	4	2	1	$X + X^2 + X^4 + X^5$	30	(12, 1), (7, 1)	0.2
1/2	3	2	1	$X + X^2 + X^3 + X^4 + X^5$	30	(8, 1), (7, 3)	0.15
1/2	2	2	1	$X^2 + X^3 + X^4 + X^5$	30	(6, 1), (5, 1)	0.15
2/3	5	-	-	-	23	(10, 1), (5, 2)	0.9
2/3	4	3	1	$X + X^3 + X^4 + X^5$	230	(7, 2), (4, 1)	0.82
2/3	3	3	1	$X^2 + X^3 + X^4 + X^5$	320	(4, 1), (4, 1)	0.8
2/3	2	2	2	$X^2 + X^3 + X^4 + X^5$	2021	(4, 3), (3, 1)	0.82
3/4	5	-	-	-	223	(7, 1), (3, 1)	1.4
3/4	4	4	1	$X + X^2 + X^3 + X^4 + X^5$	2230	(4, 1), (3, 2)	1.25
3/4	3	2	3	$X + X^2 + X^3 + X^4 + X^5$	202030	(3, 1), (3, 2)	1.25
4/5	5	-	-	-	2223	(6, 3), (3, 2)	1.65
4/5	4	5	1	$X^2 + X^3 + X^4 + X^5$	22230	(3, 1), (3, 2)	1.6
4/5	3	3	2	$X^2 + X^3 + X^4 + X^5$	220320	(2, 1), (3, 4)	1.6
4/5	2	2	4	$X^2 + X^3 + X^4 + X^5$	20202030	(2, 1), (3, 9)	1.75
5/6	5	-	-	-	22223	(4, 1), (3, 3)	1.9
5/6	4	6	1	$X^2 + X^3 + X^4 + X^5$	322220	(3, 1), (3, 5)	1.87
5/6	3	2	5	$X + X^2 + X^3 + X^4 + X^5$	2020202030	(3, 5), (3, 10)	1.86
5/6	2	2	5	$X^2 + X^3 + X^4 + X^5$	2020202030	(2, 2), (3, 16)	1.95

κοποιητές) της οικογένειας που κατασκευάσαμε, καθώς και πληροφορίες για το ρυθμό τους, τα m' , T_{pr} , p , $c'(X)$, τον πίνακα απαλοιφής κωδικών bit P , και τα $(d_2, A_{d_2,2})$ και $(d_3, A_{d_3,3})$. Η αναπαράσταση ενός συμβόλου του P στο δυαδικό σύστημα αποτελεί την αντίστοιχη στήλη του. Επίσης, για κάθε κωδικοποιητή παρέχεται η τιμή του E_b/N_0 (pinch-off limit) στην οποία ξεκινάει η περιοχή καταρράκτη για ασυμπτωτικά μεγάλο μήκος συμπλέκτη (όταν χρησιμοποιείται ο Max-Log-MAP αλγόριθμος αποκωδικοποίησης), η οποία υπολογίστηκε με τη χρήση των διαγραμμάτων EXIT.

Στο [44], παρέχονται οι καλύτεροι αναδρομικοί συστηματικοί PCC για την κατασκευή τούρμπο κωδικών, οι οποίοι προκύπτουν από (2, 1) μητρικούς κωδικοποιητές. Επίσης, κατασκευάζονται και κωδικοποιητές (PCC) μεταβλητού ρυθμού από ένα μοναδικό μητρικό (2, 1, m) κωδικοποιητή, για διάφορες τιμές του m . Σχεδόν όλοι οι κωδικοποιητές στον Πίνακα 5.1 έχουν την ίδια d_2 με τους καλύτερους κωδικοποιητές στο [44].

Στη συνέχεια, έχουμε προσομοιώσει την απόδοση τούρμπο κωδικών οι οποίοι χρησιμοποιούν

ως περιεχόμενο συγκεραστικό κωδικοποιή τον ευέλικτο μπτρικό κωδικοποιή στον Πίνακα 5.1, καθώς και την αντίστοιχη κωδική του οικογένεια, για το AWGN κανάλι, χρησιμοποιώντας BPSK διαμόρφωση. Η αποκωδικοποίηση γίνεται χρησιμοποιώντας τον αλγόριθμο Max-Log-MAP. Στο Σχ.5.5 έχουμε χρησιμοποιήσει τον ίδιο τυχαίο συμπλέκτη μήκους 1000 για όλες τις προσομοιώσεις. Επίσης, για έναν συγκεκριμένο κωδικοποιή και για συγκεκριμένη τιμή του SNR θεωρούμε ότι ο αποκωδικοποιή πραγματοποιεί σταθερό αριθμό L_{E_b/N_0} επαναλήψεων, ο οποίος είναι μικρότερος ή ίσος του 10. Πιο συγκεκριμένα, ο L_{E_b/N_0} είναι ίσος με το μικρότερο αριθμό επαναλήψεων, πέραν του οποίου η πιθανότητα σφάλματος δεν βελτιώνεται άλλο, εάν ο τελευταίος είναι μικρότερος του 10, και ίσος με 10 σε διαφορετική περίπτωση. Όπως είναι προφανές, εκτός από την υπολογιστική πολυπλοκότητα του αλγόριθμου αποκωδικοποίησης, η συνολική υπολογιστική πολυπλοκότητα της τούρμπο αποκωδικοποίησης εξαρτάται και από το πλήθος των επαναλήψεων που θα πραγματοποιηθούν. Γι'αυτό, στον Πίνακα 5.2 παρέχουμε τις τιμές του L_{E_b/N_0} που σχετίζονται με τις προσομοιώσεις στο Σχ. 5.5. Συγκεκριμένα, η πρώτη γραμμή του Πίνακα 5.2 περιέχει τον αύξοντα αριθμό των σημείων των γραφικών παραστάσεων. Ένα σημείο της γραφικής παράστασης συγκεκριμένου συγκεραστικού κωδικοποιή αντιστοιχεί σε συγκεκριμένη τιμή του SNR (Σχ. 5.5).

Όπως φαίνεται στο Σχ. 5.5, η *περιοχή καταρράκτη* (waterfall region) στις καμπύλες απόδοσης του τούρμπο κώδικα εμφανίζεται σε μικρότερες τιμές του E_b/N_0 όταν χρησιμοποιούνται ως περιεχόμενοι κωδικοποιήτες οι (n', k', m') κωδικοποιήτες του Πίνακα 5.1 για $m' < m$, σε σχέση με την περίπτωση που χρησιμοποιούνται οι (n', k', m) κωδικοποιήτες της οικογένειας (συμπεριλαμβανομένου και του μπτρικού). Αυτό το γεγονός ακολουθείται από μία σημαντική μείωση της συνολικής υπολογιστικής πολυπλοκότητας. Με λίγα λόγια, σε συγκεκριμένες περιοχές του E_b/N_0 , μπορούμε να μειώσουμε σημαντικά την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης, αυξάνοντας ταυτόχρονα την απόδοση του τούρμπο κώδικα, με τη χρήση των τεχνικών της απαλοιφής μονοπατιών του διαγράμματος trellis και της απαλοιφής κωδικών bit. Για παράδειγμα, όταν χρησιμοποιείται ο $(2, 1, 5)$ μπτρικός κωδικοποιή επιτυγχάνεται πιθανότητα σφάλματος $2 \cdot 10^{-4}$ για E_b/N_0 ίσο με 1.5 dB (Σχ. 5.5(α')). Όταν χρησιμοποιείται ο $(2, 1, 3)$ κωδικοποιή, για την ίδια τιμή του E_b/N_0 , επιτυγχάνεται πιθανότητα σφάλματος $2 \cdot 10^{-6}$, ενώ ταυτόχρονα μειώνεται η υπολογιστική πολυπλοκότητα κατά 75% (αυτό προκύπτει από τη σύντομη ανάλυση στην ενότητα 5.2 και τον αριθμό των επαναλήψεων στον Πίνακα 5.1). Επίσης, όταν χρησιμοποιείται ο $(2, 1, 5)$ μπτρικός κωδικοποιή επιτυγχάνεται πιθανότητα σφάλματος $6 \cdot 10^{-3}$ για E_b/N_0 ίσο με 1.25 dB, ενώ όταν χρησιμοποιείται ο $(2, 1, 2)$ κωδικοποιή (Σχ. 5.5(β')) επιτυγχάνεται πιθανότητα σφάλματος 10^{-4} και παράλληλα μειώνεται η υπολογιστική πολυπλοκότητα κατά 84%. Παρόμοια συμπεριφορά παρατηρείται για όλους τους κωδικούς ρυθμούς. Για την ακρίβεια, δεν χρειάζεται να χρησιμοποιηθούν οι (πολυπλοκότεροι) (n', k', m) κωδικοποιήτες κάθε ρυθμού, για πιθανότητες σφάλματος μεγαλύτερες από 10^{-6} . Η καλύτερη απόδοση (ή σε κάποιες περιπτώσεις η παρεμφερής απόδοση) των κωδικών με

μικρή τιμή του m' σε χαμηλές τιμές του E_b/N_0 προβλέπεται (ποιοτικά) και από τις τιμές του E_b/N_0 στις οποίες εμφανίζεται η περιοχή καταρράκτη στο διάγραμμα της απόδοσης τους (για ασυμπτωτικά μεγάλο συμπλέκτη), οι οποίες περιέχονται στον Πίνακα 5.1.

Οι προτεινόμενες κατασκευές μπορούν να οδηγήσουν σε τούρμπο σχήματα κωδικοποίησης που διαχειρίζονται αποδοτικότερα τους πόρους του συστήματος. Υποθέστε για παράδειγμα ότι η τιμή του E_b/N_0 είναι 1.75 dB, και ότι ο κώδικας τούρμπο χρησιμοποιεί τον (2, 1, 4) ($L_{1.75} = 9$) συγκεραστικό κωδικοποιητή της οικογένειας, επιτυγχάνοντας πιθανότητα σφάλματος $3 \cdot 10^{-7}$ (Σχ. 5.5(α')). Υποθέστε επίσης ότι η τιμή του E_b/N_0 αυξάνεται κατά 1.75 dB. Σ' αυτήν την περίπτωση δεν χρειάζεται να συνεχίσουμε να χρησιμοποιούμε έναν τόσο ισχυρό περιεχόμενο κωδικοποιητή. Για την κωδικοποίηση της επόμενης ακολουθίας που πρόκειται να μεταδοθεί, μπορούμε είτε να αυξήσουμε το ρυθμό είτε να μειώσουμε την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε τον (5, 4, 4) κωδικοποιητή (Σχ. 5.5(β')) ($L_{3.5} = 7$), αυξάνοντας σημαντικά το ρυθμό και διατηρώντας την ίδια τιμή του m' . Εναλλακτικά, θα μπορούσε να χρησιμοποιηθεί ο (3, 2, 3) κωδικοποιητής (Σχ. 5.5(β')) ($L_{3.5} = 3$), οδηγώντας σε μεγάλη μείωση του ρυθμού και της πολυπλοκότητας. Τέλος, θα μπορούσε να χρησιμοποιηθεί ο (2, 1, 2) κωδικοποιητής (Σχ. 5.5(α')) ($L_{3.5} = 3$), διατηρώντας τον ίδιο ρυθμό και μειώνοντας υπερβολικά την υπολογιστική πολυπλοκότητα. Σ' αυτό το σημείο κάποιος μπορεί να αναρωτηθεί, εάν απλά αρκούσε να μειώσουμε τον αριθμό των επαναλήψεων προκειμένου να μειώσουμε την υπολογιστική πολυπλοκότητα της αποκωδικοποίησης. Η υπολογιστική πολυπλοκότητα που αντιστοιχεί σε 3 επαναλήψεις του (2, 1, 2) αποκωδικοποιητή είναι ελαφρώς μικρότερη από την πολυπλοκότητα μιας επανάληψης του (2, 1, 4) αποκωδικοποιητή που χρησιμοποιήθηκε αρχικά. Η πιθανότητα σφάλματος όμως που επιτυγχάνεται με τον (2, 1, 4) κωδικοποιητή για E_b/N_0 ίσο με 3.5dB, πραγματοποιώντας μόνο μία επανάληψη, είναι μόλις $6 \cdot 10^{-5}$.

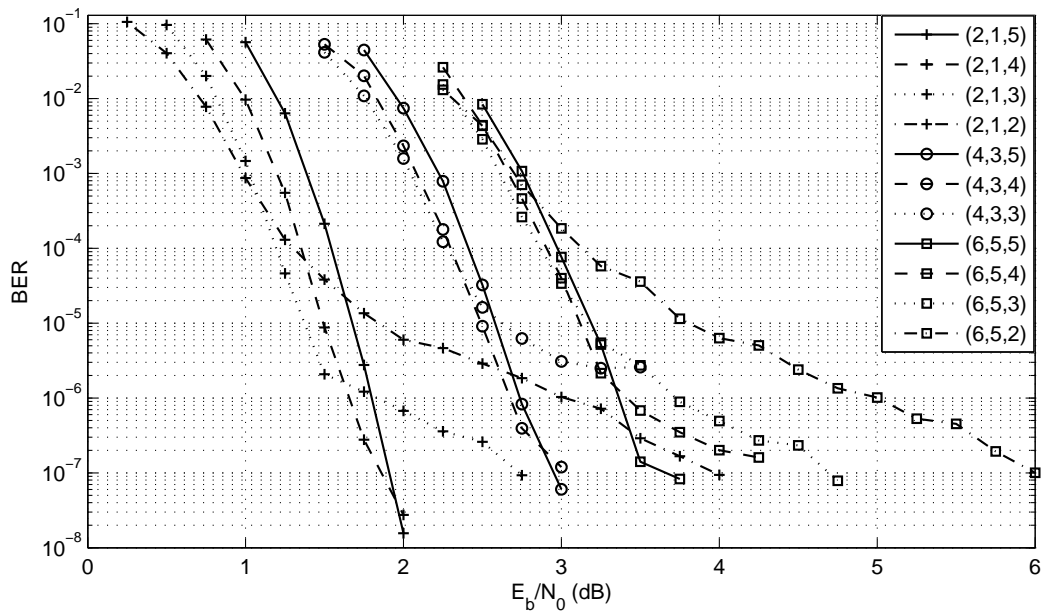
Παρόλο που πραγματοποιείται συγκεκριμένος αριθμός επαναλήψεων L_{E_b/N_0} , για κάθε κώδικα και κάθε τιμή του E_b/N_0 , πολλές από τις ληφθείσες ακολουθίες χρειάζονται λιγότερες επαναλήψεις προκειμένου να διορθωθούν όλα τα σφάλματα. Υποθέτοντας την περίπτωση που η επαναληπτική αποκωδικοποίηση σταματάει όταν διορθωθούν όλα τα σφάλματα ή όταν πραγματοποιηθούν L_{E_b/N_0} επαναλήψεις, ο μέσος αριθμός επαναλήψεων \tilde{L}_{E_b/N_0} για κάθε περιεχόμενο κωδικοποιητή και για κάθε τιμή του E_b/N_0 παρέχεται σε παρένθεση στον Πίνακα 5.1. Όπως βλέπουμε, ο μέσος αριθμός των επαναλήψεων συνήθως είναι πολύ μικρότερος από τον L_{E_b/N_0} . Αυτή η περίπτωση είναι ισοδύναμη με το κριτήριο παύσης της αποκωδικοποίησης (stopping criterion) που χρησιμοποιείται στον τούρμπο κώδικα του πρότυπου LTE (Long Term Evolution) [100], το οποίο βασίζεται σε κώδικες κυκλικού ελέγχου πλεονασμού (cyclic redundancy check CRC) [17]. Κάθε λέξη πληροφορίας κωδικοποιείται πρώτα χρησιμοποιώντας έναν CRC κώδικα και στη συνέχεια κωδικοποιείται από τον τούρμπο κωδικοποιητή. Κατά την αποκωδικοποίηση, έπειτα από κάθε επανάληψη ελέγχεται το σύνδρομο της εκτιμηθείσας

ακολουθίας, προκειμένου να διαπιστωθεί εάν είναι κωδική λέξη του CRC κώδικα. Εάν είναι, θεωρείται ότι η τούρμπο αποκωδικοποίηση διόρθωσε όλα τα σφάλματα και τερματίζεται. Προφανώς, η χρήση του παραπάνω κριτηρίου θα οδηγήσει σε μείωση της απόδοσης σε σχέση με το Σχ. 5.5, αφού η πιθανότητα μη ανίχνευσης σφάλματος του CRC κώδικα δεν είναι μηδενική.

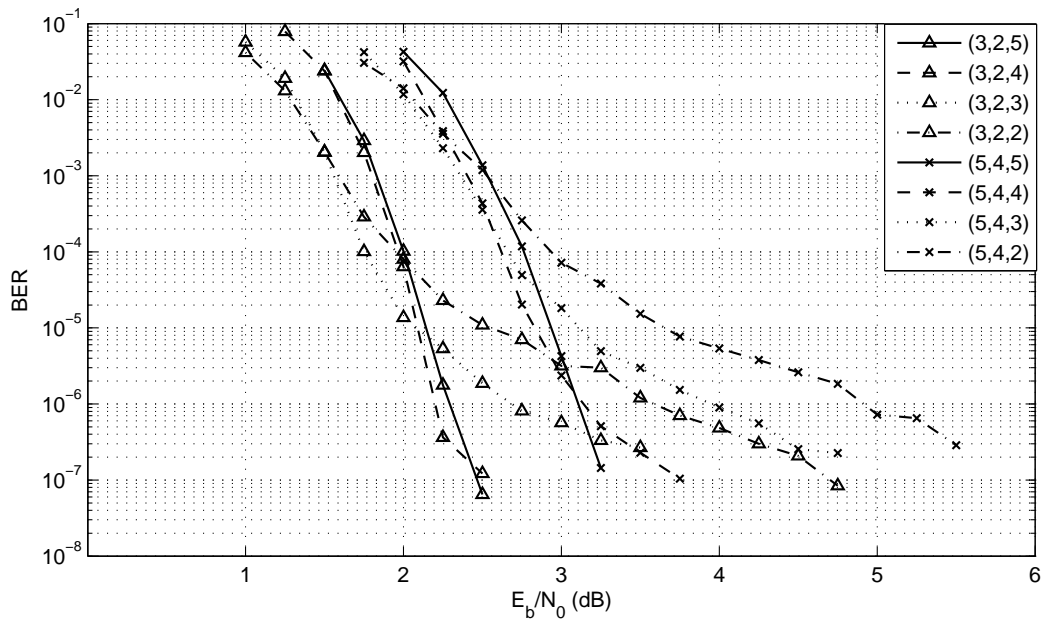
Στο Σχ. 5.6, προσομοιώνουμε τη μέση απόδοση των τούρμπο κωδίκων με συμπλέκτη μήκους 1000, που χρησιμοποιούν ως περιεχόμενους κωδικοποιητές τους κωδικοποιητές της οικογένειας που κατασκευάστηκε, διαλέγοντας για κάθε λέξη που προσομοιώνεται έναν καινούργιο τυχαίο συμπλέκτη. Ο Πίνακας 5.3 περιέχει τους αριθμούς επαναλήψεων σε αντιστοιχία με τον Πίνακα 5.2. Παρατηρώντας το Σχ. 5.6, οδηγούμαστε σε παρόμοια συμπεράσματα για τη μέση απόδοση, όπως στην περίπτωση του τούρμπο κώδικα στο Σχ.5.5.

Επίσης, στα Σχ. 5.7 και 5.8 επαναλαμβάνουμε τις παραπάνω προσομοιώσεις χρησιμοποιώντας συμπλέκτες μήκους 10000. Συγκεκριμένα, στο Σχ. 5.7 χρησιμοποιείται ο ίδιος τυχαίος συμπλέκτης για όλους τους κώδικες, ενώ στο Σχ. 5.8 επιλέγεται διαφορετικός τυχαίος συμπλέκτης για κάθε λέξη πληροφορίας που μεταδίδεται. Ο μέγιστος αριθμός των επαναλήψεων αποκωδικοποίησης είναι 4. Στους Πίνακες 5.4 και 5.5 παρέχονται πληροφορίες για τον αριθμό των επαναλήψεων της αποκωδικοποίησης που πραγματοποιήθηκαν για κάθε κώδικα και τιμή του E_b/N_0 στα Σχ. 5.7 και 5.8 αντίστοιχα. Τα συμπεράσματα που προκύπτουν είναι παρόμοια (ποιοτικά) με αυτά της προηγούμενης περίπτωσης, όπου χρησιμοποιήθηκε συμπλέκτης μήκους 1000.

Στους Πίνακες 5.6 και 5.7 παρουσιάζεται ο βέλτιστος τρόπος χρήσης (όπως προέκυψε από προσομοιώσεις) της οικογένεια στον Πίνακα 5.1, όταν ο αντίστοιχος τούρμπο κώδικας χρησιμοποιεί τον τυχαίο συμπλέκτη μήκους 1000 που χρησιμοποιήθηκε στο Σχ. 5.5. Συγκεκριμένα, για κάθε τιμή του E_b/N_0 περιέχεται (για κάθε κωδικό ρυθμό) ο συγκεραστικός κωδικοποιητής ο οποίος οδηγεί σε έναν τούρμπο κώδικα που επιτυγχάνει $P_b(E) \leq \phi$, για διάφορες τιμές του ϕ , με τη μικρότερη υπολογιστική πολυπλοκότητα αποκωδικοποίησης. Η υπολογιστική πολυπλοκότητα προκύπτει από την ανάλυση της πολυπλοκότητας του αλγόριθμου Max-Log-MAP στην ενότητα 5.2, καθώς και από το πλήθος των επαναλήψεων που πρέπει να πραγματοποιηθούν προκειμένου να επιτευχθεί η συγκεκριμένη πιθανότητα σφάλματος. Εάν δύο κώδικες έχουν παρεμφερή τιμή πολυπλοκότητας και διαφορετικό ρυθμό, περιέχεται μόνο ο κώδικας με το μεγαλύτερο ρυθμό. Στους Πίνακες 5.6 και 5.7 επαναλαμβάνεται η ίδια διαδικασία, μόνο που για κάθε λέξη που μεταδίδεται επιλέγεται καινούργιος τυχαίος συμπλέκτης μήκους 1000.



(α)

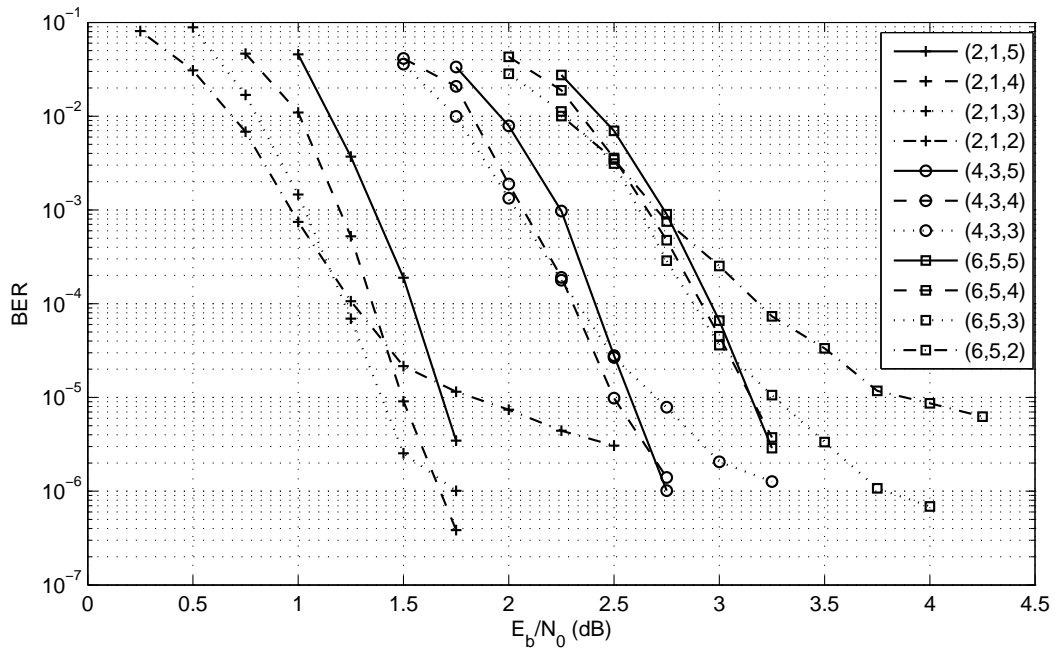


(β)

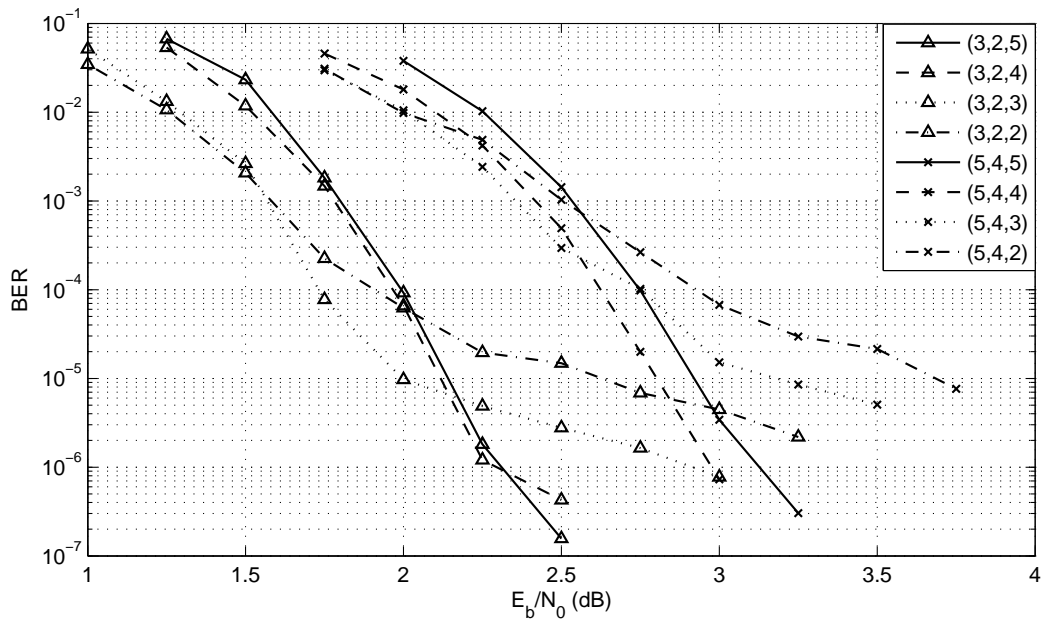
Σχήμα 5.5. Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Χρησιμοποιείται τυχαίος συμπλέκτης μήκους 1000.

Πίνακας 5.2. Αριθμός επαναλήψεων αποκωδικοποίησης των κωδίκων στο Σχ. 5.5

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(2,1,5)	3 (2,8)	8 (2,9)	10 (2,2)	10 (1,9)	10 (1,9)											
(2,1,4)	5 (4,4)	7 (3,8)	10 (2,8)	10 (2,2)	9 (2,0)	9 (1,8)										
(2,1,3)	4 (3,9)	7 (4,9)	10 (3,8)	10 (2,9)	10 (2,4)	9 (2,1)	7 (1,9)	6 (1,8)	5 (1,6)	3 (1,4)						
(2,1,2)	3 (3)	6 (5,7)	10 (6,2)	10 (4,6)	10 (3,5)	8 (2,8)	8(2,4)	7 (2,1)	5 (2,0)	5 (1,8)	5 (1,7)	3 (1,5)	3 (1,3)	3 (1,2)	3 (1,1)	3 (1)
(3,2,5)	7 (4,1)	10 (2,9)	10 (2,1)	10 (1,8)	9 (1,7)											
(3,2,4)	2 (2)	5 (3,5)	10 (2,8)	10 (2,1)	10 (1,8)	10 (1,7)										
(3,2,3)	5 (4,7)	6 (4,9)	10 (4,2)	10 (3)	10 (2,4)	8 (2,1)	8 (1,9)	7 (1,7)	4 (1,4)	3 (1,2)	3 (1,1)					
(3,2,2)	5 (4,8)	7 (5,6)	10 (4,8)	10 (3,6)	8 (2,8)	8 (2,3)	7 (2,1)	7 (1,9)	6 (1,7)	4 (1,5)	4 (1,3)	4 (1,1)	3 (1,1)	3 (1)	3 (1)	3 (1)
(4,3,5)	5 (4)	10 (3,7)	10 (2,4)	10 (1,9)	10 (1,6)	10 (1,3)										
(4,3,4)	5 (4,5)	6 (4,3)	10 (3,2)	10 (2,4)	10 (2)	10 (1,7)	8 (1,5)									
(4,3,3)	4 (3,8)	8 (5)	10 (3,7)	10 (2,7)	10 (2,2)	10 (1,9)	7 (1,7)	6 (1,5)	5 (1,3)							
(5,4,5)	3 (2,8)	6 (3,4)	10 (2,5)	10 (1,8)	10 (1,5)	10 (1,2)										
(5,4,4)	4 (3,5)	9 (3,6)	10 (2,5)	10 (2)	10 (1,7)	10 (1,4)	7 (1,2)	6 (1,1)								
(5,4,3)	3 (3)	7 (5,2)	10 (4,4)	10 (3,1)	9 (2,4)	8 (2)	8 (1,7)	7 (1,5)	6 (1,3)	5 (1,2)	5 (1,1)	4 (1)	4 (1)			
(5,4,2)	3 (3)	5 (4,5)	7 (4,6)	8 (3,7)	8 (2,9)	8 (2,4)	7 (2)	6 (1,7)	6 (1,5)	6 (1,3)	5 (1,2)	3 (1,1)	3 (1)	3 (1)	2 (1)	2 (1)
(6,5,5)	7 (3,5)	10 (2,4)	10 (1,8)	10 (1,4)	10 (1,2)	7 (1)										
(6,5,4)	5 (4)	10 (3,7)	10 (2,4)	10 (1,9)	10 (1,5)	7 (1,3)	6 (1,1)	6 (1)	6 (1)							
(6,5,3)	5 (4)	9 (3,7)	10 (2,6)	10 (2,1)	9 (1,7)	8 (1,4)	7 (1,2)	6 (1,1)	6 (1)	5 (1)	5 (1)					
(6,5,2)	4 (3,7)	6 (4,2)	10 (3,7)	10 (2,7)	8 (2,2)	7 (1,8)	7 (1,6)	6 (1,3)	5 (1,2)	5 (1,1)	3 (1)	3 (1)	3 (1)	2 (1)	2 (1)	2 (1)



(α)

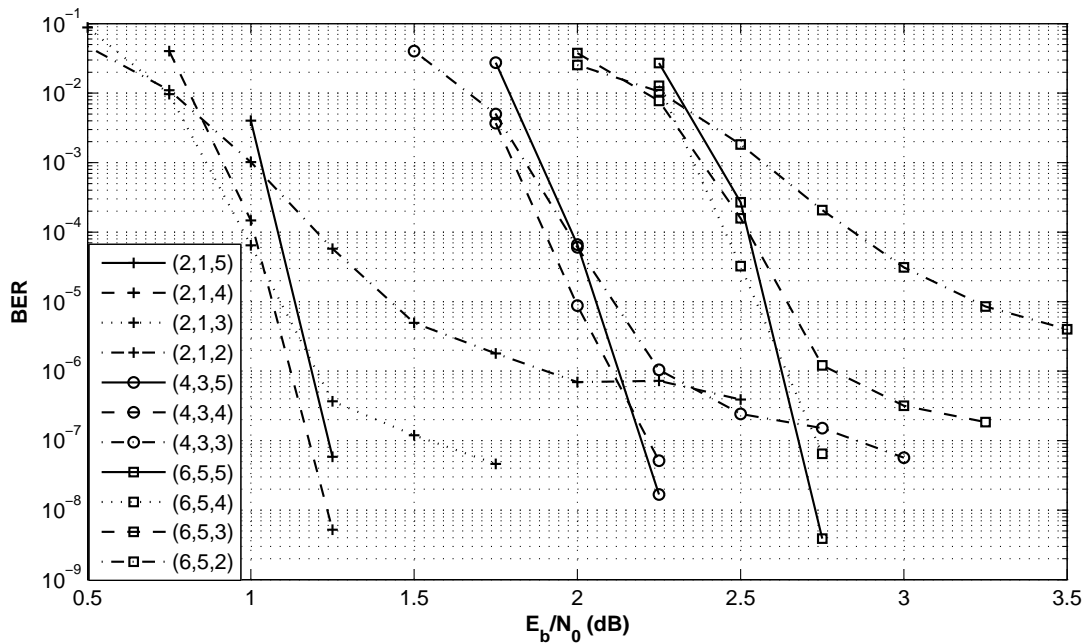


(β)

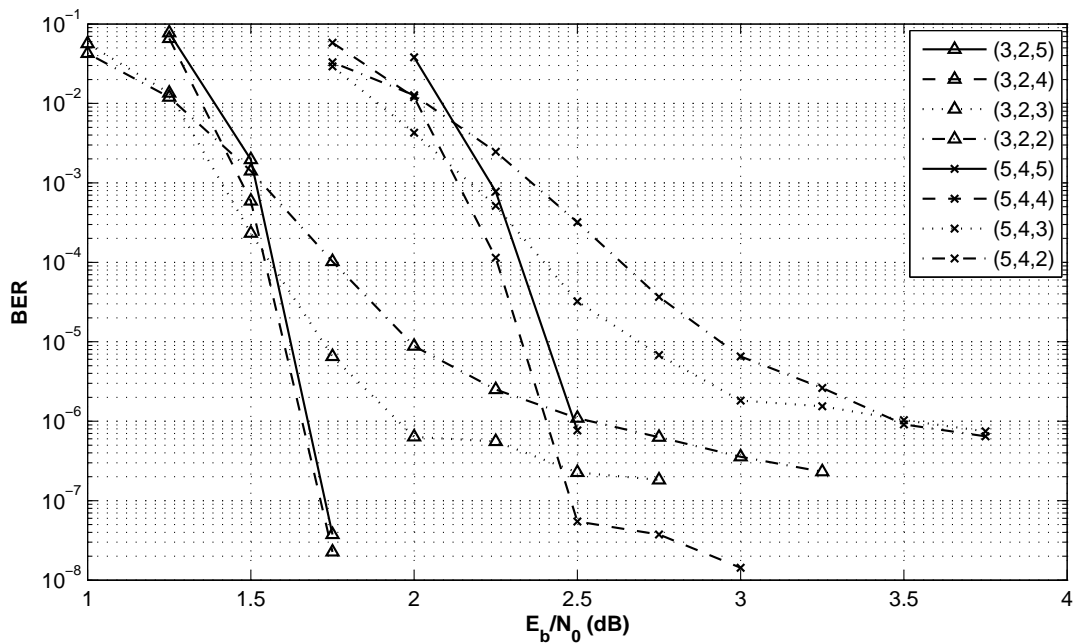
Σχήμα 5.6. Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Επιλέγεται καινούργιος τυχαίος συμπλέκτης μήκους 1000 για κάθε λέξη που μεταδίδεται.

Πίνακας 5.3. Αριθμός επαναλήψεων αποκωδικοποίησης των κωδικών στο Σχ. 5.6

code	1	2	3	4	5	6	7	8	9	10
(2,1,5)	4 (3,4)	7 (2,9)	10 (2,2)	10 (1,9)						
(2,1,4)	5 (4,3)	10 (4,1)	10 (2,8)	10 (2,2)	10 (2)					
(2,1,3)	4 (3,9)	9 (5,4)	10 (3,9)	10 (2,9)	10 (2,4)	7 (2,1)				
(2,1,2)	3 (3)	8 (7)	10 (6,5)	10 (4,6)	10 (3,6)	8 (2,9)	8 (2,5)	6 (2,18)	5 (2)	4 (1,9)
(3,2,5)	3 (2,9)	6 (3,9)	10 (2,9)	10 (2,2)	10 (1,9)	10 (1,6)				
(3,2,4)	3 (2,9)	8 (4)	10 (2,8)	10 (2,2)	10 (1,9)	10 (1,7)				
(3,2,3)	3 (3)	6 (4,9)	10 (4,4)	10 (3)	10 (2,5)	8 (2,1)	8 (1,9)	6 (1,7)	5 (1,4)	
(3,2,2)	5 (4,8)	6 (5,2)	10 (4,9)	10 (3,6)	9 (2,8)	7 (2,4)	6 (2,1)	6 (1,9)	4 (1,7)	4 (1,5)
(4,3,5)	3 (2,9)	7 (3,3)	10 (2,4)	10 (1,9)	10 (1,6)					
(4,3,4)	5 (4,4)	6 (4,2)	10 (3,2)	10 (2,4)	10 (2)	10 (1,8)				
(4,3,3)	5 (4,7)	8 (5,4)	9 (3,6)	10 (2,8)	10 (2,3)	8 (1,9)	7 (1,7)	5 (1,5)		
(5,4,5)	4 (3,6)	7 (3,6)	10 (2,6)	10 (1,9)	10 (1,5)	10 (1,3)				
(5,4,4)	2 (2)	6 (4,5)	8 (3,7)	10 (2,6)	10 (2)	10 (1,7)				
(5,4,3)	2 (2)	6 (5,1)	6 (4,6)	6 (3,5)	10 (3)	7 (2,4)	6 (2)	5 (1,8)	5 (1,5)	
(5,4,2)	2 (2)	6 (5,2)	6 (4,6)	6 (3,5)	10 (3)	7 (2,4)	6 (2)	5 (1,8)	5 (1,5)	
(6,5,5)	4 (3,3)	8 (3,7)	10 (2,4)	10 (1,8)	10 (1,5)					
(6,5,4)	3 (2,9)	5 (3,9)	10 (3,7)	10 (2,5)	10 (1,9)	10 (1,6)				
(6,5,3)	3 (2,9)	6 (4,7)	10 (3,9)	10 (2,7)	10 (2,1)	10 (1,8)	8 (1,5)	8 (1,3)	7 (1,14)	
(6,5,2)	3 (2,9)	4 (3,4)	7 (3,4)	8 (2,8)	8 (2,2)	8 (1,9)	6 (1,6)	6 (1,4)	4 (1,2)	



(α)

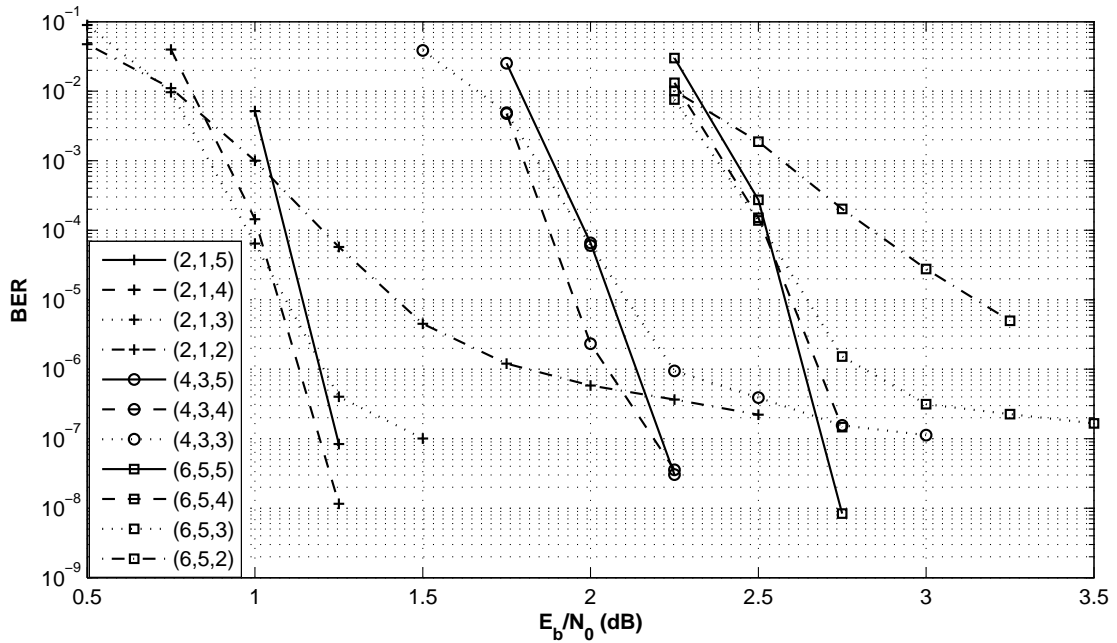


(β)

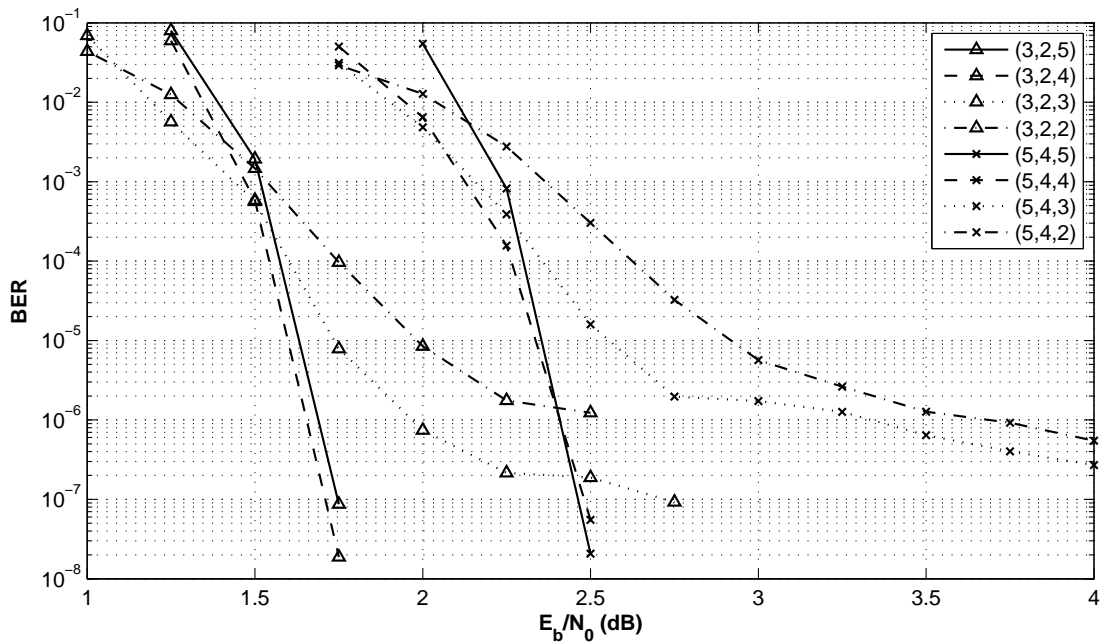
Σχήμα 5.7. Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Χρησιμοποιείται τυχαίος συμπλέκτης μήκους 10000.

Πίνακας 5.4. Αριθμός επαναλήψεων αποκωδικοποίησης των κωδίκων στο Σχ. 5.7

code	1	2	3	4	5	6	7	8	9	10
(2,1,5)	4 (3,7)	4 (2,9)								
(2,1,4)	4 (4)	4 (3,7)	4 (3)							
(2,1,3)	4 (4)	4 (3,9)	4 (3,9)	4 (3,1)	4 (2,9)	4 (2,4)				
(2,1,2)	4 (3,9)	4 (3,8)	4 (3,6)	4 (3,2)	4 (2,8)	4 (2,4)	4 (2,1)	4 (2)	4 (1,8)	
(3,2,5)	3 (3)	4 (3,6)	4 (2,9)							
(3,2,4)	3 (3)	4 (3,6)	4 (2,9)							
(3,2,3)	4 (4)	4 (3,7)	4 (3)	4 (2,8)	4 (2,5)	4 (2,1)	4 (2)			
(3,2,2)	4 (4)	4 (4)	4 (3,9)	4 (3,8)	4 (3,3)	4 (2,9)	4 (2,5)	4 (2,1)	3 (2)	3(2)
(4,3,5)	4 (3,9)	4 (3,1)	4 (2,4)							
(4,3,4)	4 (4)	4 (3,7)	4 (3)							
(4,3,3)	4 (4)	4 (3,9)	4 (3,7)	4 (3,1)	4 (2,6)	4 (2,2)	4 (2)			
(5,4,5)	3 (3)	4 (3,3)	4 (2,4)							
(5,4,4)	2 (2)	4 (4)	4 (3,9)	4 (3,4)	4 (3)	4 (2,9)				
(5,4,3)	2 (2)	4 (4)	4 (3,9)	4 (3,5)	4 (3,1)	4 (2,8)	4 (2,4)	3 (2,3)	3 (2,2)	3 (2)
(5,4,2)	2 (2)	4 (4)	4 (4)	4 (3,9)	4 (3,5)	4 (3)	4 (2,5)	4 (2,2)	4 (2)	4 (1,9)
(6,5,5)	2 (2)	4 (3,2)	4 (2,4)							
(6,5,4)	2 (2)	4 (3,5)	4 (2,8)							
(6,5,3)	2 (2)	4 (3,9)	4 (3,7)	4 (3)	4 (2,4)	4 (2,1)				
(6,5,2)	2 (2)	4 (4)	4 (3,9)	4 (3,8)	4 (3,3)	4 (2,8)	4 (2,4)	4 (2,1)	4 (2)	



(α')



(β')

Σχήμα 5.8. Προσομοιώσεις τούρμπο κωδίκων που χρησιμοποιούν τους συγκεραστικούς κωδικοποιητές του Πίνακα 5.1. Επιλέγεται καινούργιος τυχαίος συμπλέκτης μήκους 10000 για κάθε λέξη που μεταδίδεται.

Πίνακας 5.5. Αριθμός επαναλήψεων αποκωδικοποίησης των κωδικών στο Σχ. 5.8

code	1	2	3	4	5	6	7	8	9	10
(2,1,5)	4 (3,7)	4 (2,9)								
(2,1,4)	4 (4)	4 (3,7)	4 (3)							
(2,1,3)	4 (4)	4 (4)	4 (3,9)	4 (3,8)	4 (2,9)					
(2,1,2)	4 (4)	4 (4)	4 (3,9)	4 (3,8)	4 (3,3)	4 (3)	4 (2,6)	4 (2,2)	4 (2)	
(3,2,5)	2 (2)	4 (3,6)	4 (2,9)							
(3,2,4)	2 (2)	4 (3,6)	4 (2,9)							
(3,2,3)	2 (2)	4 (4)	4 (3,6)	4 (3)	4 (2,9)	4 (2,4)	4 (2,1)	4 (2)		
(3,2,2)	2 (2)	4 (4)	4 (3,9)	4 (3,8)	4 (3,3)	4 (2,9)	4 (2,5)			
(4,3,5)	4 (3,9)	4 (3,1)	4 (2,4)							
(4,3,4)	4 (3,9)	4 (3,4)	4 (2,7)							
(4,3,3)	4 (4)	4 (4)	4 (3,7)	4 (3,4)	4 (3,1)	4 (2,8)	4 (2,6)			
(5,4,5)	3 (3)	4 (3,3)	4 (2,4)							
(5,4,4)	3 (3)	4 (4)	4 (3,8)	4 (3,2)						
(5,4,3)	3 (3)	4 (4)	4 (3,9)	4 (3,4)	4 (3)	4 (2,8)	4 (2,5)	4 (2,2)	4 (1,8)	3 (1,5)
(5,4,2)	3 (3)	4 (4)	4 (4)	4 (3,9)	4 (3,4)	4 (3)	4 (2,5)	4 (2,2)	3 (2)	
(6,5,5)	2 (2)	4 (3,2)	4 (2,4)							
(6,5,4)	2 (2)	4 (3,5)	4 (2,9)							
(6,5,3)	2 (2)	4 (3,7)	4 (3)	4 (2,4)	4 (2,1)	3 (2)				
(6,5,2)	2 (2)	4 (3,9)	4 (3,8)	4 (2,8)	4 (2,3)					

Πίνακας 5.6. Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1, όταν χρησιμοποιείται ο τυχαίος συμπλεκτής μήκους 1000 των προσομοιώσεων στο Σχ. 5.5.

E_b/N_0 (dB)	1	1.25	1.5	1.75	2	2.25	2.5	2.75	3
$P_b \leq 10^{-3}$	(2, 1, 2) 9	(2, 1, 2) 5	(2, 1, 2) 3	(2, 1, 2) 3 (3, 2, 2) 5	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 17	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 4	(3, 2, 2) 2 (4, 3, 3) 3 (5, 4, 3) 5	(3, 2, 2) 2 (5, 4, 2) 4 (6, 5, 2) 6	(3, 2, 2) 2 (6, 5, 2) 3
$P_b \leq 10^{-4}$		(2, 1, 2) 13	(2, 1, 2) 5	(2, 1, 2) 4	(2, 1, 2) 3 (3, 2, 2) 7	(2, 1, 2) 3 (3, 2, 2) 4 (4, 3, 3) 11	(3, 2, 2) 3 (4, 3, 3) 4	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 3 (5, 4, 3) 7	(3, 2, 2) 2 (4, 3, 3) 2 (5, 4, 2) 7 (6, 5, 3) 6
$P_b \leq 10^{-5}$			(2, 1, 3) 6	(2, 1, 3) 5	(2, 1, 2) 6 (3, 2, 3) 12	(2, 1, 2) 3 (3, 2, 3) 5	(2, 1, 2) 3 (3, 2, 3) 3 (4, 3, 3) 11	(2, 1, 2) 3 (3, 2, 3) 3 (4, 3, 4) 3 (5, 4, 4) 17	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 3 (5, 4, 4) 6
$P_b \leq 10^{-6}$			(2, 1, 3) 18	(2, 1, 4) 7	(2, 1, 3) 6	(2, 1, 3) 3 (3, 2, 4) 8	(2, 1, 3) 3 (3, 2, 4) 6	(2, 1, 3) 3 (3, 2, 4) 3 (4, 3, 4) 7	(2, 1, 2) 3 (3, 2, 3) 3 (4, 3, 4) 3
$P_b \leq 10^{-7}$					(2, 1, 4) 8	(2, 1, 4) 3	(2, 1, 4) 3 (3, 2, 4) 10	(2, 1, 3) 3 (3, 2, 4) 7 (4, 3, 4) 18	(2, 1, 3) 3 (3, 2, 4) 3 (4, 3, 4) 7

Μετά από κάθε (n', k', m') κώδικα αναγράφεται το πλήθος των απαιτούμενων επαναλήψεων προκειμένου να επιτευχθεί η συγκεκριμένη πιθανότητα σφάλματος για τη συγκεκριμένη τιμή του E_b/N_0 .

Πίνακας 5.7. Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1, όταν χρησιμοποιείται ο τυχαίος συμπλέκτης μήκους 1000 των προσομοιώσεων στο Σχ. 5.5 (συνέχεια).

E_b/N_0 (dB)	3.25	3.5	3.75	4	4.25	4.5	4.75	5
$P_b \leq 10^{-3}$	(6, 5, 2) 2	(2, 1, 2) 1 (6, 5, 2) 2	(3, 2, 2) 1 (6, 5, 2) 2	(5, 4, 2) 1 (6, 5, 2) 2	(6, 5, 2) 1	(6, 5, 2) 1	(6, 5, 2) 1	(6, 5, 2) 1
$P_b \leq 10^{-4}$	(3, 2, 2) 2 (5, 4, 2) 4 (6, 5, 3) 3	(3, 2, 2) 2 (5, 4, 2) 3 (6, 5, 2) 4	(5, 4, 2) 2 (6, 5, 2) 3	(6, 5, 2) 2	(2, 1, 2) 1 (6, 5, 2) 2	(3, 2, 2) 1 (6, 5, 2) 2	(6, 5, 2) 2	(6, 5, 2) 1
$P_b \leq 10^{-5}$	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 3 (6, 5, 3) 6	(2, 1, 2) 2 (4, 3, 3) 2 (5, 4, 3) 3 (6, 5, 3) 4	(3, 2, 2) 2 (5, 4, 2) 3 (6, 5, 3) 3	(3, 2, 2) 2 (6, 5, 2) 3	(3, 2, 2) 2 (6, 5, 2) 3	(6, 5, 2) 2	(6, 5, 2) 2	(2, 1, 2) 1 (6, 5, 2) 2
$P_b \leq 10^{-6}$	(2, 1, 2) 3 (3, 2, 3) 3 (4, 3, 4) 3 (5, 4, 4) 8	(2, 1, 2) 2 (3, 2, 3) 2 (4, 3, 4) 2 (5, 4, 4) 3 (6, 5, 4) 6	(2, 1, 2) 2 (3, 2, 2) 3 (5, 4, 4) 2 (6, 5, 3) 7	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 3 (6, 5, 3) 4	(3, 2, 2) 2 (4, 3, 3) 2 (6, 5, 3) 3	(3, 2, 2) 2	(3, 2, 2) 2 (6, 5, 3) 2	(3, 2, 2) 2 (6, 5, 2) 3
$P_b \leq 10^{-7}$	(2, 1, 3) 3 (3, 2, 4) 3 (4, 3, 4) 7 (5, 4, 5) 11	(2, 1, 3) 2 (3, 2, 4) 2 (4, 3, 4) 3 (5, 4, 5) 6 (6, 5, 5) 13	(2, 1, 3) 2 (3, 2, 4) 2 (4, 3, 4) 3 (5, 4, 4) 6 (6, 5, 5) 6	(2, 1, 3) 2 (4, 3, 4) 2 (5, 4, 5) 2 (6, 5, 5) 6	(2, 1, 2) 2 (3, 2, 3) 2 (5, 4, 4) 2 (6, 5, 5) 2	(2, 1, 2) 2 (3, 2, 3) 2 (5, 4, 4) 2 (6, 5, 5) 2	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 2 (6, 5, 4) 2	(3, 2, 2) 2 (6, 5, 3) 2

Μετά από κάθε (n', k', m') κώδικα αναγράφεται το πλήθος των απαιτούμενων επαναλήψεων προκειμένου να επιτευχθεί η συγκεκριμένη πιθανότητα σφάλματος για τη συγκεκριμένη τιμή του E_b/N_0 .

Πίνακας 5.8. Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1. Χρησιμοποιείται διαφορετικός τυχαίος συμπλέκτης μήκους 1000 για κάθε μεταδιδόμενη λέξη.

E_b/N_0 (dB)	1	1.25	1.5	1.75	2	2.25	2.5	2.75	3
$P_b \leq 10^{-3}$	(2, 1, 2) 9	(2, 1, 2) 5	(2, 1, 2) 3	(2, 1, 2) 3 (3, 2, 2) 5	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 16	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 4	(3, 2, 2) 2 (5, 4, 2) 6	(3, 2, 2) 2 (5, 4, 2) 3 (6, 5, 2) 6	(3, 2, 2) 2 (6, 5, 2) 3
$P_b \leq 10^{-4}$		(2, 1, 2) 10	(2, 1, 2) 5	(2, 1, 2) 4 (3, 2, 2) 10	(2, 1, 2) 3 (3, 2, 2) 8	(2, 1, 2) 2 (3, 2, 2) 4	(3, 2, 2) 3 (4, 3, 3) 4	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 3 (5, 4, 3) 8	(3, 2, 2) 2 (4, 3, 3) 2 (5, 4, 3) 4 (6, 5, 3) 6
$P_b \leq 10^{-5}$			(2, 1, 3) 6	(2, 1, 3) 4	(2, 1, 2) 6 (3, 2, 3) 11	(2, 1, 2) 4 (3, 2, 3) 5	(2, 1, 2) 3 (3, 2, 3) 3 (4, 3, 4) 9	(2, 1, 2) 3 (3, 2, 2) 5 (4, 3, 3) 6 (5, 4, 4) 15	(2, 1, 2) 2 (3, 2, 2) 3 (4, 3, 3) 6 (5, 4, 4) 4
$P_b \leq 10^{-6}$				(2, 1, 3) 8	(2, 1, 3) 6	(2, 1, 3) 3 (3, 2, 4) 13	(2, 1, 3) 3 (3, 2, 4) 6	(2, 1, 3) 3 (3, 2, 4) 3 (4, 3, 4) 11	(2, 1, 2) 3 (3, 2, 3) 3 (4, 3, 4) 4
$P_b \leq 10^{-7}$					(2, 1, 4) 9	(2, 1, 4) 5	(2, 1, 4) 3 (3, 2, 5) 11	(2, 1, 3) 3 (3, 2, 4) 7	(3, 2, 3) 3 (4, 3, 4) 7

Μετά από κάθε (n', k', m') κώδικα αναγράφεται το πλήθος των απαιτούμενων επαναλήψεων προκειμένου να επιτευχθεί η συγκεκριμένη πιθανότητα σφάλματος για τη συγκεκριμένη τιμή του E_b/N_0 .

Πίνακας 5.9. Βέλτιστος τρόπος χρήσης της κωδικής οικογένειας στον Πίνακα 5.1. Χρησιμοποιείται διαφορετικός τυχαίος συμπλέκτης μήκους 1000 για κάθε μεταδιδόμενη λέξη (συνέχεια).

E_b/N_0 (dB)	3.25	3.5	3.75	4	4.25	4.5	4.75	5
$P_b \leq 10^{-3}$	(6, 5, 2) 2	(2, 1, 2) 1 (6, 5, 2) 2	(3, 2, 2) 1 (6, 5, 2) 2	(3, 2, 2) 1 (6, 5, 2) 2	(6, 5, 2) 1	(6, 5, 2) 1	(6, 5, 2) 1	(6, 5, 2) 1
	$P_b \leq 10^{-4}$	(3, 2, 2) 2	(3, 2, 2) 2	(3, 2, 2) 2	(6, 5, 2) 2	(2, 1, 2) 1	(3, 2, 2) 1	(6, 5, 2) 2
(5, 4, 2) 3		(6, 5, 2) 3	(6, 5, 2) 3		(6, 5, 2) 2	(6, 5, 2) 2		
(6, 5, 3) 3								
$P_b \leq 10^{-5}$	(2, 1, 2) 2	(3, 2, 2) 2	(3, 2, 2) 2	(3, 2, 2) 2	(3, 2, 2) 2	(6, 5, 2) 2	(6, 5, 2) 2	(2, 1, 2) 1
	(3, 2, 2) 3	(4, 3, 3) 2	(5, 4, 2) 4	(6, 5, 3) 2	(6, 5, 2) 3			(6, 5, 2) 2
	(4, 3, 3) 3	(5, 4, 3) 3	(6, 5, 3) 3					
	(5, 4, 4) 3	(6, 5, 3) 4						
	(6, 5, 3) 10							
$P_b \leq 10^{-6}$	(2, 1, 2) 2	(2, 1, 2) 2	(2, 1, 2) 2	(2, 1, 2) 2	(3, 2, 2) 2	(3, 2, 2) 2	(3, 2, 2) 2	(3, 2, 2) 2
	(3, 2, 3) 3	(3, 2, 3) 2	(3, 2, 3) 2	(3, 2, 2) 3	(6, 5, 3) 3	(6, 5, 3) 2	(6, 5, 3) 2	(6, 5, 2) 3
	(4, 3, 4) 3	(4, 3, 4) 2	(4, 3, 3) 3	(4, 3, 3) 2				
	(5, 4, 4) 8	(5, 4, 4) 3	(5, 4, 4) 2	(6, 5, 4) 2				
	(6, 5, 4) 15	(6, 5, 4) 6	(6, 5, 4) 5					
$P_b \leq 10^{-7}$	(2, 1, 3) 2	(2, 1, 3) 2	(2, 1, 3) 2	(2, 1, 3) 2	(2, 1, 2) 2	(2, 1, 2) 2	(2, 1, 2) 2	(3, 2, 2) 2
	(3, 2, 4) 3	(3, 2, 4) 2	(3, 2, 4) 2	(5, 4, 4) 2	(3, 2, 3) 2	(3, 2, 3) 2	(3, 2, 3) 2	(6, 5, 3) 2
	(4, 3, 4) 7	(4, 3, 4) 3	(4, 3, 4) 3	(6, 5, 4) 5	(5, 4, 4) 2	(6, 5, 4) 2	(4, 3, 3) 3	
	(5, 4, 5) 12	(5, 4, 5) 7	(5, 4, 4) 6		(6, 5, 5) 2		(6, 5, 4) 2	

Μετά από κάθε (n', k', m') κώδικα αναγράφεται το πλήθος των απαιτούμενων επαναλήψεων προκειμένου να επιτευχθεί η συγκεκριμένη πιθανότητα σφάλματος για τη συγκεκριμένη τιμή του E_b/N_0 .

6. ΣΥΝΟΨΗ-ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Η παρούσα διατριβή πραγματεύτηκε θέματα σχεδίασης συγκεραστικών κωδίκων οι οποίοι θα χρησιμοποιηθούν είτε ως αυτόνομα σχήματα κωδικοποίησης, είτε ως δομικά στοιχεία των τούρμπο κωδίκων, με γνώμονα την πολυπλοκότητα του διαγράμματος trellis, η οποία επηρεάζει άμεσα την πολυπλοκότητα όλων των σημαντικών αλγορίθμων αποκωδικοποίησης. Τα νέα αποτελέσματα που απορρέουν από τη συγκεκριμένη εργασία συνοψίζονται στη συνέχεια:

Νέοι συγκεραστικοί κώδικες χαμηλής πολυπλοκότητας: Παρουσιάστηκε μια οικογένεια συγκεραστικών κωδίκων χαμηλής πολυπλοκότητας, οι οποίοι παράγονται τροποποιώντας το trellis module των PCC, καθώς και μια αποδοτική τεχνική για την κατασκευή του ελάχιστου trellis module. Επίσης, αναπτύχθηκαν αρκετοί κώδικες της προτεινόμενης κατηγορίας, οι οποίοι σε πολλές περιπτώσεις έχουν πολύ καλύτερη σχέση απόδοσης-πολυπλοκότητας συγκριτικά με τους PCC, καθώς μία μικρή αύξηση της πολυπλοκότητας μπορεί να αυξήσει την ελεύθερη απόσταση του κώδικα. Συγκεκριμένοι περιορισμοί και ιδιότητες στο σχεδιασμό των γεννητόρων περιορίζουν το χώρο αναζήτησης, με αποτέλεσμα να είναι δυνατή η αναζήτηση κωδίκων πολύ μεγαλύτερης ελάχιστης πολυπλοκότητας (και μνήμης), απ' ό,τι στην περίπτωση των GPCC [33] και των κωδίκων στα [29] και [30]. Ακόμα, αποδείχθηκαν ικανές και αναγκαίες συνθήκες που καθιστούν το κατασκευαζόμενο διάγραμμα trellis βέλτιστο για την υποκατηγορία των $(n, n - 1)$ συγκεραστικών κωδίκων. Τέλος, αναπτύχθηκε ένα άνω φράγμα για την ελάχιστη απόσταση των εν λόγω κωδίκων.

Κωδικά σχήματα μεταβλητού ρυθμού και πολυπλοκότητας: Περιγράφηκε μία μέθοδος για την κατασκευή ευέλικτων συγκεραστικών κωδίκων μεταβλητού ρυθμού και πολυπλοκότητας, η οποία συνδυάζει τις τεχνικές απαλοιφής κωδικών bit και απαλοιφής μονοπατιών του διαγράμματος trellis. Μελετήθηκαν οι επιπτώσεις των εν λόγω τεχνικών στο προφίλ πολυπλοκότητας του μητρικού διαγράμματος trellis και παρουσιάστηκε μια μέθοδος ανάπτυξης οικογενειών ευέλικτων κωδικοποιητών συγκεκριμένης πολυπλοκότητας. Κατασκευάστηκαν τρεις οικογένειες από μητρικούς κώδικες ρυθμού $1/2$, $1/3$ και $1/4$, και μνήμης $m = 8$. Τα φάσματα των κατασκευαζόμενων κωδίκων είναι συγκρίσιμα με τα αντίστοιχα των PCC μεταβλητού ρυθμού, γεγονός που σημαίνει ότι η επιπρόσθετη δυνατότητα μεταβολής της υπολογιστικής πολυπλοκότητας της αποκωδικοποίησης δεν μειώνει την απόδοση.

Αναδρομικοί ευέλικτοι κωδικοποιητές για τούρμπο κώδικες: Η έρευνα στους ευέλικτους κωδικοποιητές επεκτάθηκε και στην περίπτωση των τούρμπο κωδίκων. Αναλύθηκε η εφαρμογή της τεχνικής απαλοιφής μονοπατιών σε αναδρομικούς συγκεραστικούς κωδικοποιητές, ενώ σχεδιάστηκε και μια κωδική οικογένεια χρησιμοποιώντας έναν μητρικό κωδικοποιητή ρυθμού $1/2$

και μνήμης $m = 5$, κατάλληλη για να χρησιμοποιηθεί σε τούρμπο κώδικες. Οι συγκεκριμένοι κώδικες έχουν σχεδόν σε όλες τις περιπτώσεις την ίδια απόσταση d_2 με τους καλύτερους συστηματικούς αναδρομικούς PCC. Προσομιώσεις της συγκεκριμένης οικογένειας συγκεραστικών κωδικοποιητών χρησιμοποιώντας διάφορους συμπλέκτες δείχνουν ότι επιλέγοντας τον κατάλληλο πίνακα απαλοιφής κωδικών bit και την κατάλληλη είσοδο απαλοιφής μονοπατιών, η περιοχή καταρράκτη στην καμπύλη απόδοσης μπορεί να εμφανιστεί σε μικρότερη τιμή του σηματοθορυβικού λόγου, μειώνοντας ταυτόχρονα και την πολυπλοκότητα της αποκωδικοποίησης.

6.1 Μελλοντικές Κατευθύνσεις

Στη συνέχεια, παρατίθενται ερευνητικές κατευθύνσεις συναφείς με την παρούσα διατριβή:

- Στο κεφάλαιο 3 θεωρήσαμε ότι η διάσταση του χώρου καταστάσεων του trellis module των κατασκευαζόμενων κωδίκων παίρνει μόνο τις τιμές m και $m + 1$. Θα ήταν ενδιαφέρουσα η διερεύνηση των κωδίκων και των trellis modules που θα προέκυπταν αν διάσταση του χώρου καταστάσεων μπορούσε να πάρει κι άλλες τιμές.
- Η κατασκευή συγκεραστικών κωδίκων χαμηλής trellis πολυπλοκότητας κατάλληλων για τούρμπο κώδικες είναι σημαντική, καθώς η αποκωδικοποίηση των τούρμπο κωδίκων είναι αρκετά πολύπλοκη. Κάποια πρώτα αποτελέσματα προς αυτήν την κατεύθυνση παρουσιάζονται στο [71].
- Ενδιαφέρον έχει επίσης η κατασκευή κωδίκων χαμηλής πολυπλοκότητας όπως αυτοί στο κεφάλαιο 3 με την επιπρόσθετη δυνατότητα της μεταβολής του ρυθμού.
- Το άνω φράγμα που παρουσιάζεται στο Θεώρημα 3.10 δεν εξαρτάται από τα ιδιαίτερα χαρακτηριστικά του κώδικα, παρά μόνο από τις διαστάσεις των χώρων καταστάσεων και ακμών του trellis module, καθώς και από την κατανομή των κωδικών bit. Έτσι, θα μπορούσε να φανεί χρήσιμο και σε άλλες κατηγορίες συγκεραστικών κωδίκων, όπως π.χ. τους GPCC και τους κώδικες στα [29] και [30].
- Η τεχνική της απαλοιφής μονοπατιών του διαγράμματος trellis χρησιμοποιείται με πολύ συγκεκριμένο τρόπο στο κεφάλαιο 4. Μια παραλλαγή θα μπορούσε ενδεχομένως να οδηγήσει σε καλύτερους κώδικες.
- Οι Ευέλικτοι κώδικες που αναπτύχθηκαν στο κεφάλαιο 4 έχουν την πολυπλοκότητα των PCC, προκειμένου να μπορεί να γίνει άμεσα σύγκριση με τους PCC μεταβλητού ρυθμού. Ενδιαφέρον θα είχαν και διαφορετικές προσεγγίσεις, όπως για παράδειγμα να επιλέγεται σαν μέλος της οικογένειας ο λιγότερο πολύπλοκος κώδικας που επιτυγχάνει κάθε τιμή

της ελεύθερης απόστασης, βελτιστοποιώντας έτσι τη σχέση πολυπλοκότητας-απόδοσης για ασυμπτωτικά μεγάλες τιμές του σηματοθορυβικού λόγου.

- Όσον αφορά τους ευέλικτους κωδικοποιητές για τούρμπο κώδικες, είναι σημαντική η αναζήτηση συμπλέκτη που θα οδηγεί σε αυξημένη απόδοση κάθε μέλος της κωδικής οικογένειας. Διαφορετικά, θα μπορούσε να εξεταστεί η χρήση πολυωνύμων μετάθεσης (permutation polynomials), όπως στην περίπτωση του τούρμπο κώδικα του προτύπου LTE, έτσι ώστε ο κάθε κώδικας της οικογένειας να έχει και τον δικό του βελτιστοποιημένο συμπλέκτη χωρίς ιδιαίτερη αύξηση της πολυπλοκότητας, αφού για κάθε πολώνυμο (δευτέρου βαθμού) χρειάζεται να αποθηκευθούν μόνο οι τρεις ακέραιοι συντελεστές του.

ΑΝΑΦΟΡΕΣ

- [1] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.
- [2] R. W. Hamming, “Error detecting and error correcting codes,” *Bell System Technical Journal*, vol. 29, pp. 147–160, 1950.
- [3] M. J. E. Golay, “Notes on digital coding,” *Proc. IRE*, vol. 37, p. 657, 1949.
- [4] D. E. Muller, “Application of Boolean algebra to switching circuit design and to error detection,” *IRE Trans. Electron. Comput.*, vol. EC-3, pp. 6–12, Sep. 1954.
- [5] A. Hocquenghem, “Codes correcteurs d’erreurs,” *Chiffres*, vol. 2, pp. 147–156, 1959.
- [6] J. L. Massey, “Shift-register synthesis and BCH decoding,” *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122–127, 1969.
- [7] I. S. Reed, “A class of multiple error correcting codes and the decoding scheme,” *IRE Trans. Inform. Theory*, vol. IT-4, pp. 38–49, Sep. 1954.
- [8] P. Elias, “Coding for noisy channels,” *IRE Conv. Rec.*, vol. 4, pp. 37–46, Mar. 1955.
- [9] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.
- [10] J. B. Cain, J. C. Clark Jr, and J. M. Geist, “Punctured convolutional codes of rate $(n - 1)/n$ and simplified maximum likelihood decoding,” *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 97–100, Jan. 1979.
- [11] Y. Yasuda, K. Kashiki, and Y. Hirata, “High-rate punctured convolutional codes for soft decision Viterbi decoding,” *IEEE Trans. Commun.*, vol. COM-32, pp. 315–319, Mar. 1984.
- [12] P. J. Lee, “Constructions of rate $(n - 1)/n$ punctured convolutional codes with minimal required SNR criterion,” *IEEE Trans. Commun.*, vol. COM-36, pp. 1171–1173, Oct. 1988.
- [13] G. Begin, D. Haccoun, and C. Paquin, “Further results on high-rate punctured convolutional codes for Viterbi and sequential decoding,” *IEEE Trans. Commun.*, vol. COM-38, pp. 1922–1928, Nov. 1990.
- [14] K. J. Hole, “Punctured convolutional codes for the 1-D partial-response channel,” *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 808–817, May 1991.

- [15] I. E. Bocharova and B. D. Kudryashov, "Rational rate punctured convolutional codes for soft-decision Viterbi decoding," *IEEE Trans. Inform. Theory*, vol. IT-43, pp. 1305–1313, 1997.
- [16] A. Dholakia, *Introduction to convolutional codes with applications*. Kluwer Academic Publishers, 1994.
- [17] S. Lin and D. J. Costello, *Error control coding*. Prentice Hall, 2004.
- [18] C. R. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. Shellhamer, and W. Caldwell, "IEEE 802.22: The first cognitive radio wireless regional area network standard," *IEEE Commun. Mag.*, vol. 47, pp. 130–138, Jan. 2009.
- [19] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. COM-36, pp. 389–400, Apr. 1988.
- [20] C. H. Wang and C. C. Chao, "Path-compatible pruned convolutional (PCPC) codes: a new scheme for unequal error protection," in *IEEE ISIT'98*, Aug. 1998, p. 306.
- [21] —, "Path-compatible pruned convolutional (PCPC) codes," *IEEE Trans. Commun.*, vol. COM-50, pp. 213–224, Feb. 2002.
- [22] C. H. Wang and Y. H. Lin, "Combined puncturing and path pruning for convolutional codes and the application to unequal error protection," *IEEE Trans. Commun.*, vol. COM-56, pp. 1385–1389, Sep. 2008.
- [23] L. R. Bahl, J. Cocke, G. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.
- [24] R. J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 1072–1092, July 1996.
- [25] R. J. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 1855–1864, Nov. 1996.
- [26] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 1663–1666, Sept. 1994.
- [27] P. CharnKeitKong, H. Imai, and K. Yamaguchi, "On classes of rate $k/(k+1)$ convolutional codes and their decoding techniques," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 2181–2193, Nov. 1996.

- [28] E. Rosnes and Ø. Ytrehus, “On maximum length convolutional codes under a trellis complexity constraint,” *Journal of Complexity*, vol. 20, no. 372-408, March-June 2004.
- [29] H. H. Tang and M. C. Lin, “On $(n, n - 1)$ convolutional codes with low trellis complexity,” *IEEE Trans. Commun.*, vol. COM-50, pp. 37–47, Jan. 2002.
- [30] H. H. Tang, M. C. Lin, and B. F. Uchoa-Filho, “Minimal trellis modules and equivalent convolutional codes,” *IEEE Trans. Inform. Theory*, vol. IT-52, pp. 3738–3746, Aug. 2006.
- [31] B. F. Uchoa-Filho, R. D. Souza, C. Pimentel, and M. C. Lin, “Generalized punctured convolutional codes,” *IEEE Com. Letters*, vol. 9, pp. 1070–1072, Dec. 2005.
- [32] B. F. Uchoa-Filho, R. D. Souza, C. Pimentel, and M. Jar, “Further results on convolutional codes based on a minimal trellis complexity measure,” in *2006 IEEE Intern. Telecom. Symp.*, Sept. 2006, pp. 123–128.
- [33] —, “Convolutional codes under a minimal trellis complexity measure,” *IEEE Trans. Commun.*, pp. 1–5, Jan. 2009.
- [34] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo codes,” in *IEEE ICC’93*, May 1993, pp. 1064–1070.
- [35] S. Benedetto and G. Montorsi, “Unveiling turbo codes: some results on parallel concatenated coding,” *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 409–428, Mar. 1996.
- [36] —, “Design of parallel concatenated convolutional codes,” *IEEE Trans. Commun.*, vol. COM-44, pp. 591–600, May 1996.
- [37] D. Divsalar and R. J. McEliece, “Effective free distance of turbo codes,” *Electron. Lett.*, vol. 32, Feb. 1996.
- [38] S. Benedetto, R. Garello, and G. Montorsi, “A search for good convolutional codes to be used in the construction of turbo codes,” *IEEE Trans. Commun.*, vol. COM-46, pp. 1101–1105, Sep. 1998.
- [39] O. F. Acikel and W. E. Ryan, “Punctured turbo codes for BPSK/QPSK channels,” *IEEE Trans. Commun.*, vol. COM-47, pp. 1315–1323, Sep. 1999.
- [40] M. A. Kousa and A. H. Mugaibel, “Puncturing effects on turbo codes,” *IEE Proc. Commun.*, vol. 149, pp. 132–138, June 2002.
- [41] F. Babich, G. Montorsi, and F. Vatta, “On rate compatible punctured turbo codes design,” *EURASIP Journal Appl. Sig. Processing*, vol. 6, pp. 784–794, 2005.

- [42] R. Garello, P. Pierleoni, and S. S. Benedetto, “Computing the free distance of turbo codes and serially concatenated codes with interleavers: algorithms and applications,” *IEEE Journal Sel. Areas Commun.*, vol. 19, pp. 800–812, May 2001.
- [43] B. Shen, A. Patapoutian, and P. A. McEwen, “Punctured recursive convolutional encoders and their applications in turbo codes,” *IEEE Trans. Inform. Theory*, vol. IT-47, pp. 2300–2320, Sep. 2001.
- [44] F. Daneshgaran, M. Laddomada, and M. Mondin, “High-rate recursive convolutional codes for concatenated channel codes,” *IEEE Trans. Commun.*, vol. COM-52, pp. 1846–1850, Nov. 2004.
- [45] J. G. Proakis, *Digital communications*. McGraw-Hill, New York, 2000.
- [46] S. Benedetto and E. Biglieri, *Principles of digital transmission*. Springer, 1999.
- [47] G. D. Forney Jr, “Convolutional codes I: Algebraic structure,” *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.
- [48] R. Johannesson and K. Zigangirov, *Fundamentals of convolutional coding*. IEEE press, 1999.
- [49] R. J. McEliece, “The algebraic theory of convolutional codes,” in *Handbook of Coding Theory*, V. Pless and W. Huffman, Eds. Elsevier, 1998, pp. 1065–1138.
- [50] J. A. Heller, “Short constraint length convolutional codes,” Jet Propulsion Lab. California Inst. Technol., Pasadena, Tech. Rep., 1968.
- [51] J. H. Griesmer, “A bound for error correcting codes,” IBM J. Res. Develop., Tech. Rep., 1960.
- [52] K. L. Larsen, “Short convolutional codes with maximal free distance for rates $1/2$, $1/3$ and $1/4$,” *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 371–372, May 1973.
- [53] R. Johannesson and E. Paaske, “Further results on binary convolutional codes with an optimum distance profile,” *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 264–268, Mar. 1978.
- [54] D. G. Daut, J. W. Modestino, and L. D. Wismer, “New short constraint length convolutional code constructions for selected rational rates,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 794–800, Sept. 1982.
- [55] P. J. Lee, “Further results on rate $1/N$ convolutional code constructions with minimum required SNR criterion,” *IEEE Trans. Commun.*, pp. 395–399, Apr. 1986.

- [56] J. J. Chang, D. J. Hwang, and M. C. Lin, “Some extended results on the search for good convolutional codes,” *IEEE Trans. Inform. Theory*, vol. IT-43, pp. 1682–1697, Sept. 1997.
- [57] Ø. Ytrehus, “Binary convolutional codes of high rate,” Depart. of Inform., Univ. of Bergen, Tech. Rep., 1992.
- [58] D. J. Johnsson, F. Bjarkeson, M. Hell, and F. Hug, “Searching for new convolutional codes using the cell broadband engine architecture,” *IEEE Com. Letters*, vol. 15, pp. 560–561, May 2011.
- [59] C. B. Schlegel and L. C. Perez, *Trellis and turbo coding*. Wiley-IEEE Press, 2004.
- [60] G. Begin and D. Haccoun, “High-rate punctured convolutional codes: structure properties and construction technique,” *IEEE Trans. Commun.*, pp. 1381–1385, 1989.
- [61] K. J. Hole, “New short constraint length rate $(n - 1)/n$ punctured convolutional codes for soft decision Viterbi decoding,” *IEEE Trans. Inform. Theory*, vol. 34, pp. 1979–1081, 1988.
- [62] M. Mooser, “Some periodic convolutional codes better than any fixed code,” *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 750–751, Sept. 1983.
- [63] R. Palazzo, Jr., “A time-varying convolutional encoder better than the best time invariant encoder,” *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 1109–1110, May 1993.
- [64] Q. Hu and L. C. Perez, “Some periodic time-varying convolutional codes with free distance achieving the Heller bound,” in *IEEE ISIT*, June 2001, p. 247.
- [65] G. D. Forney Jr, “The Viterbi algorithm,” *Proceedings IEEE*, vol. 61, pp. 268–278, 1973.
- [66] D. J. Muder, “Minimal trellises for block codes,” *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, 1988.
- [67] F. R. Kschischang and V. Sorokine, “On the trellis structure of block codes,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 1924–1937, 1996.
- [68] A. Vardy, “Trellis structure of codes,” in *Handbook of Coding Theory*, V. Pless and W. Huffman, Eds. Elsevier, 1998, pp. 1989–2117.
- [69] I. E. Bocharova, R. Johannesson, and B. D. Kudryashov, “Trellis complexity of short linear codes,” *IEEE Trans. Inform. Theory*, vol. IT-53, pp. 361–368, 2007.
- [70] W. Lin, R. J. McEliece, and M. Xu, “Trellis-canonical generator matrices for convolutional codes,” in *IEEE Int. Symp. Inf. Theory (ISIT)*, 1997, p. 286.

- [71] C. Pimentel, R. D. Souza, B. F. Uchoa-Filho, and I. Bencimol, “Minimal trellis for systematic recursive convolutional encoders,” in *IEEE ISIT*, June 2011, pp. 2478–2481.
- [72] P. H. Wu, “On the complexity of turbo decoding algorithms,” in *Veh. Tech. Conf.*, 2001, pp. 1439–1443.
- [73] B. U. Pedroni, V. A. Pedroni, and R. D. Souza, “Hardware implementation of viterbi decoder using the minimal trellis,” in *Int. Symp. Comm. Control and Sig. Proc. (ISCCSP)*, 2010.
- [74] C. Pimentel, R. D. Souza, B. F. Uchoa-Filho, and M. E. Pellenz, “Generalized punctured convolutional codes with unequal error protection,” *J. Advances in Sig. Proc.*, pp. 1–6, 2008.
- [75] A. Katsiotis, P. Rizomiliotis, and N. Kalouptsidis, “New constructions of low complexity convolutional codes,” in *Proc. of IEEE Int. Conf. on Com. (ICC'2009)*, June 2009.
- [76] —, “New constructions of high performance low complexity convolutional codes,” *IEEE Trans. Commun.*, vol. 58, pp. 1950–1961, July 2010.
- [77] J. Justesen, “New convolutional code constructions and a class of asymptotically good time-varying codes,” *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 220–225, Mar. 1973.
- [78] K. J. Hole, “Rate $k/(k + 1)$ minimal punctured convolutional encoders,” *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 653–655, May 1991.
- [79] C. O. Donoghue and C. Burkley, “Catastrophicity test for time-varying convolutional encoders,” *Cryptography and Coding 1999, LNCS*, vol. 1746, pp. 153–162, Jan. 1991.
- [80] N. Ogasahara, M. Kobayashi, and S. Hirasawa, “The construction of periodically time-variant convolutional codes using binary linear block codes,” *WILEY Electr. and Com in Japan*, vol. 90, pp. 31–40, Apr. 2007.
- [81] R. D. Souza and B. F. Uchoa-Filho, “On trellis modules for convolutional codes,” in *2002 IEEE ISIT*, July 2002, p. 417.
- [82] H. T. Moorthy, S. Lin, and G. T. Uehara, “Good trellises for IC implementation of Viterbi decoders for linear block codes,” *IEEE Trans. Commun.*, vol. 45, pp. 52–63, Jan. 1997.
- [83] I. Kang and A. N. Wilson Jr., “Low-power Viterbi decoder for CDMA mobile terminals,” *IEEE J. Solid-State Circuits*, vol. 33, pp. 473–482, Mar. 1998.
- [84] A. Lafourcade and A. Vardy, “Optimal sectionalization of a trellis,” *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 689–703, May 1996.

- [85] A. Katsiotis and N. Kalouptsidis, “On $(n, n - 1)$ punctured convolutional codes and their trellis modules,” *IEEE Trans. Commun.*, vol. 59, pp. 1213–1217, May 2011.
- [86] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [87] W. Xu and J. Romme, “A class of multirate convolutional codes by dummy bit insertion,” in *IEEE GLOBECOM'00*, Aug. 2000, pp. 830–834.
- [88] A. Katsiotis, P. Rizomiliotis, and N. Kalouptsidis, “Flexible convolutional codes: variable rate and complexity,” *IEEE Trans. Commun.*, March 2012.
- [89] J. Pons and P. Duvaut, “New criteria for determining improved perforation patterns for variable-rate convolutional codes,” in *IEEE ICC'05*, May 2005, pp. 423–427.
- [90] A. Katsiotis and N. Kalouptsidis, “Recursive flexible convolutional codes for parallel concatenation,” *IEEE Trans. Wireless Commun.*, submitted, 2012.
- [91] S. Benedetto, G. Montorsi, and D. Divsalar, “Concatenated convolutional codes with interleavers,” *IEEE Commun. Mag.*, pp. 102–109, Aug. 2003.
- [92] B. Vucetic and J. Yuan, *Turbo codes: principles and applications*. Kluwer Academic Publishers, 2000.
- [93] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, March 1996.
- [94] I. Chatzigeorgiou, M. Rodrigues, I. Wassell, and R. Carrasco, “Analysis and design of punctured rate-1/2 turbo codes exhibiting low error floors,” *IEEE Journal Sel. Areas Commun.*, vol. 27, pp. 944–953, 2009.
- [95] D. Divsalar and F. Pollara, “On the design of turbo codes,” TDA Progress Report 42-123, Tech. Rep., 1995.
- [96] A. Graell i Amat, G. Montorsi, and S. Benedetto, “Design and decoding of optimal high-rate convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 867–881, May 2004.
- [97] S. Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [98] J. Hagenauer, “The EXIT chart - introduction to extrinsic information transfer in iterative processing,” in *Europ. Signal Proc. Conf. (EUSIPCO)*, 2004.
- [99] Y. Saouter and C. Berrou, “Time varying convolutional codes for punctured turbo codes,” in *Wireless Telecom. Symp. (WTS)*, 2011.

[100] *Multiplexing and Channel Coding (Release 8)*, 3GPP Technical Specification 36.212 Std., 2008.

ΕΥΡΕΤΗΡΙΟ

- Ανάλυση μεταφοράς εξωτερικής πληροφορίας (EXIT), 138
- Απαρίθμηση βάρους
συνάρτηση, 45
συνάρτηση εισόδου-εξόδου, 44
υπό συνθήκη συνάρτηση τούρμπο κώδικα (CWEF), 137
- Αποκωδικοποιητής
αλγόριθμος BCJR, 47
αλγόριθμος Viterbi, 45
τούρμπο, 132
χαμηλή πολυπλοκότητα, 80
μέγιστη a posteriori πιθανότητα (MAP), 47
- Διαμόρφωση μετατόπισης φάσης (MPSK), 31
- Ελάχιστη απόσταση, 35
φράγμα Griesmer, 44
φράγμα Plotkin, 44
- Ελεύθερη απόσταση, 43
άνω φράγματα για νέους κώδικες, 91
φράγμα Heller, 43
- Κέρδος συμπλέκτη, 138
- Κανάλι επικοινωνίας
διακριτό χωρίς μνήμη (DMC), 32
δυαδικό συμμετρικό (BSC), 32
προσθετικός λευκός Γκαουσιανό θόρυβος (AWGN), 32
- Κωδικοποιητής
αναδρομικός ευέλικτος συγκεραστικός, 141
αναδρομικός συστηματικός, 39
μνήμη, 38
περιοδικός χρονικά μεταβαλλόμενος, 50
συγκεραστικός, 36
- Κώδικας
trellis module, 42
βέλτιστο, 85
τμηματοποιημένο, 72
γραμμικός, 34
διάγραμμα trellis, 41
ελάχιστο trellis module, 57, 69
ελάχιστο διάγραμμα trellis, 54
μπλοκ, 33
συγκεραστικός, 36
γενικευμένος διάτρητος (generalized punctured), 61
διάγραμμα καταστάσεων, 41
διάτρητος (punctured), 50
νέες προτεινόμενες κατασκευές, 76–78
περιοδικός χρονικά μεταβαλλόμενος, 50
τούρμπο
ομοιόμορφος συμπλέκτης, 137
περιοχή δαπέδου σφάλματος (error floor), 136
περιοχή καταρράκτη (waterfall region), 136
περιοχή μη σύγκλισης (non convergence region), 136
συμπλέκτης, 132
σχεδίαση, 136
φάσμα βάρους, 44
- Λογάριθμος λόγου πιθανοφάνειας (LLR), 47
αλγόριθμος BCJR, 48
κώδικας τούρμπο, 134
- Νέοι αναδρομικοί συγκεραστικοί κωδικοποιητές, 141
- Νέοι αναδρομικοί συγκεραστικοί κωδικοποιητές για τούρμπο κώδικες

- αναζήτηση και προσομοιώσεις, 144
- προτεινόμενες κατασκευές, 145
- Νέοι συγκεραστικοί κώδικες
 - απαλοιφή κωδικών bit, 106
 - ευέλικτοι, 116
 - κατασκευή–αναζήτηση, 109
 - προσομοιώσεις, 121
 - προτεινόμενες κατασκευές, 117–119
- Πίνακας γεννήτορας, 34
 - trellis κανονικός, 58
 - βασικός, 39
 - ελαχίστου ενεργού διαστήματος, 54
 - ισοδύναμος, 39
 - κανονικός ή ελάχιστος-βασικός, 40
 - καταστροφικός, 40
 - πεδίο χρόνου, 37
 - πολυωνυμικός, 39
 - συστηματικός, 35
- Πίνακας ισοτιμίας, 35
- Πιθανότητα σφάλματος, 33
 - αποκωδικοποίηση Viterbi, 46
 - τούρμπο κώδικα, 137
- Πολυπλοκότητα
 - trellis, 57
 - PCC trellis module, 107
 - ελάχιστη, 69
 - ελάχιστο trellis module, 70
 - νέοι ευέλικτοι συγκεραστικοί κώδικες, 107
 - trellis (TC), 53
 - αλγόριθμος Viterbi, 46
 - υπολογιστική, 48